

ORDIX news

Das IT-Magazin der ORDIX AG



Transaction survival – der Panther überlebt

44 | Informix: Neue High Availability-Funktion ab der Version 11.70

6 | Android - Von Layouts und Locations
Java Programmierung auf mobilen Geräten

36 | Scrum - Agiles Projektmanagement
Interview mit Lars Eisenblatt

14 | Jour Fixe – aber richtig!
Projektmanagement in der Praxis

40 | Virtualisierung & Hochverfügbarkeit
Technische Umsetzung mit Oracle VM

Eine gute ARCHITEKTUR ist das FUNDAMENT für Ihren Erfolg!

Besuchen Sie unsere
Seminare zum Thema
IT-Architekturen!



IT-Architekturen

Mit unseren Seminaren bieten wir Ihnen einen umfassenden Überblick über Konzepte und Methoden für das Management von IT-Unternehmensarchitekturen, dem Enterprise Architecture Management (EAM). Dabei reicht das Spektrum von der Erarbeitung effizienter IT-Strategien über Arbeiten mit der Architektur-Governance bis zur erfolgreichen Gestaltung von IT-Landschaften.

Die nächsten Seminartermine:

16.11.-18.11.2011	IT-Architekturen	Trainingszentrum Wiesbaden
13.02.-15.02.2012	IT-Architekturen	Trainingszentrum Wiesbaden
23.04.-25.04.2012	IT-Architekturen	Trainingszentrum Wiesbaden

Des Weiteren bieten wir Ihnen noch ein umfangreiches Seminarprogramm zu den Themen:
IT-Management | IT-Strategien | IT-Risikomanagement | IT-Prozessmanagement

Weitere Informationen und Anmeldung unter:
<http://training.ordix.de/>

ORDIX AG

Jobs weg

Paderborn, September 2011

Bei der Überschrift handelt es sich weder um eine Nachricht der Bundesagentur für Arbeit, noch eine Schlagzeile der Bildzeitung zum Sommerloch oder gar eine drohende neue Weltwirtschaftskrise (auf Wunsch kann man Weltwirtschaft auch durch Euro ersetzen), vielmehr ist darunter der Rücktritt von Steven Jobs zu verstehen. Damit ist zwar auch ein Job weg, aber der ist mit Tim Cook schon wieder besetzt.

Zu der Zeit als man mit Pad noch den Kosmetik Wattebausch bezeichnete und noch keiner an einen Tablet PC dachte, stand Apple kurz vor der Pleite und überlebte nur dank Millionen des „Feindes“ Microsoft. Dann „erfand“ Jobs den MP3 Player (oder genau genommen er erfand den Begriff iPod, denn MP3 Player gab es schon länger) und schon verdiente Apple wieder Geld.

Wenn der Gutmensch Paul McCartney (jener von den Beatles) eben kein solcher wäre, er hätte Apple wieder in die Pleite schicken können. Denn als Jobs den Namen Apple wählte, hatten Lennon und McCartney schon weit über 10 Jahre ihre Musik unter dem gleichen Namen veröffentlicht. Wozniak (der Partner von Jobs) und Jobs durften den Namen nur verwenden, so lange Apple (der mit dem Computer) nicht im Musikgeschäft mitmischen würde.

Als Apple nun mit iTunes den Musikmarkt revolutionierte, gab es erst mal keine Apple Musik (von den Beatles) bei Apple. Erst zu Beginn des Jahres einigte man sich (für Apple, die mit dem Computer) gütlich vor allem für Apple, der mit dem Computer, vermutlich weil McCartney trotz Scheidung immer noch genug Geld hat.

Apple war also in den letzten Jahren der Shooter schlechthin und hat mittlerweile einen Börsenwert, der ungefähr viermal so hoch ist wie der von Siemens (immerhin dem teuersten Unternehmen im DAX). Woran das liegt? Sicher nicht an der Dividende (denn die schüttet Apple nicht aus) und eigentlich auch nicht an den Produkten, die sind zwar m.E. ganz nett, auch die Ideen sind sicherlich sehr gut, nur im Detail stecken vielfach die Probleme und seien es nur finanzielle, weil das Gerät als solches ohne teure Zusatz Apps mehr oder weniger unbrauchbar ist ¹.

Man wird also sehen, ob Mr. Cook Nichtigkeiten um die Apple Produkte genauso clever verkaufen kann wie Steven Jobs, der eigentlich (man siehe selbst beim Namen) immer zu spät kam, das aber immer hervorragend verkaufte. Die Börse hat diese Nachricht erstaunlich gelassen quittiert.

Neben dem Abgang von Jobs gibt es zur Zeit auch vom Niedergang einer anderen Institution zu berichten: Die FDP, eine Partei und ihre Wähler schaffen sich ab. Ich will nicht annehmen, dass das an der intensiven Nutzung von iPads (hier jetzt die Tablet PCs) während den Vorstandssitzungen liegt, wie man neulich in einem Bericht des heute-Journals nach der Pleite in Mecklenburg-Vorpommern sehen konnte. Vielmehr scheint die Partei mit ihren „Bubis an der Spitze“ (Zitat meiner Mutter, 89, und immerhin bis weit in die 70er jahrelange FDP Wählerin) eben kaum einen Wähler mehr zu erreichen (in „Meck-Pomm“ waren es gerade mal ca. 20.000).

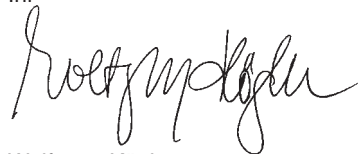
Natürlich sind das noch ein paar mehr Wähler als unsere News Leser hat, aber wir wollen auch nicht mitregieren sondern Sie gut informieren. Dieses Mal überraschen wir Sie mit dem Thema Scrum (zum ersten Mal in der News), weniger überraschend als vielmehr informativ kommen die Fortsetzungen einiger Artikel daher. Dazu zählen neben Oracle APEX und einem Anwendungsbeispiel aus dem Bereich Backup und Recovery, auch neue Features der aktuellen Oracle Grid Control Version und neue Hochverfügbarkeitsaspekte bei Informix.

Neu sind unsere Artikel über DB2 und REXX sowie über die Implementierung von Dependency Injection mit dem Framework Weld und nicht zuletzt zeigen wir, dass es neben dem iPhone auch andere Smart Phones gibt und wie Sie Anwendungen für Android Handys programmieren können.

Mit Gates (Jahrgang 1955) und Jobs (ebenfalls 1955 geboren) sind nun zwei Personen von der aktiven Bühne gegangen und wirken nur noch im Hintergrund oder in ganz anderen Bereichen. Vielleicht sollte ich (ebenfalls Jahrgang 1955) mir mal Gedanken machen.

Bis dahin wünsche ich Ihnen viel Spaß mit Apple Produkten 😊, keinen Stress mit dem Euro und sonstigen Krisen sowie viel Vergnügen beim Lesen der News.

Ihr



Wolfgang Kögler



¹ W. Kögler, ORDIX News Editorial 4/2010



6 | Android - Von Layouts und Locations



14 | Jour Fixe – aber richtig!

Java/JEE

- 6..... **Java Programmierung auf mobilen Geräten (Teil III): Android - Von Layouts und Locations**
Dieser Artikel gibt anhand einer konkreten Beispielapplikation einen Einblick in die Programmierung mit Android.
- 26..... **Weld: Der Java Einspritzmotor**
Das Framework Weld hilft die Abhängigkeiten zwischen Komponenten oder Objekten zu minimieren, dabei nutzt es das Entwurfsmuster der Context Dependency Injection.

Projektmanagement

- 14..... **Projektmanagement in der Praxis (Teil II): Jour Fixe – aber richtig!**
Die wichtigsten Regeln für regelmäßige Besprechungen in Projekten werden in diesem Artikel vorgestellt.
- 36..... **Agiles Projektmanagement mit Scrum: Agil im Betriebsprojekt! Ein Interview mit Lars Eisenblatt**
Anhand eines konkreten Projektbeispiels erläutert Lars Eisenblatt welche Faktoren und Vorbereitungen getroffen werden müssen, wenn Scrum im Projektmanagement eingesetzt werden soll.

Datenbanken

- 10..... **OEM Grid Control 11g (Teil III): Service bitte! SLA Monitoring mit dem Enterprise Manager 11gR1**
Dieser Artikel zeigt die Möglichkeiten, die der aktuelle Enterprise Manager bietet, um Services zu definieren, zu überwachen und auszuwerten.
- 18..... **Oracle Application Express (APEX) (Teil II): Backup-Überwachung in der ORDIX Backup GUI**
Wurden Ihre Datenbanken ordnungsgemäß und zu den gewünschten Zeiten gesichert? Die mit APEX entwickelte ORDIX Backup GUI (ORBAG) hilft Ihnen dabei diese Frage zu beantworten.
- 21..... **DB2 Administration mit REXX**
Perl und Shell sind die gebräuchlichen Skriptsprachen für DB2 Datenbanken. In diesem Artikel stellen wir Ihnen die alternative Skriptsprache REXX vor.
- 31..... **ETL im Data Warehouse am Beispiel IBM DataStage (Teil III): Stage für Stage zum Ziel**
Im letzten Teil dieser Reihe werden der Aufbau und die Funktionsweise eines DataStage Job erläutert.
- 40..... **Zertifizierte Servervirtualisierung aus dem Hause Oracle (Teil II): Virtualisierung und Hochverfügbarkeit mit Oracle VM**
Dieser Artikel erläutert die technische Umsetzung einer Servervirtualisierung mit Oracle VM.
- 44..... **Informix IDS 11.70 (Teil II): Neue Hochverfügbarkeitsfunktion Transaction survival – der Panther überlebt**
Dieser Artikel gibt einen Einblick in die neue Hochverfügbarkeitsfunktion Transaction survival.



26 | Weld: Der Java Einspritzmotor



36 | Agiles Projektmanagement mit Scrum

Aktuell

- 17..... Larry Ratlos
- 34..... DOAG Konferenz und Schulungstag 2011
Wissentransfer auf hohem Niveau

Standards

- 03..... Editorial
- 04..... Inhalt
- 05..... Impressum
- 24..... Seminarübersicht:
September 2011 bis Februar 2012

Impressum

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

Redaktion: Jens Pothmann, Evelyn Ernst

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

Auflage: 9.200

Druck: Druckerei Bösmann, Detmold

Bildnachweis:

- © istockphoto.com | mediaphotos | Useful yoga
- © shutterstock.com | Photocreo | Wind turbines
- © flickr.com | 3370627355_2ea9d46297_o.jpg
- © flickr.com | LGEPR
- © sxc.hu | spekulator
- © shutterstock.com | Jorge Barradas de Casais | Sunset
- © sxc.hu | hubidos | fire flames
- © sxc.hu | leocub | laptop1
- © istockphoto.com | GlobalIP | Black leopard
- © flickr.com | See-ming Lee | SML

Autoren dieser Ausgabe:

Andreas Baier, Ole Breimann, Lars Eisenblatt, Evelyn Ernst, Andreas Flügge, Benedikt Georgi, Patrick Hecker, Carsten Hummel, Matthias Jung, Wolfgang Kögler, Andreas Massmann, Michael Spoden

Die Zeitschrift ORDIX News wird von der ORDIX AG an ausgewählte Kunden verteilt und kann für 3,20 Euro bestellt werden. Sie können die Zusendung der ORDIX News jederzeit ohne Angabe von Gründen schriftlich (z.B. Brief, Fax, E-Mail) abbestellen. Die neueste Ausgabe wie auch ältere Ausgaben finden Sie im Archiv der ORDIX News im Internet unter: <http://www.ordix.de>. Schauen Sie mal rein!

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

Copyright:

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

Haftung:

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

Warenzeichen:

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



Java Programmierung auf mobilen Geräten (Teil III)

Android - Von Layouts und Locations

Der Artikel richtet sich an Entwickler, die sich für die Programmierung auf mobilen Geräten interessieren.

Nach den Erläuterungen und Vorbereitungen aus den ersten beiden Teilen unserer Artikelreihe ([1] [2]), wollen wir uns nun einer konkreten Beispiellapplikation zuwenden. Dazu werden wir das Gerüst der vom Wizard generierten Anwendung Schritt für Schritt erweitern und die gewünschten Funktionalitäten hinzufügen.

MyLocation

Für einen ersten Überblick wollen wir kurz festhalten, was die zu entwickelnde Applikation „MyLocation“ leisten soll. Im Wesentlichen wird sie aus zwei Teilen bestehen. Der erste Teil wird in einer View den aktuellen Status des GPS-Empfängers textuell darstellen, also die zuletzt festgestellten Geokoordinaten. Die View wird zusätzlich einen Button enthalten, der beim Betätigen eine zweite View öffnet, die die aktuelle Position auf einer Karte anzeigt.

Layout und Views

Das Layout von Android-Anwendungen wird standardmäßig in XML-Dateien gespeichert. Der Wizard hat beim Anlegen unserer Anwendung bereits das Default-Layout erstellt, es befindet sich in der Datei `main.xml` im Verzeichnis `res/layout`. Für die Bearbei-

tung des Layouts, stellt das Android Software Development Kit (SDK) einen grafischen Editor zur Verfügung. Alternativ können die XML-Dateien natürlich auch mit jedem beliebigen Text-Editor bearbeitet werden.

Wir wollen unser Layout einfach halten und alle Elemente untereinander darstellen. Der Umgang mit dem Editor bedarf einer gewissen Einarbeitung, die Bedeutung der einzelnen Elemente und Attribute findet man auf den Referenzseiten des SDK. Für unsere Applikation bedienen wir uns des `LinearLayout`, in das wir den Applikationsnamen, die zwei Titel- und Werttexte sowie den Button platzieren.

Mit den Attributen der einzelnen Elemente sorgen wir für die linksbündige Ausrichtung der Label, die rechtsbündige Ausrichtung der Werte und die Zentrierung von Applikationsnamen und Button. Das dazugehörige Layout wird im Editor, wie in Abbildung 1 zu sehen ist,

dargestellt. Die XML-Darstellung entspricht der Abbildung 2.

Die dort referenzierten Texte befinden sich in der Datei `strings.xml` im Verzeichnis `res/values` (siehe Abbildung 3). Diese können ebenfalls mit dem im SDK enthaltenen Editor oder auch textuell bearbeitet werden. Soll die Anwendung später internationalisiert werden, muss lediglich eine sprachspezifische Version dieser Datei erstellt und dann in einer leicht modifizierten Verzeichnisstruktur (`res/values-<länderkürzel>`) abgelegt werden. Das Android Framework sorgt anhand der Spracheinstellungen des Gerätes dann für die Verwendung der entsprechenden Datei.

Starten wir die Applikation nun in Eclipse (MyLocation -> Run As -> Android Application), wird die Anwendung im Emulator installiert und gestartet. Übersetzt und gebaut wird sie automatisch mit den Standardeinstellungen bei jeder Änderung. Wir sehen die erstellte View auf dem virtuellen Gerät. Diese sieht noch reichlich unspektakulär aus, denn es werden nur die Default-Daten angezeigt und hinter dem Button liegt noch keine Funktionalität.

Standortbestimmung

Im nächsten Schritt möchten wir erreichen, dass die Anzeige automatisch aktualisiert wird, sobald sich der momentane Standort verändert. Android stellt zu diesem Zweck einige Klassen zur Verfügung, die das Auslesen des GPS Device ermöglichen und die Applikation darüber informieren können.

Im Package `android.location` befinden sich die zur Standortbestimmung benötigten Klassen. Von zentraler Bedeutung ist hier der `LocationManager`. Dieser ist in der Lage vom System einen Dienst anzufordern, der uns Geoinformationen liefern kann. Diese Klasse (`LocationManager`) wird allerdings nicht direkt instanziiert. Die Methode `getSystemService(Context.LOCATION_SERVICE)` liefert eine solche Instanz, die wir u.a. dazu benutzen können, um uns über veränderte Geopositionen informieren zu lassen.

Grundlage dafür ist, dass wir eine Klasse das Interface `LocationListener` implementieren lassen, die über die Methode `onLocationChanged` die Informationen über die veränderte Position verarbeiten kann. Das Interface verlangt die Implementierung weiterer Methoden, die in unserem Fall aber ohne Funktionalität bleiben können.

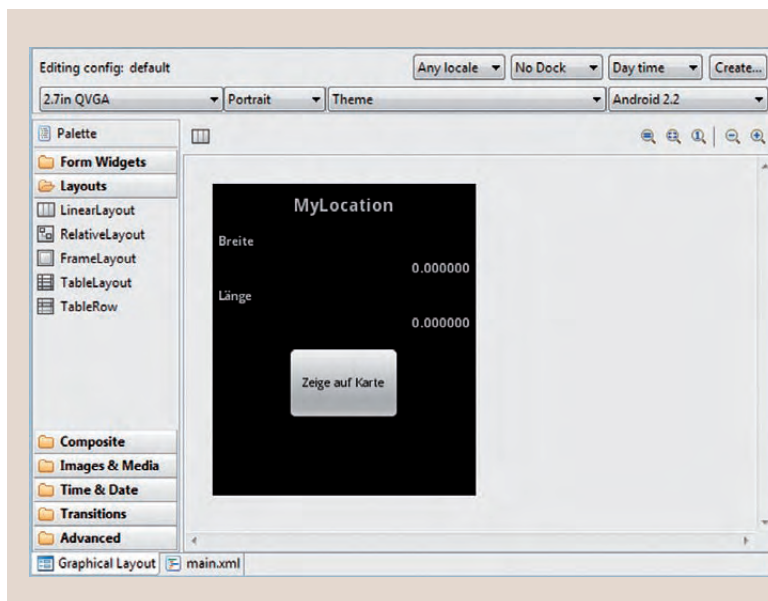


Abb. 1: Layout im grafischen Editor.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:orientation="vertical"
    android:layout_height="wrap_content">
    <TextView
        android:layout_height="wrap_content"
        android:layout_margin="10px"
        android:textSize="8pt"
        android:text="@string/app_name"
        android:layout_width="wrap_content"
        android:gravity="center_horizontal"
        android:layout_gravity="center_horizontal"
        android:textStyle="bold"
        android:id="@+id/TitleTextView"/>
    <TextView
        android:layout_height="wrap_content"
        android:layout_margin="5px"
        android:textSize="6pt"
        android:text="@string/breiteString"
        android:layout_width="wrap_content"
        android:layout_gravity="left"
        android:textStyle="bold"
        android:id="@+id/BreiteTextView"/>
    ...
    <Button
        android:id="@+id/KarteButton"
        android:text="@string/karteString"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="10px">
    </Button>
</LinearLayout>
```

Abb. 2: Layout in `main.xml`.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">MyLocation</string>
  <string name="laengeString">Länge</string>
  <string name="breiteString">Breite</string>
  <string name="laengeWert">0.000000</string>
  <string name="breiteWert">0.000000</string>
  <string name="karteString">Zeige auf Karte</string>
</resources>
```

Abb. 3: Texte in strings.xml.

```
public class MyLocation extends Activity implements Location-
Listener {
  LocationManager locationManager;
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    locationManager=(LocationManager) getSystemService(
      Context.LOCATION_SERVICE);
  }
  @Override
  protected void onPause() {
    super.onPause();
    locationManager.removeUpdates(this);
  }
  @Override
  protected void onResume() {
    super.onResume();
    locationManager.requestLocationUpdates(
      LocationManager.GPS_PROVIDER, 0, 0, this);
  }
  @Override
  public void onLocationChanged(Location location) {
    TextView breiteView = (TextView)findViewById(
      R.id.BreiteWertView);
    TextView laengeView = (TextView)findViewById(
      R.id.LaengeWertView);
    breiteView.setText(String.valueOf(
      location.getLatitude()));
    laengeView.setText(String.valueOf(
      location.getLongitude()));
  }
  /* unwichtige Methoden ausgelassen ... */
}
```

Abb. 4: MyLocation-Activity.

Der Einfachheit halber machen wir dies direkt mit unserer Haupt-Activity **MyLocation**.

Die Methode **onLocationChanged** in unserem Beispiel aktualisiert lediglich die Werte (Texte) in der View. Sie stellt mittels der Methode **findViewById** die Views der Koordinatenwerte in der Activity fest und ändert ihre Inhalte entsprechend.

Jetzt fehlt nur noch die Registrierung unserer Activity als **LocationListener**. Dazu nutzen wir die beiden Lebenszyklusmethoden **onResume** und **onPause**. Wir erinnern uns: wird eine Activity sichtbar, so wird deren Methode **onResume** aufgerufen und sie wechselt in den Status „Aktiv“. Wird sie hingegen durch andere Komponenten verdeckt, wechselt sie in den Status „Inaktiv“, nachdem die Methode **onPause** passiert wurde. Praktisch bedeutet dies, dass wir in unserer Activity beide Methoden überschreiben und die Registrierung bzw. Deregistrierung unseres Listener in der jeweiligen Methode durchführen.

Energiespar-Aspekte

Die Klasse **LocationManager** bietet dazu die beiden Methoden **requestLocationUpdates** und **removeUpdates** an. Durch dieses Vorgehen garantieren wir ein möglichst energieeffizientes Arbeiten unserer Applikation. Die Daten werden nur dann angefordert und ausgewertet, wenn die Anwendung auch wirklich aktiv ist.

Die Registrierungsmethode verlangt als Parameter den Typ des Location Provider (neben dem GPS-Empfänger kann es noch weitere geben), den zu registrierenden Listener sowie Angaben (Zeit und Entfernung) zum Intervall der Aktualisierung. Diese letzten beiden Informationen haben ebenfalls Auswirkungen auf den Energieverbrauch und sollten je nach Anwendungsfall geeignet gewählt werden. In unserem Beispiel wird der Einfachheit halber eine ständige Aktualisierung parametrisiert, die aber auch einem maximalen Energiebedarf entspricht. Abbildung 4 zeigt den Code für unsere Activity.

Berechtigung

Um die Applikation starten zu können, ist jetzt noch ein letzter Schritt notwendig. Durch die Nutzung der Location-API haben wir unserer Anwendung eine Fähigkeit verliehen, die eine besondere Berechtigung verlangt, nämlich die Nutzung des GPS-Empfängers.

Das Android-Sicherheitskonzept kennt ungefähr 100 verschiedene Berechtigungen, die dem Anwender zeigen, was eine zu installierende Applikation auf seinem Gerät alles ausführen kann. Jede dieser Berechtigungen wird vom Entwickler im Manifest der Anwendung registriert. Vergisst er dies, beendet sich die

Anwendung, sobald eine entsprechende Funktion angesprochen wird mit einer Exception.

Das Manifest ist ebenfalls eine XML-Datei. Das SDK bringt aber auch hierfür einen eigenen Editor mit. In unserem Fall müssen wir die Berechtigung `ACCESS_FINE_LOCATION` hinzufügen. Abbildung 5 zeigt das entstandene Manifest.

Test im Emulator

Für einen ersten Test benötigen wir kein reales Android Device, der Emulator hat die Möglichkeit für unsere Anwendung GPS-Koordinaten zu simulieren. Dazu wechseln wir nach dem Start der Applikation in die DDMS-Perspektive (Dalvik Debug Monitor Service). Dort befinden sich im linken unteren Bereich des Emulator-Control-Reiters die Location Controls.

Hier eingegebene Koordinaten werden nach Betätigung des Send-Buttons dem virtuellen Device als neue Position übermittelt. Wie erwartet, werden daraufhin die Felder unserer Activity aktualisiert (siehe Abbildung 6).

Fazit

Wir haben gesehen, wie eine Android-Applikation prinzipiell aufgebaut ist. Die Funktionsweise von Layouts und Views sowie das Auslesen des GPS-Empfängers haben wir ebenso betrachtet, wie die Auswirkungen der Nutzung von sicherheitsrelevanten Funktionen auf das Manifest.

Wir konnten aufzeigen, dass wir mit relativ wenig Source Code in kurzer Zeit eine voll funktionsfähige Anwendung erstellen können und diese auch ohne physisches Gerät trotz hardwarenaher Funktionen sehr gut testen konnten.

Im nächsten Artikel dieser Reihe werden wir mit der Kartendarstellung ein fortgeschrittenes Thema behandeln und uns genauer mit externen Libraries und Intents beschäftigen.



Andreas Flügge
(info@ordix.de).

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/
android"
    package="de.objectsystems.mylocation"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission android:name="android.permission.ACCESS_
FINE_LOCATION"></uses-permission>
    <application android:icon="@drawable/icon"
android:label="@string/app_name">
        <activity android:name=".MyLocation"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Abb. 5: AndroidManifest.xml.

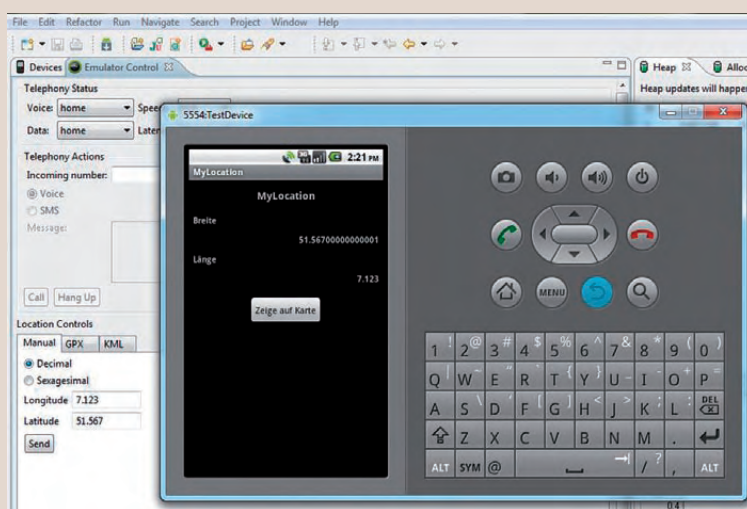


Abb. 6: Simulierte GPS-Koordinaten.

Links

- ▶ [1] ORDIX News Artikel 1/2011 „Android - Java macht mobil“:
http://www.ordix.de/ORDIXNews/1_2011/android_java_macht_mobil.html
- ▶ [2] ORDIX News Artikel 2/2011 „Android - Von Aktivitäten und Absichtserklärungen“:
http://www.ordix.de/ORDIXNews/2_2011/android_t2.html
- ▶ [3] Downloadseite des Android Developer:
<http://developer.android.com/sdk/index.html>
- ▶ [4] Installation des Android SDK:
<http://developer.android.com/sdk/installing.html>

Das Werkzeug Grid Control für das geschäftsorientierte IT-Management von Oracle (Teil III)

Service bitte! SLA Monitoring mit dem Enterprise Manager 11gR1

Der Artikel richtet sich an Administratoren bzw. Personen, die sich für Service Level Management interessieren.

Die Vereinbarung und Überwachung von Service Level Agreements (SLA) wird zunehmend wichtiger. Oftmals besteht eine Unsicherheit, wie mit den daraus resultierenden Reporting-Anforderungen umgegangen werden soll. Mit dem vorliegenden Artikel möchten wir Ihnen näherbringen, wie man mit der aktuellen Version des Enterprise Manager (11gR1) Services definieren, überwachen und auswerten kann.

SLA - oder was?

In der IT gibt es zahlreiche Dienstleistungsvereinbarungen zwischen den unterschiedlichen Parteien. Es kann sich hierbei um eine klassische Kunden-Dienstleister-Beziehung oder um eine Vereinbarung zwischen zwei internen Abteilungen (z.B. zwischen IT und Controlling) handeln.

Über Service Level Agreements wird versucht, eine transparente Kontrollmöglichkeit zwischen den Vertragsparteien zu schaffen, um zu ermitteln, ob die vereinbarte Leistung tatsächlich erbracht wurde. Die Grundlage eines SLA beruht dabei auf Parametern wie der Leistungsbeschreibung, der Verfügbarkeit und der Performance der zu erbringenden Leistung (Service).

SLA + EM GC 11gR1

Eine der Hauptaufgaben des Enterprise Manager (EM) ist die Überwachung der Verfügbarkeit und die Erhebung von Performance-Indikatoren der unterschiedlichen IT-Komponenten.

Mittlerweile lassen sich auch Oracle-fremde Produkte (wie z.B. ein Microsoft SQL Server und / oder ein NetApp Filer) über diverse Plugins problemlos in den EM einbinden.

Damit ist ein Großteil der essentiell notwendigen Datenbasis zur Ermittlung eines Service Level „ab Werk“ im EM abruf- und nutzbar.

SLA mit System

Vor der Definition der zu erbringenden Dienstleistung ist zunächst einmal das dafür erforderliche System zu beschreiben. Unter dem Stichwort System ist die Infrastruktur zu verstehen, welche für die Erbringung der Dienstleistung notwendig ist.

Im Rahmen dieses Artikels wollen wir einen Beispielservice etablieren: Es handelt sich dabei um eine kleine Webanwendung (Web Shop), die über einen Apache Webserver auf eine Oracle Datenbank zugreift. Die für den Service notwendigen Komponenten sind in diesem Fall:

- Das Betriebssystem (Oracle Enterprise Linux 5.5)
- Die Oracle Datenbank (Oracle 10g Express Edition)
- Die APEX (HTML 3.2) Anwendung (inkl. Webserver)
- Der Listener (auch wenn dieser für den Betrieb der Anwendung nicht notwendig ist)

Systembildung

Ein System ist aus Sicht des EM ein eigener Zieltyp, der dialogorientiert erstellt werden kann. Im Wesentlichen werden die bereits im EM registrierten Ziele (ein Agent muss also die entsprechenden Komponenten erfasst haben) unter einem frei wählbaren „Label“ zusammgeführt.

Auf Wunsch kann man auch eine topografische Übersicht der Systemlandschaft generieren (siehe Abbildung 1). Sobald das System definiert ist, können die Dienste in Angriff genommen werden.

Dienst (!) Leistung (!)

Auch die Dienste sind eigene Zieltypen, die angelegt werden müssen. Nachdem wir einen Namen gewählt und das dahinter stehende System (es können auch mehrere sein) bestimmt haben, können die zu erbringenden Eckdaten definiert werden.

Derzeit stehen ca. 20 verschiedene Testgruppen zur Verfügung. Sie reichen von einem „Host Ping“ über „SQL-Statements“ und per Browser aufgezeichnete Webtransaktionen (Folge von Webseiten-Zugriffen) bis hin zur Einbindung von eigenen Skripten (z.B. Shell).

Um in diesem Beispiel den Überblick nicht zu verlieren, beschränken wir uns auf drei kleinere Tests:

- Eine Folge von Webseiten-Zugriffen (aufgezeichnet über das Oracle Tool Transaction Recorder)
- Die Ausführung eines SQL-Kommandos auf der Oracle Datenbank
- Das Anpingen des Datenbank-Server

Für alle unsere Testszenarien können nun Metriken und Performance-Kriterien (gerne auch mehrere je Test) definiert werden. Wir halten das Beispiel an dieser Stelle einfach und definieren je Test jeweils eine Metrik:

- Der Aufruf der Webseiten muss innerhalb von 5 Sekunden komplett abgelaufen sein.
- Das SQL-Kommando muss innerhalb von 1 Sekunde min. 5 Ergebniszeilen geliefert haben.
- Der Ping auf den Host muss nach 20 Millisekunden erfolgreich erledigt sein.

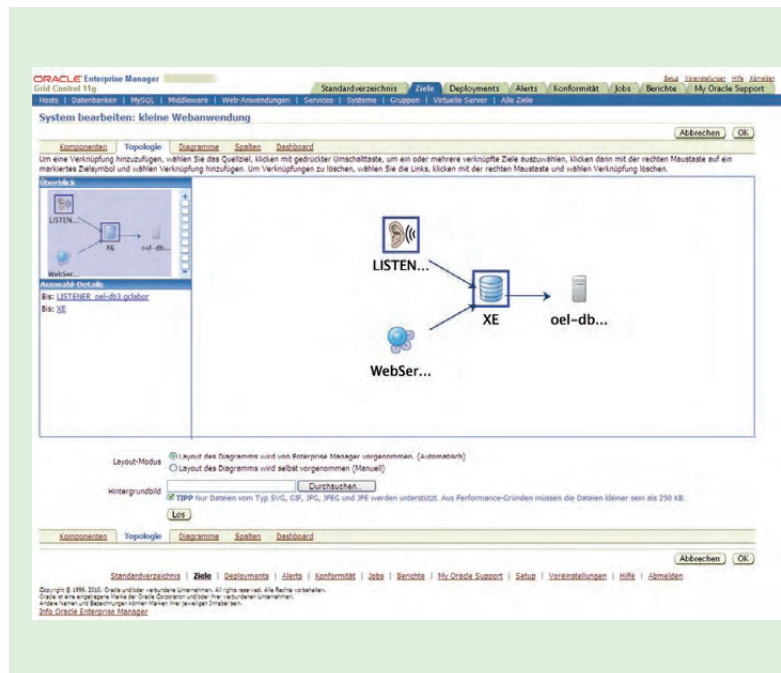


Abb. 1: Die Grundlage für die Überwachung von Services ist ein System. Dieses kann einfach als eigener Zieltyp dialogorientiert erstellt werden.



Abb. 2: SLA-Informationen im Überblick. Auf einen Blick sind die Testergebnisse und der Zielerreichungsgrad des SLA einsehbar.

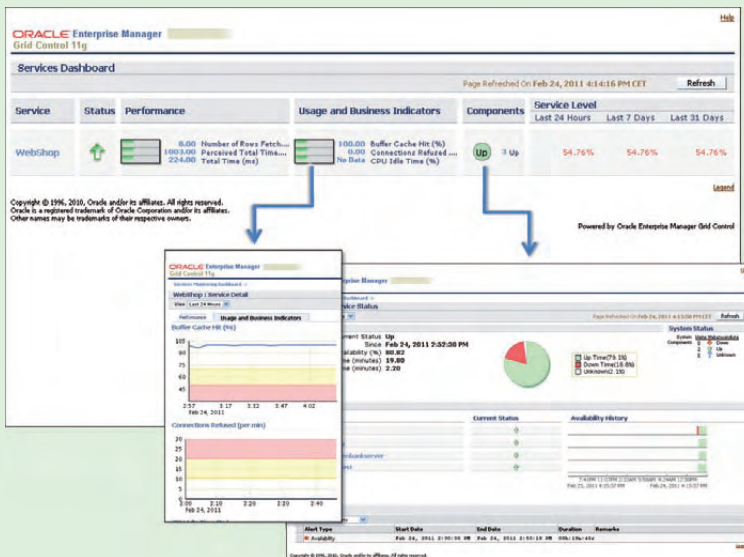


Abb. 3: Reports können beispielsweise per Dashboard und / oder per Mail an Kunden übermittelt werden.

Aushilfskräfte

Um ein aussagekräftiges Bild über die Güte der Dienstleistung zu bekommen, brauchen wir nun noch Tester, welche die Qualität und Verfügbarkeit der Services überprüfen. Im Oracle Sprachgebrauch werden diese Komponenten Beacons (engl. für Signal- oder Leuchtfeuer) genannt.

Dies sind in der Regel EM-Agenten, welche die entsprechenden Testszenarien ausführen und protokollieren. In unserem Fall beauftragen wir zwei Agenten mit der Qualitätskontrolle:

- einen Agenten, der lokal auf dem Datenbank-Server installiert ist
- einen entfernten Agenten, der beispielsweise bei einem Kunden oder an einem anderen Standort installiert wurde

Bei der Auswertung der Testergebnisse können nun verschiedene Algorithmen definiert werden. Wahlweise werden nur die Ergebnisse des schnellsten / langsamsten Agenten berücksichtigt, die Ergebnisse summiert oder gemittelt oder es werden dedizierte Vorgaben

je Beacon und Test definiert, die einzuhalten sind.

So kann z.B. bestimmt werden, dass die Antwortzeit eines Agenten am anderen Ende der Welt doppelt so lang sein kann, wie die des lokal installierten Agenten.

Die Abrechnung

Am Ende wird abgerechnet. Für die Errechnung des SLA werden neben der generellen Verfügbarkeit der Systemkomponenten auch die Performance-Messungen (sofern dies gewünscht ist) berücksichtigt.

Auf der entsprechenden Serviceseite werden nun Quoten für die Serviceverfügbarkeit und das Service Level Agreement ausgewiesen. Über die unterschiedlichen Grafiken und Links lassen sich per Drill-Down-Funktionalität auch Details zu den einzelnen Tests und deren Ergebnissen ermitteln (siehe Abbildung 2). So kann im Bedarfsfall sehr genau analysiert werden, welche Faktoren zur Erreichung des SLA optimiert werden müssen.

Die Quittung

Auf Grundlage der erhobenen Daten lässt sich natürlich auch ein kundenorientiertes Berichtswesen erstellen, ohne dass der Kunde zwingend Zugriff auf den Enterprise Manager erhalten muss. Dies ist über die normale Reporting- und / oder Dashboard-Funktion des Enterprise Manager problemlos möglich.

Die Berichte können dabei periodisch automatisch erstellt, archiviert und per Mail oder Webfrontend dem Dienstleistungsempfänger übermittelt werden. Für diese Zwecke können die von Oracle bereits mitgelieferten Report-Templates oder eigene Reports genutzt werden (siehe Abbildung 3).

Fazit

Mit dem Enterprise Manager lassen sich SLAs definieren und überwachen. Es stehen unzählige Parameter für die Definition der Servicequalität bereit. Dies hilft sicherlich auf der einen Seite auch unerfahrenen Anwendern, sich diesem Thema zu nähern, da anhand der vorgegebenen Indikatoren schnell sinnvolle Definitionen zusammengestellt werden können. Darüber hinaus können diese Auswahllisten gut in der Kundenkommunikation als „Bestellliste“ eingesetzt werden.

Auf der anderen Seite kann eine zu großzügige Definition von Metriken auch schnell zu Verwirrungen und ggf. unübersichtlichen Ergebnissen führen. Alles in allem ist der Enterprise Manager jedoch ein gutes Instrumentarium, um sich mit dem Thema SLA Management praxisorientiert auseinander zu setzen.



Matthias Jung
(info@ordix.de).

Glossar

SLA	Service Level Agreement – Bezeichnet einen Vertrag über die Art und Güte (Dienstgütevereinbarung) der zu erbringenden Leistung zwischen Auftraggeber und Auftragnehmer.
Agent	Komponente des Enterprise Manager, welche auf dem Zielsystem Status und Performance-Daten (Metriken) sammelt und meldet.
Beacon	Agent, der neben seinen üblichen Aufgaben Service-Level-Tests ausführt.
Transaction Recorder	Mit dieser Funktion kann die Anwendungsnutzung von Web-Usern simuliert und aufgezeichnet werden. Beacons können diese Aufzeichnung als Testscenario verwenden.
APEX	Oracle Application Express (vormals HTML DB) - Tool für die Entwicklung von Webapplikationen.

Link

- ▶ [1] Ordix News Artikel 1/2011 „OEM Grid Control 11g (Teil I): Unter die Lupe genommen!": http://www.ordix.de/ORDIXNews/1_2011/oem_grid_control_version_11.html
- ▶ [2] Ordix News Artikel 2/2011 „OEM Grid Control 11g (Teil II): Bleiben Sie am Ball!": http://www.ordix.de/ORDIXNews/2_2011/oem_grid_control_t2.html

Seminarempfehlung: Oracle Grid Control

- ▶ **Informationen/Online-Anmeldung:** <http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Datenbanken/nr/552/index.html>

In diesem sehr übungsintensiven Workshop lernen Sie, mit Oracle Grid Control zu arbeiten und umzugehen. Sie erhalten einen Überblick über die Oberfläche und die Installation der Agenten. Nach dem Seminar sind Sie in der Lage, Oracle Datenbanken über die GUI-Oberfläche zu verwalten und zu administrieren.

Seminarinhalte

- Überblick über die Oberfläche
- Installation Grid Control
- Migration von OMS zu Grid
- Installation der Agenten
- Administration
- Alerts, Metriken, Jobs
- Advisor Framework
- ADDM
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

14.11. - 16.11.2011 in Wiesbaden

Seminar-ID: DB-ORA-35

Dauer: 3 Tage

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

Frühbucherpreis: 1.161,00 € (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>.



Projektmanagement in der Praxis: „Wie aus dem richtigen Leben ...“ (Teil II)

Jour Fixe – aber richtig!

Dieser Artikel richtet sich an Menschen, die in Projekten tätig sind.

Im Gesellschaftsleben des 18. Jahrhunderts war ein Jour Fixe eine Reihe regelmäßiger zumeist kultureller Veranstaltungen, die an einem festen Tag in der Woche oder im Monat stattfanden. Im Bereich Projektmanagement hat sich der Jour Fixe als ein fester Termin etabliert, an dem sich ein Projektteam regelmäßig zu einer Besprechung trifft, um den Projektstatus auszutauschen, offene Punkte zu klären und Entscheidungen zu treffen. Damit ein Jour Fixe ein effizientes Mittel für die Steuerung eines Projektes werden kann, sollten einige wichtige Regeln eingehalten werden. Die Wichtigsten werden im vorliegenden Artikel vorgestellt.

Jour Fixe versus Plauderstunde

Wer kennt Sie nicht - Projektsitzungen, die zu Plauderstunden verkommen oder als Bühne für die Selbstdarstellung einiger Weniger herhalten müssen? Oder auch langwierige Besprechungen, in denen das „Wer-hat-Recht?“-Spiel gespielt wird oder tiefgreifende fachliche Diskurse ohne Ergebnis geführt werden. Häufig werden solche Sitzungen Jour Fixe genannt - allerdings vollkommen zu Unrecht!

Der feste Termin

Ein Jour Fixe im Projektmanagement ist eine regelmäßige Besprechung, die an einem festen Tag stattfindet (z.B. jeden Dienstag,

dienstags alle 14 Tage, der 1. Dienstag im Monat etc.). Ein Jour Fixe taktet somit ein Projekt. Alle Projektmitglieder wissen, dass Sie ihre wichtigen Anliegen im Jour Fixe adressieren können.

Verantwortlich für die Planung und Durchführung eines Jour Fixe ist der Projektleiter. Die Teilnehmer sollten alle Mitglieder des Projektteams - bei sehr großen Projekten - die des Kernteams sein. Aus der Projektorganisation sollten die Teilnehmer der Jour Fixe ersichtlich sein.

Ziel eines Jour Fixe ist es, gemeinsam den Status des Projektes zu ermitteln, über übergreifende Themen zu informieren, offene Punkte zu besprechen, Entscheidungen zu treffen

und die nächsten Schritte zu vereinbaren. Ziel ist es nicht, Konzepte zu erarbeiten, Fleißberichte abzugeben oder tiefgreifende Fachdiskussionen zu führen. Insofern sollte ein Jour Fixe nicht länger als 3 Stunden dauern. Gut eingespielte Projektteams können es auch in einer halben Stunde schaffen.

Die Regeln

Damit der Jour Fixe effizient durchgeführt werden kann, sollte der Projektleiter beim Kickoff des Projektes oder bei dem ersten Jour Fixe gemeinsam mit dem Projektteam die Regeln für die Durchführung des Jour Fixe vereinbaren, die dann von allen zwingend einzuhalten sind.

Der Moderator

Die Rolle des Moderators sollte jemand übernehmen, der dafür sorgt, dass die vereinbarten Regeln eingehalten werden. Ein guter Moderator sorgt zusätzlich dafür, dass alle zu Wort kommen, dass abweichende Meinungen berücksichtigt werden, dass man gemeinsam zu Entscheidungen gelangt und dass Ergebnisse festgehalten werden und nicht verloren gehen.

Er sorgt dafür, dass ein angemessenes Verhältnis zwischen Information/Präsentation einerseits und Diskussion/Entscheidungsfindung andererseits eingehalten wird. Er sorgt für Gesprächsdisziplin, lässt aber auch - wenn nötig - einmal Raum für Kreativität.

Weiterhin achtet er darauf, dass nicht ausschließlich auf die Probleme fokussiert wird, sondern dass auch Erfolge ihren Platz im Jour Fixe finden.

Wenn eine Diskussion auszufern droht, sollte der Moderator die bisher erzielten Ergebnisse zusammenfassen und eine weitere Vorgehensweise (Maßnahme) vorschlagen, die dann als offener Punkt festgehalten wird.

Das Protokoll

Darüber hinaus sollte ein Ergebnisprotokoll geführt werden – wobei die Betonung auf „Ergebnis“ liegt. Das Protokoll soll nicht den Gesprächsverlauf dokumentieren, sondern nur die wichtigsten Statusinformationen, Entscheidungen und nächsten Schritte beinhalten.

Glossar

Jour Fixe Ein fester Tag, an dem sich ein Projektteam regelmäßig zu einer Sitzung trifft, um den Projektstatus auszutauschen, offene Punkte zu klären und Entscheidungen zu treffen.

Hilfreich ist es, zusätzlich zum Protokoll eine Liste der offenen Punkte (Wer? Was? Bis wann?) zu führen, die in jedem Jour Fixe durchgesprochen und aktualisiert wird.

Damit der Projektleiter sich auf die Führung seines Projektes und auf seine eigenen fachlichen und organisatorischen Wortbeiträge konzentrieren kann, sollte er nicht selbst moderieren oder Protokoll führen, sondern diese Aufgaben entweder an fest definierte Mitglieder des Projektteams delegieren oder sie rollierend durchführen lassen.

Ein weiterer Vorteil dieser Vorgehensweise ist es, dass der Projektleiter den Jour Fixe nicht zu stark dominiert und nicht Gefahr läuft zum Alleinunterhalter zu werden. Der Jour Fixe sollte allen Projektmitgliedern „gehören“ und nicht nur dem Projektleiter.

Der Status

Die Grundvoraussetzung für einen effizienten Jour Fixe ist, dass alle Teilnehmer gut vorbereitet in die Besprechung gehen. D.h. sie kennen die Agenda, sie haben das Protokoll des letzten Treffens gelesen, sie haben die Liste der offenen Punkte bearbeitet und sie haben ihren eigenen Status vorbereitet.

Für den eigenen Status muss jeder Folgendes aufzeigen:

- Wie ist der aktuelle Stand der eigenen Aufgaben?
- Was wurde seit dem letzten Jour Fixe erreicht / nicht erreicht?
- Welche Entscheidungen müssen getroffen werden?
- Was sind die nächsten Schritte?

Es empfiehlt sich, für die Darstellung des Status eine - für das gesamte Projekt einheitliche - Dokumentenvorlage zu verwenden, die jeweils vor jedem Meeting aktualisiert wird.

Die Disziplin

Es klingt zwar banal, aber der Projektleiter muss sehr darauf achten, dass alle Teilnehmer gut vorbereitet sind und sich aktiv einbringen. Denn wenn der Schlendrian erst Einzug hält, geht die Wirksamkeit des Jour Fixe schnell verloren und die Autorität des Projektleiters ist schnell verbraucht.



Benedikt Georgi
(info@ordix.de).

Auch verlieren die gut vorbereiteten und engagierten Mitglieder schnell die Motivation, wenn andere Mitglieder wiederholt unvorbereitet und passiv am Jour Fixe teilnehmen - ohne dass das irgendwelche Konsequenzen hat. Im Übrigen sind ein pünktlicher gemeinsamer Beginn, ein pünktliches gemeinsames Ende und die Vermeidung von Störungen (auch Smartphones kann man ausschalten!) hilfreiche Tugenden auch in einem Jour Fixe.

Das Fazit

Wenn Sie nun sagen, dass das was ich über die Durchführung eines Jour Fixe in dem vorliegenden Artikel geschrieben habe, doch alles selbstverständlich sein sollte, dann gebe ich Ihnen vollkommen recht.

Leider musste ich in der Praxis häufig erleben, dass Projektmeetings in endlose Diskussionen ausarteten und ohne greifbares Ergebnis endeten. Häufig wurde mir bei Projektbeginn versichert, dass das Festlegen von Jour Fixe Regeln „eigentlich“ gar nicht nötig sei. Am Ende derselben Projekte war man sich im Rückblick regelmäßig einig, dass das Festlegen der Regeln doch eine sehr gute Idee war.

Seminarempfehlung: IT-Projektmanagement

► **Informationen/Online-Anmeldung:** <http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Datenbanken/nr/223/index.html>

In diesem Seminar erhalten Sie einen umfassenden und systematischen Überblick über alle Begriffe, Methoden und Arbeitstechniken zum Management von IT-Projekten. Sie lernen, IT-Projekte zu planen und zu steuern sowie verfügbare Arbeitshilfen projektspezifisch anzupassen.

Seminarinhalte

- Grundlagen
- Projektdefinition
- Projektplanung
- Projektdurchführung
- Projekt-Controlling
- Qualitäts- und Risikomanagement
- Arbeits- und Hilfsmittel
- Fallbeispiele und Übungen

Termine

10.10. - 14.10.2011 in Wiesbaden
16.01. - 20.01.2012 in Wiesbaden
16.04. - 20.04.2012 in Wiesbaden
06.08. - 10.08.2012 in Wiesbaden

Seminar-ID: PM-01

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>.

Larry Ratlos

Larry hat ein Passwort-Problem



Da Larry sich mit der letzten Aufgabe als Java-Spezialist ausgezeichnet hat, kommt sein Kollege Hubert Weisnixmehr mit einem Problem auf ihn zu. Um die Sicherheit im Unternehmen zu verbessern hat Hubert die Aufgabe erhalten, eine Passwort-Abfrage zu programmieren, bei der die Mitarbeiter alle 8 Wochen ihr Passwort ändern müssen.

Fehler: „Passwort ist älter als 8 Wochen“

Hubert Weisnixmehr hat bereits etwas programmiert, allerdings erhält er jedes mal die Meldung: „Passwort ist älter als 8 Wochen“, obwohl in Zeile 10 des Programms der Calendar nur 6 Wochen zurückgestellt wird. Er versteht diesen Fehler nicht.

Wie lässt sich dieser Fehler beheben bzw. wie lässt sich das Problem mit einem einzigen Buchstaben korrigieren?

Können Sie Larry helfen?

Auf Ihren Lösungsvorschlag freut sich Larry bis zum **7. Oktober 2011** an kniffel@ordix.de.

Das Larry-Problem wurde gelöst!

Viele freundlichen Helfer konnten durch ihre Lösungen überzeugen.

Er bedankte sich mit einem kleinen Präsent bei den schnellsten Einsendern:

Herr Christian Krone, Herr Klaus Schüller, Herr Christoph Stockmayer und Herr Matthias Laux.

Larry freut sich auf Ihre Lösungen zum Passwort-Problem!

```
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;

public class Login {
    public static void main (String args[]) {
        // Setzen eines Datums mit dem Wert vor 6 Wochen
        Calendar passwortGewechseltCalendar = new GregorianCalendar();
        passwortGewechseltCalendar.set(Calendar.WEEK_OF_YEAR, passwortGewechseltCalendar.get(Calendar.WEEK_OF_YEAR) -6 );
        Date passwortGewechseltDate = passwortGewechseltCalendar.getTime();

        // Anzahl der Wochen, wann das Passwort gewechselt werden muss
        int wochen = 8;
        // Wochenanzahl in Millisekunden
        long wochenInMilli = wochen * 7 * 24 * 60 * 60 * 1000;

        // Wenn das aktuelle Datum weniger dem letzten Passwortwechsel größer ist als die 8 Wochen,
        // muss das Passwort gewechselt werden.
        if ( new java.util.Date().getTime() - passwortGewechseltDate.getTime() > wochenInMilli ) {
            System.out.println(„Passwort ist älter als 8 Wochen!“);
            // passwortWechsel();
        }
    }
}
```

Abb. 1: Der Source Code von Hubert Weisnixmehr - Was muss Larry ändern, damit er die richtige Ausgabe erhält?

Backup-Überwachung in der ORDIX® Backup GUI

Der Artikel richtet sich an Entwickler, die sich für APEX interessieren.

Mit Application Express (APEX) stellt Oracle Entwicklern und Administratoren ein Werkzeug zur Verfügung, mit dem einfach und unkompliziert Anwendungen und Reports erstellt werden können. ORDIX® hat mit APEX die ORDIX® Backup GUI (ORBAG) entwickelt, um die Backups von Oracle Datenbanken zentral überwachen und auswerten zu können.

Good to know

Gerade in komplexeren Systemlandschaften mit mehreren hundert Oracle Datenbanken ist es extrem schwierig, einen Überblick über die erstellten Sicherungen zu behalten.

Wurde wirklich jede Datenbank ordnungsgemäß zu den gewünschten Zeiten gesichert? Wie groß sind die Backup-Volumina? Reicht der beschaffte Storage auch im nächsten Jahr für die Speicherung der Backups

aus? Entsprechen die erstellten Backups den Anforderungen der SLA?

Die APEX-Anwendung ORBAG von ORDIX® unterstützt Sie dabei diese Fragen zu beantworten.

Voraussetzung für ORBAG

Zum Betrieb der Backup GUI wird eine Oracle Datenbank mit einem installierten APEX Framework der Version 4.0 (oder höher) benötigt. Darüber hinaus setzt die Anwendung voraus, dass die Backup-Informationen in einem Oracle Recovery-Katalog zentral verwaltet werden. Für die Benutzung und Konfiguration von ORBAG wird lediglich ein aktueller Browser benötigt.

Backup Reporting leicht gemacht

ORBAG ermöglicht es, viele Informationen aus dem RMAN-Katalog auszulesen und aufzubereiten. Um den Datentransfer zwischen dem RMAN-Katalog und der APEX-Datenbank zu verringern, werden in konfigurierbaren Abständen die notwendigen Daten des RMAN-Katalogs als Materialized Views im APEX-Schema abgelegt.

Diese Views bilden die Grundlage für das Reporting. Die Ausgabe erfolgt sowohl in Tabellen- als auch in Diagrammform (siehe Abbildung 1). Hierzu gehören zum Beispiel

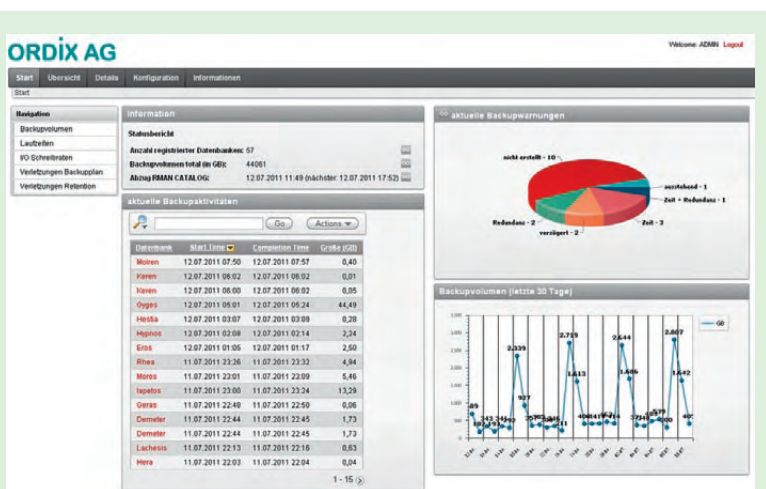


Abb. 1: Startseite der ORDIX Backup GUI.

Auswertungen hinsichtlich des gesamten Backup-Volumens, sowie des Anteils einzelner Sicherungen am Gesamtvolumen. Darüberhinaus werden bspw. auch Backup-Laufzeiten und ihre prognostizierte Entwicklung aufgezeigt.

Vertrauen ist gut ...

Bevor Backups überwacht werden können, müssen sie zunächst geplant werden. Dazu stellt ORBAG eine Konfigurationsvariante bereit, die es ermöglicht Sicherungen dediziert zu definieren. Der Anwender gibt hierzu die Datenbank, den Sicherungstyp sowie den Zeitpunkt der Sicherung an (Abbildung 2).

Diese Planung stößt kein neues Backup an, sondern dient lediglich der Überwachung von auszuführenden Backups. Auf Wunsch können diese Zeitpläne auch aus den Daten eines bestehenden Recovery-Katalogs extrahiert werden.


Anhand des Backup-Plans und der Daten aus dem RMAN-Katalog überprüft ORBAG, ob ein geplantes Backup auch tatsächlich durchgeführt wurde. Dabei unterscheidet die Applikation zwischen verschiedenen Zuständen, wie z.B. „planmäßig“, „leicht verzögert“, „stark verzögert“ oder „kein Backup“ (siehe Abbildung 3). Die Schwellwerte für die einzelnen Warnstufen können innerhalb der GUI konfiguriert werden.

... Kontrolle ist besser

Ein weiterer Teil des Monitoring von ORBAG besteht in der Überprüfung der Retentions. Hierbei können sowohl Zeiträume, für die Sicherungen vorgehalten werden sollen, als auch die Anzahl der vorzuhaltenden Backups konfiguriert werden. Diese Einstellungen können global vorgenommen oder auch für einzelne Datenbanken separat eingestellt werden.

Anhand dieser Konfiguration überprüft die Anwendung, ob die Zeiträume oder Redundanzen auch tatsächlich eingehalten werden können.

Sollte ORBAG einmal einen Fehler oder eine Warnung feststellen, so wird diese Information sofort an den zuständigen Administrator und an den Ansprechpartner der Datenbank (z.B. Fachabteilung) weitergeleitet, damit diese zeitnah eingreifen können. Diese Weiterleitung erfolgt in Form von Reports und / oder Mails.



Datenbank	Backup Typ	Tag	Zeit
Ares	Full	Donnerstag	23:00
Ares	Full	Montag	22:00
Arges	Full	Dienstag	19:00
Atropos	Full	Montag	21:45
Briareos	Full	Montag	04:00
Brontes	Full	Dienstag	21:30
Brontes	Full	Dienstag	21:45
Chaos	Full	Dienstag	22:30
Demeter	Incremental	Mittwoch	22:45
Demeter	Full	Dienstag	03:00
Eris	Full	Montag	21:30

Abb. 2: Backup-Plan.

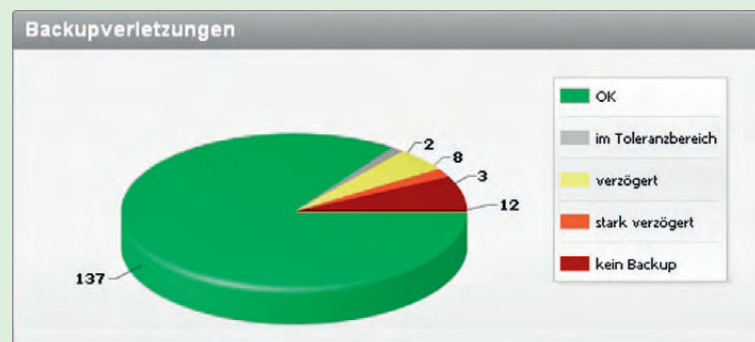


Abb. 3: Grafische Auswertung des Backup-Plans.

Systeminformationen

Eine weitere praktische Funktion bietet ORBAG mit der Ein- und Ausgabe von Systeminformationen. Hierzu gehören zum Beispiel Informationen über die Server (wie die IP-Adresse, das Serverbetriebssystem) oder Informationen zur Datenbankinstanz (z.B. Datenbankversion).

Wurden die Systeminformationen einmal konfiguriert, gibt die GUI sie an zentralen Stellen

Glossar

GUI	Graphical User Interface - Grafische Benutzeroberfläche.
Materialized View	Spezielle Sichten auf Tabellen, deren Inhalt temporär physikalisch gespeichert wird, um den Aufwand zur Berechnung zu minimieren.
RMAN	Recovery Manager - Oracle Werkzeug zur Sicherung und Wiederherstellung von Datenbanken (Backup, Restore, Recovery und Konvertierung).
Recovery-Katalog	Zentrales Repository für die Backup-Daten des RMAN.
SLA	Service Level Agreement - Bezeichnet einen Vertrag über die Art und Güte (Dienstgütevereinbarung) der zu erbringenden Leistung zwischen Auftraggeber und Auftragnehmer.

Fazit

Die ORDIX® Backup GUI bietet eine komfortable Möglichkeit zur Überwachung und Auswertung von Backups, denen ein RMAN mit Katalog zugrunde liegt.

Sowohl die Überwachung als auch die Konfiguration erfolgen hierbei intuitiv und unkompliziert über einen Browser. Sie möchten mehr über dieses Thema erfahren, dann nehmen Sie Kontakt zu uns auf. Wir helfen Ihnen gerne weiter.

aus, um den Administratoren lange Sucharbeiten zu ersparen.

Diese Informationen dienen weiterhin dazu, Datenbanken nach ihren Versionen oder Betriebssystemen zu gruppieren, um dementsprechende Statistiken (z.B. das Backup-Volumen eines gesamten Server) ausgeben zu können.



Patrick Hecker
(info@ordix.de)

Seminarempfehlung: Oracle APEX-Programmierung

► **Informationen/Online-Anmeldung:** <http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Datenbanken/nr/1010/index.html>

In diesem Seminar erhalten Sie einen Einblick in die Funktionen von APEX (ORACLE Application Express). Oracle Application Express ist eine vollständige, in die Oracle-Datenbank integrierte Entwicklungs- und Laufzeitumgebung für Webanwendungen. Mit einem normalen Webbrowser werden Anwendungen schnell und einfach entwickelt. In einem Wechsel von Theorie und Praxis lernen Sie, wie Anwendungen aufgebaut, entwickelt und verwaltet werden können. Am Ende des Kurses sind Sie in der Lage selbstständig APEX-Anwendungen zu erstellen.

Seminarinhalte

- APEX-Architektur
- Websheet Application
- Database Applications
 - Pages, ITEMS und Regions
 - Variablen, Links und Drilldown-Funktionalitäten
 - Prozesse und Dynamic Actions
 - Shared Components
 - Formulare
- Benutzerverwaltung
- User-Authentifizierung
- Workspace-Administration
- Anbindung / Nutzung des SQL Developer
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

- 19.09. - 21.09.2011 in Bielefeld
- 24.10. - 26.10.2011 in Wiesbaden
- 12.12. - 14.12.2011 in Wiesbaden

Seminar-ID: DB-ORA-43

Dauer: 3 Tage

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

Frühbucherpreis: 1.161,00 € (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>.

Eine Alternative zu den Skriptsprachen Perl und Shell in DB2

DB2 Administration mit REXX

Typischerweise werden Administrationsskripte für DB2 Datenbanken auf den Plattformen Linux, Unix und Windows meist in den Skriptsprachen Perl oder Shell (oft im Verbund mit Sed und Awk) geschrieben. Ein kleines Nischendasein hat hier die Skriptsprache REXX. Zu recht oder zu unrecht?

Dieser Artikel richtet sich an Administratoren, die eine Alternative zu Perl oder Shell suchen.

Historie

REXX steht für „REstructured eXtended eXecutor“ (Language) und wurde Anfang der 1980er Jahren von IBM entwickelt. Sie fand vor allem auf den Host-Systemen Zuspruch und ist dort auch heute noch eine der meist eingesetzten Skriptsprachen. Die Programmiersprache ist eine Interpretersprache, wobei mittlerweile auch für einzelne Betriebssysteme native Compiler verfügbar sind. Zunächst wurde REXX von IBM nur kommerziell vertrieben, ist aber seit 2004 auf verschiedenen Windows, Linux und Unix Plattformen als Open Source unter dem Namen ooRexx (Open Object REXX) verfügbar.

Inzwischen existieren aber auch zahlreiche weitere REXX-Implementierungen, wobei Regina REXX und ooRexx die beiden am weitesten verbreiteten Derivate sind.

Sprachumfang

Eine Auflistung der von REXX unterstützten Befehle würde den Rahmen des Artikels sprengen. Daher nur so viel: Alle üblichen und notwendigen Sprachelemente wie Schleifen, If-Then-Else-Konstrukte etc. sind enthalten. Ebenso sind Arrays, diverse Build-In-Funktionen und das Handling von Dateizugriffen implementiert. Sogar Funktionalitäten um in Windows Fenstertechniken zu nutzen, z.B. um kleine GUI Tools zu schreiben, sind verfügbar.

Datenbank API

Um mit DB2 Datenbanken arbeiten zu können, existieren drei API welche zunächst registriert werden müssen (siehe Abbildung 1 + 2). Die API `SQLEXEC` ist für SQL-Anweisungen, `SQLDBS` und `SQLDB2` für Datenbankkommandos. Alle API werden grundsätz-

```
rcy = SysAddFuncPkg("db2rex")
If rcy \= 0 then
do
say 'DB2 APIs konnten nicht registriert werden'
exit -1
end
```

Abb. 1: DB2 API-Registrierung in Unix.

```
/* Registriere SQLDBS, SQLEXEC und SQLDB2 */
rcy = RxFuncAdd( 'SQLDBS', 'db2ar', 'SQLDBS' )
If rcy \= 0 then
do
say 'SQLDBS konnte nicht im REXX registriert werden'
exit -1
end
rcy = RxFuncAdd( 'SQLEXEC', 'db2ar', 'SQLEXEC' )
If rcy \= 0 then
do
say 'SQLEXEC konnte nicht im REXX registriert werden'
exit -1
end
rcy = RxFuncAdd( 'SQLDB2', 'db2ar', 'SQLDB2' )
If rcy \= 0 then
do
say 'SQLDB2 konnte nicht im REXX registriert werden'
exit -1
end
```

Abb. 2: DB2 API-Registrierung in Windows.

```
CALL SQLDBS '<Kommando>'
CALL SQLDB2 '<Kommando>'
CALL SQLEXEC '<Kommando>'
```

Abb. 3: Aufruf von DB2 API in REXX.

```
/* Connect zur Datenbank */
CALL SQLEXEC 'CONNECT TO DBADTEST'
CALL CHECKERR
```

Abb. 4: Connect mit der SQLEXEC-API.

```
/* Fehlerpruefung */
CHECKERR:
  if ( SQLCA.SQLCODE = 0 ) then
    return 0
  else do
    say 'FEHLER!!!!'
    say '--> SQLCODE: ' SQLCA.SQLCODE
    call SQLDBS 'GET MESSAGE INTO :errmsg'
    say errmsg
    if (SQLCA.SQLCODE < 0 ) then do
      CALL SQLEXEC 'CONNECT RESET'
      say 'Skript mit Fehler beendet'
      exit 8
    end
  else do
    say 'WARNUNG - Programm wird mit Fehlern fortgesetzt'
    SQLCA.SQLCODE = 0
    return 0
  end
end
```

Abb. 5: Fehlerbehandlung.

```
/* Erzeuge Array mit Runstats-Kommandos */
rebindarray=.array~new
i=0
st = "select 'RUNSTATS ON TABLE '",
      "concat rtrim(tabschema) concat ",
      " '.' concat rtrim(tabname)",
      "from syscat.tables",
      "where type='T'",
      "order by tabschema,tabname"
call SQLEXEC 'PREPARE s1 FROM :st'
call CHECKERR
call SQLEXEC 'DECLARE c1 CURSOR FOR s1'
call CHECKERR
call SQLEXEC 'OPEN c1'
do while ( SQLCA.SQLCODE = 0 )
  call SQLEXEC 'FETCH c1 INTO :db2_stmt'
  if (SQLCA.SQLCODE = 0) then do
    i=i+1
    rebindarray~put(db2_stmt,i)
  end
end
call SQLEXEC 'CLOSE c1'
call CHECKERR
```

Abb. 6: Selektieren von Daten.

lich mit einem Kommando `CALL` aufgerufen (siehe Abbildung 3).

Zwischen SQL-Anweisungen und Kommandos muss nicht nur bei der Verwendung der richtigen API unterschieden werden, sondern auch beim Verbinden zur Datenbank. D.h. für SQL-Anweisungen muss der `Connect` über die `SQLEXEC`-API geschehen, bei Kommandos über die `SQLDB2`-API (siehe Abbildung 4).

Fehlerprüfung

Nach jedem Datenbankbefehl wird der SQL-Fehlercode in der Variable `SQLCA.SQLCODE` gespeichert. Wobei der Code gleich Null bei einer erfolgreichen Ausführung, größer Null bei Warnungen und kleiner Null bei allen Fehlern ist. Die dazugehörige Fehlermeldung kann mit dem Befehl `Get error message` in eine Variable eingelesen werden.

Sinnvollerweise sollte in jedem REXX-Skript eine zentrale Funktion für das Fehlerbehandlung eingebaut werden (siehe Abbildung 5).

SQL-Anweisungen

Das Lesen von Daten aus einer Tabelle oder View erfolgt mit dynamischem SQL, d.h. die SQL-Anweisung muss zunächst mit übersetzt werden. Anschließend können über einen Cursor die einzelnen Datensätze ausgelesen werden (siehe Abbildung 6).

Ähnlich verhält es sich bei Insert-, Update- und Delete-Anweisungen. Auch diese müssen zunächst übersetzt werden (Kommando `PREPARE`) und können danach ausgeführt werden (Kommando `EXECUTE`). Alternativ kann auch direkt das Kommando `EXECUTE-IMMEDIATE` verwendet werden (Abbildung 7).

Datenbankkommandos

Datenbankkommandos sind die Befehle, welche in erster Linie für eine Administration von DB2 Datenbanken notwendig sind. Zwar können mittlerweile viele Kommandos über die Prozedur `ADMIN_CMD` auch als SQL Statement ausgeführt werden, dennoch kommt man um das Ausführen von Kommandos meist nicht herum.

Die Kommandos müssen, wie bereits erwähnt, mit der `SQLDBS`- oder `SQLDB2`-API ausgeführt werden. Welche API für welchen Befehl zu verwenden ist, kann der DB2

Dokumentation entnommen werden. Ein Beispiel zeigt Abbildung 8.

Fazit

REXX ist eine leicht zu erlernende Skriptsprache, welche von DB2 unterstützt wird. Insbesondere wenn die Sprache schon z.B. auf z/OS-Systemen existiert, ist die Sprache eine durchaus überlegenswerte Alternative zu Perl und Shell.

Einzig das äußerst dürftige Literaturangebot, sowohl in gedruckter Form als auch im Web, ist als einziger Negativpunkt anzumerken.



Michael Spoden
(info@ordix.de).

```
/* Insert in dba Tabelle */
st = "INSERT INTO dba_jobs VALUES (current timestamp, 'Runstats')"
call SQRXEC 'EXECUTE IMMEDIATE :st'
call CHECKERR
/* Connect Reset */
call SQRXEC 'CONNECT RESET'
call CHECKERR
exit 0 /* ENDE HAUPTPROGRAMM */
```

Abb. 7: Einfügen von Daten.

```
/* Fuehre Runstats aus */
CALL SQLDB2 'CONNECT TO DBADTEST'
loop statement over rebindarray
  say 'Fuehre Runstats durch:' statement
  CALL SQLDB2 statement
  call CHECKERR
end
```

Abb. 8: Benutzung der SQLDB2-API.

Glossar

API Application Programming Interface - Sammlung von Klassen, welche eine Schnittstelle zu einer Hardware oder Anwendung definieren.

Links

- ▶ [1] Open Object REXX-Homepage: <http://www.oorexx.org>
- ▶ [2] Homepage der REXX Language Association: <http://www.rexxla.org>
- ▶ [3] REXX-Beispiele im DB2 Information Center: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.apdv.samptop.doc/doc/r0008934.html>
- ▶ [4] Aufruf von DB2 API in REXX (DB2 Information Center): <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.apdv.api.doc/doc/c0006728.html>

Seminarempfehlung: IBM DB2 für Linux/Unix/Windows Administration

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

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

Seminarinhalte

- Überblick über die DB2 Architektur
- Einführung in die GUI Tools (Control Center, CLP)
- DB2 Installation, Anlegen einer Datenbank
- Datenbankobjekte verwalten | Backup und Recovery
- Maintenance Tools wie Reorganisation, Runstats, Reorgcheck usw.
- Sperrmechanismen (Locking und Monitoring)
- Security: Benutzerverwaltung, Zugriffsschutz
- Datenbank-Tools wie Load, Import und Export Utility, Tipps und Tricks
- Vertiefung der Theorie durch praktische Übungen und Beispiele

Termine

28.11. - 02.12.2011 in Wiesbaden
 27.02. - 02.03.2012 in Wiesbaden
 07.05. - 11.05.2012 in Wiesbaden
 13.08. - 17.08.2012 in Wiesbaden

Seminar-ID: DB-DB2-02

Dauer: 5 Tage

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

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

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

Datenbanken				September 2011 - Februar 2012	
DB-DB-02	Datenbank-Hochverfügbarkeitslösungen für Entscheider	1 Tag	550,00 €	18.10.2011	
DB-DB-03	Datawarehouse Grundlagen	3 Tage	1.290,00 €	09.01.2012	
DB-DB-01	Datenbank-Modellierung	2 Tage	890,00 €	24.10.2011 09.01.2012	
DB-ORA-01	Oracle SQL	5 Tage	1.890,00 €	10.10.2011 28.11.2011 30.01.2012 06.02.2012	
DB-ORA-01A	Oracle SQL für Experten	3 Tage	1.290,00 €	07.11.2011 27.02.2012	
DB-ORA-02	Oracle Datenbankprogrammierung mit PL/SQL Grundlagen	5 Tage	1.890,00 €	17.10.2011 21.11.2011 06.02.2012	
DB-ORA-34	Oracle Datenbankprogrammierung mit PL/SQL Aufbau	5 Tage	1.890,00 €	24.10.2011 05.12.2011 20.02.2012	
DB-ORA-42	Oracle PL/SQL Tuning	3 Tage	1.890,00 €	26.09.2011 07.11.2011 21.11.2011	
DB-ORA-44	Entwicklung von Web-Anwendungen mit Oracle PL/SQL	2 Tage	1.090,00 €	27.10.2011 23.02.2012	
DB-ORA-43	Oracle APEX Programmierung	3 Tage	1.890,00 €	24.10.2011 12.12.2011 30.01.2012 13.02.2012	
DB-ORA-38	Objektorientierung in Oracle	5 Tage	1.990,00 €	24.10.2011 30.01.2012	
DB-ORA-03	Oracle Datenbankadministration Grundlagen	5 Tage	1.990,00 €	24.10.2011 12.12.2011 13.02.2012	
DB-ORA-04	Oracle Datenbankadministration Aufbau	5 Tage	1.990,00 €	28.11.2011 23.01.2012	
DB-ORA-06	Manuelles Oracle Backup und Recovery & weitere Konzepte	5 Tage	1.990,00 €	10.10.2011 28.11.2011	
DB-ORA-32	Oracle Backup und Recovery mit RMAN	5 Tage	1.990,00 €	21.11.2011 27.02.2011	
DB-ORA-07	Oracle Tuning und Monitoring	5 Tage	1.990,00 €	10.10.2011 05.12.2011	
DB-ORA-41	Oracle AWR und ASH Analyse und Interpretation	3 Tage	1.290,00 €	02.11.2011	
DB-ORA-11	Oracle Troubleshooting Workshop	5 Tage	1.990,00 €	10.10.2011	
DB-ORA-08	Oracle 11gR2 RAC und Grid Infrastructure	5 Tage	1.990,00 €	26.09.2011 14.11.2011 23.01.2012 20.02.2012	
DB-ORA-15	Oracle 11g Neuheiten	5 Tage	1.990,00 €	17.10.2011 09.01.2012 20.02.2012	
DB-ORA-33A	Oracle Security	4 Tage	1.590,00 €	04.10.2011 21.11.2011 06.02.2012	
DB-ORA-31	Oracle Data Guard	4 Tage	1.590,00 €	04.10.2011 07.11.2011 13.02.2012	
DB-ORA-35	Oracle Grid Control	3 Tage	1.290,00 €	14.11.2011	
DB-ORA-36	Oracle Replikation	3 Tage	1.290,00 €	auf Anfrage	
DB-ORA-37	Oracle Streams Advanced Queuing	3 Tage	1.290,00 €	17.10.2011 16.01.2012	
DB-ORA-39	Oracle Migration und Patching	3 Tage	1.290,00 €	14.11.2011	
DB-ORA-40	Oracle Capacity Planning	3 Tage	1.290,00 €	auf Anfrage	
DB-INF-01	IBM Informix SQL	5 Tage	1.790,00 €	07.11.2011 13.02.2012	
DB-INF-02	IBM Informix Administration	5 Tage	1.990,00 €	14.11.2011 27.02.2012	
DB-INF-04	IBM Informix Dynamic Server Backup und Recovery	3 Tage	1.290,00 €	05.12.2011	
DB-INF-03	IBM Informix Tuning und Monitoring	5 Tage	1.990,00 €	26.09.2011	
DB-INF-07	IBM Informix Hochverfügbarkeits-Technologien unter Unix	4 Tage	1.590,00 €	28.11.2011	
DB-DB2-01	IBM DB2 für Linux/Unix/Windows SQL Grundlagen	5 Tage	1.890,00 €	07.11.2011 20.02.2012	
DB-DB2-02	IBM DB2 für Linux/Unix/Windows Administration	5 Tage	1.990,00 €	28.11.2011 27.02.2012	
DB-DB2-05	IBM DB2 für Linux/Unix/Windows Monitoring und Tuning	3 Tage	1.290,00 €	26.09.2011	
DB-MY-01	MySQL Administration	3 Tage	1.190,00 €	02.11.2011 13.02.2012	
DB-MSSQL-1	Microsoft SQL Server Administration	5 Tage	1.790,00 €	12.12.2011	
Entwicklung					
E-UML-01	Anforderungsanalyse mit UML	3 Tage	1.190,00 €	30.01.2012	
E-SWA-01	Softwarearchitekturen	5 Tage	1.890,00 €	auf Anfrage	
P-PHP-01	PHP Programmierung Grundlagen	5 Tage	1.690,00 €	24.10.2011 16.01.2012	
P-PHP-02	PHP Programmierung Aufbau	3 Tage	1.190,00 €	07.11.2011 23.01.2012	
P-PERL-01	Perl Programmierung Grundlagen	5 Tage	1.690,00 €	10.10.2011 30.01.2012	
P-PERL-02	Perl Programmierung Aufbau	5 Tage	1.690,00 €	17.10.2011 05.12.2011 13.02.2012	
P-UNIX-01	Shell Awk und Sed	5 Tage	1.690,00 €	26.09.2011 21.11.2011 06.02.2012	
P-UNIX-01A	Awk Intensiv-Workshop	3 Tage	1.190,00 €	10.10.2011 09.01.2012	
P-XML-01	Einführung in XML	3 Tage	1.190,00 €	auf Anfrage	
P-XML-02	XML Programmierung unter Java mit DOM und SAX	2 Tage	890,00 €	auf Anfrage	
P-XML-03	Oracle und XML	3 Tage	1.190,00 €	auf Anfrage	
Web- und Applikations-Server					
INT-04	Apache Web-Server Installation und Administration	3 Tage	1.190,00 €	26.09.2011 02.11.2011 23.01.2012 27.02.2012	
INT-07	Tomcat Konfiguration und Administration	3 Tage	1.190,00 €	04.10.2011 05.12.2011 27.02.2012	
INT-08	WebSphere Application Server Installation und Administration	3 Tage	1.390,00 €	21.11.2011	
INT-11	Administration und Konfiguration für JBoss	3 Tage	1.190,00 €	02.11.2011 12.12.2011 20.02.2012	

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.

Online-Anmeldung, aktuelle Seminarinhalte und Termine unter:

<http://training.ordix.de>

Systemmanagement				September 2011 - Februar 2012	
SM-NAG-01	Systemüberwachung mit Nagios Grundlagen	3 Tage	1.190,00 €	21.11.2011	
SM-NAG-01	Systemüberwachung mit Nagios Aufbau	2 Tage	890,00 €	24.11.2011	
Java-JEE					
OO-01	Einführung in die Objektorientierte Programmierung	3 Tage	1.190,00 €	10.10.2011 09.01.2012	
P-JAVA-01	Java Programmierung Grundlagen	5 Tage	1.690,00 €	17.10.2011 23.01.2012	
P-JAVA-03	Java Programmierung Aufbau	5 Tage	1.690,00 €	14.11.2011 06.02.2012	
P-JAVA-02	Java GUI Entwicklung mit Swing	5 Tage	1.690,00 €	05.12.2011 13.02.2012	
P-JEE-01	JEE für Entscheider	1 Tag	590,00 €	24.10.2011 16.01.2012	
P-JEE-02	Einführung in JEE	3 Tage	1.290,00 €	25.10.2011 17.01.2012	
P-JEE-03	JSP und Servlet Programmierung	5 Tage	1.590,00 €	21.11.2011	
P-JEE-04	EJB Programmierung	5 Tage	1.590,00 €	07.11.2011 20.02.2012	
P-JEE-05	Web-Anwendungen mit JavaServer Faces (JSF)	5 Tage	1.590,00 €	28.11.2011 27.02.2012	
P-JEE-06	Entwickeln mit dem Spring-Framework	3 Tage	1.190,00 €	17.10.2011 12.12.2011 16.01.2012	
INT-05	Java Web Services	3 Tage	1.190,00 €	04.10.2011 09.01.2012	
J-HIB-01	Hibernate und die Java Persistence API	5 Tage	1.690,00 €	10.10.2011 23.01.2012	
P-JEE-08	Java Performance Tuning	3 Tage	1.290,00 €	04.10.2011 16.01.2012	
Betriebssysteme					
BS-01	Unix/Linux Grundlagen für Einsteiger	5 Tage	1.690,00 €	14.11.2011 16.01.2012 20.02.2012	
BS-25	Unix AufbauSeminar für Datenbank- und Applikationsbetreuer	5 Tage	1.790,00 €	06.02.2012	
BS-02	Linux Systemadministration	5 Tage	1.690,00 €	10.10.2011 12.12.2011 23.01.2012	
BS-09	Linux Hochverfügbarkeits-Cluster	3 Tage	1.290,00 €	17.10.2011 05.12.2011 06.02.2012	
BS-19	Linux Cluster mit Pacemaker und Heartbeat 3	3 Tage	1.190,00 €	auf Anfrage	
BS-16	OpenLDAP - Praxiseinsatz im Netzwerk	4 Tage	1.490,00 €	04.10.2011	
BS-03	Solaris Systemadministration Teil I	5 Tage	1.990,00 €	21.11.2011 30.01.2012	
BS-04	Solaris Systemadministration Teil II	5 Tage	1.990,00 €	26.09.2011 28.11.2011 20.02.2012	
BS-06	Solaris kompakt für erfahrene Unix/Linux-Umsteiger	5 Tage	1.990,00 €	24.10.2011	
BS-24	Solaris 11 Administration Neuheiten	5 Tage	1.990,00 €	auf Anfrage	
BS-18	Solaris Virtualisierung mit ZFS und Container (Zonen)	5 Tage	1.990,00 €	10.10.2011 12.12.2011	
BS-23	Solaris Virtualisierung mit LDOM	3 Tage	1.290,00 €	17.10.2011 09.01.2012	
AIX-01	IBM AIX Systemadministration Grundlagen	5 Tage	1.990,00 €	14.11.2011 27.02.2012	
AIX-02	IBM AIX Installation, Backup und Recovery mit NIM	3 Tage	1.290,00 €	04.10.2011 05.12.2011	
AIX-03	Analyse v. komplexen AIX Performance Problemen	5 Tage	2.300,00 €	17.10.2011 30.01.2012	
Projektmanagement					
PM-01	IT-Projektmanagement - Methoden und Techniken	5 Tage	1.990,00 €	10.10.2011 16.01.2012	
PM-06	System. Projektmanagement - Projektteams souverän führen	4 Tage	1.850,00 €	22.11.2011 20.02.2012	
PM-08	Agiles Projektmanagement mit SCRUM	2 Tage	1.100,00 €	03.11.2011	
PM-09	Capability Maturity Model Integration (CMMI)	2 Tage	1.100,00 €	07.11.2011	
PM-10	Kennzahlen der IT	2 Tage	1.100,00 €	17.10.2011 09.02.2012	
PM-07	Krisenmanagement in Projekten	2 Tage	1.100,00 €	19.10.2011 30.01.2012	
PM-05	IT-Projektcontrolling	3 Tage	1.290,00 €	04.10.2011 23.01.2012	
PM-11	Konfliktmanagement	2 Tage	1.100,00 €	29.11.2011 02.02.2012	
IT-Management					
MGM-02	IT-Architekturen	3 Tage	1.650,00 €	16.11.2011 13.02.2012	
MGM-01	E-Business	2 Tage	1.100,00 €	14.11.2011 16.02.2012	
MGM-07	IT-Strategien	2 Tage	1.100,00 €	09.11.2011	
MGM-08	Strategien effizient entwickeln	3 Tage	1.650,00 €	11.01.2012	
MGM-03	IT-Management	5 Tage	1.990,00 €	24.10.2011 05.12.2011	
MGM-05	IT-Risikomanagement	3 Tage	1.650,00 €	26.09.2011 28.11.2011 27.02.2012	
MGM-04	IT-Prozessmanagement	3 Tage	1.650,00 €	04.10.2011 06.02.2012	

Zentrale: **ORDIX AG**
 Westermauer 12 - 16
 33098 Paderborn
 Tel.: 05251 1063-0

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

Unsere Seminarstandorte sind: Wiesbaden, Bielefeld und Hannover.
 Die Preise gelten pro Seminar pro Teilnehmer in Euro
 zzgl. ges. MwSt., Inhouse-Preise auf Anfrage.



Weld: Der Java Einspritzmotor

Dieser Artikel richtet sich an Java Entwickler, die ihre Programme flexibler, wiederverwendbarer und testbarer gestalten wollen.

Kennen Sie diese Situation? Sie suchen nach einem Framework, einer Library oder einem Tool und werden im Internet fündig. Alles funktioniert so, wie Sie es sich wünschen - naja, fast so wie Sie es sich wünschen, denn ein klitzekleines Detail entspricht nicht Ihren Vorstellungen. Für die notwendigen Anpassungen müssen Sie jedoch eine existierende Klasse austauschen, in den gefundenen Sourcen stehen explizite `new`-Aufrufe und die existierende Klasse wird direkt referenziert. Somit sind die Anpassungen nur sehr schwer möglich. Dabei wäre alles so einfach gewesen, wenn die Entwickler statt einer direkten Klassenreferenz ein Interface verwendet hätten und statt expliziter `new`-Aufrufe ein Framework für eine Dependency Injection. Weld ist ein solches Framework, das den aktuellen Standard (JSR-299 - Contexts and Dependency Injection) definiert, und sogar in SE-Applikationen verwendbar ist. Wie man es einsetzt wird in diesem Artikel genauer beschrieben.

Stellen wir uns vor, wir haben die Aufgabe, den Login-Dialog für eine Client-Applikation zu entwickeln, wobei der eingegebene Benutzername und das eingegebene Passwort von einem Server überprüft werden sollen. Den Server-Aufruf lagern wir in eine separate Klasse aus (siehe Abbildung 1).

Im Client wird die Dialogsteuerung für den Login-Dialog von einem Controller durchgeführt. Dieser Controller besitzt auch die `LoginServerConnection`, um die Zugangsbeurteilung zu überprüfen (siehe Abbildung 2).

Testbarkeit

Aber wie sollen nun automatische Tests des `LoginController` funktionieren? Mit der dargestellten Klasse würden diese Tests mit dem Server kommunizieren. Somit müssten in der Datenbank des Server die User-Objekte für das Login in einem Zustand vorliegen, wie es der durchzuführende Test erwartet. Automatische Tests unter diesen Voraussetzungen sind daher sehr kompliziert und nur sehr aufwändig zu erstellen.

Die Lösung hierzu sind sogenannte Mock-Implementierungen. Hierunter werden Implementierungen verstanden, die nur vorgeben etwas zu tun.

Grundvoraussetzung für den Einsatz von Mock-Implementierungen ist die Verwendung eines entsprechenden Interface (siehe Abbildung 3).

Unsere Klasse benennen wird in `LoginServerConnectionImpl` um und lassen sie das neugeschaffene Interface implementieren. Die Mock-Implementierung könnte beispielsweise wie in Abbildung 4 dargestellt aussehen.

Erweiterbarkeit

Die Existenz von zwei Implementierungen des Interface erfordert im Source eine Fallunterscheidung (siehe Abbildung 5).

Dieser unschöne Source Code kann natürlich noch in einer Factory gekapselt und somit schön versteckt werden. Doch auch dieses Vorgehen lässt die Abhängigkeit des Programms von der Mock-Implementierung nicht verschwinden. Die Mock-Implementierung, die eigentlich nur für unsere Tests gedacht ist, schleicht sich so klammheimlich in das produktive System ein. Das schadet zwar nicht, ist aber auch nicht sehr schön und bläht den Client unnützlich auf.

Wiederverwendbarkeit

Die oben dargestellte Fallunterscheidung verhindert bzw. erschwert auch eine mögliche Wiederverwendung des `LoginController` in ähnlichen Programmen, denn die Fallunterscheidung müsste für jede Wiederverwendung erweitert werden. Somit werden alle Programme, die den `LoginController` verwenden, voneinander abhängig, was allerdings nicht gewollt ist. Letztendlich liefe eine Wiederverwendung auf Copy-Paste und Anpassung hinaus.

Die Lösung

Diese Probleme sind nicht neu und die allgemein akzeptierte Lösung hierzu heißt Dependency Injection. Bis vor kurzem gab es keinen allgemeinen Standard für eine Realisierung dieses Prinzips. Dies führte zu mehreren, voneinander unabhängigen Implementierungen (z.B. hivemind, spring, guice). Doch

```
public class LoginServerConnection{
    public boolean login(String userName, String password){...}
}
```

Abb. 1: Aufruf des Server.

```
public class LoginController{
    private LoginServerConnection connection;
    ...
    public LoginController(){
        connection = new LoginServerConnection();
        ...
    }
    ...
}
```

Abb. 2: Controller zur Steuerung des Login-Dialogs.

```
public Interface LoginServerConnection{
    public boolean login(String userName, String password);
}
```

Abb. 3: Interface für Mock-Implementierungen.

```
public class LoginServerConnectionMock
    implements LoginServerConnection{
    public boolean login(String userName, String password){
        return "john".equals(userName) && "doe".equals(password);
    }
}
```

Abb. 4: Beispiel für eine Mock-Implementierung.

```
public class LoginController{
    private LoginServerConnection connection;
    ...
    public LoginController(){
        if (testMode){
            connection = new LoginServerConnectionMock();
        }else{
            connection = new LoginServerConnectionImpl();
        }
        ...
    }
    ...
}
```

Abb. 5: Fallunterscheidung bei zwei Implementierungen.

mit dem JSR-299 gibt es seit fast zwei Jahren eine Vorgabe für einen Standard. Leider steht schon im Titel dieser Spezifikation, dass sie nur für eine EE-Umgebung vorgesehen ist. Obwohl Weld die Referenzimplementierung

```
public class LoginController{
    @Inject private LoginServerConnection connection;
    ...
    public LoginController(){
        ...
    }
    ...
}
```

Abb. 6: Implementierung des LoginController mit Weld.

```
@ApplicationScoped
public class TestClient {
    ...
    public void loginAndProceed(
        @Observes ContainerInitialized init) {
        if (controller.doLogin()){
            ...
        }
    }
}
```

Abb. 7: Start der Client-Applikation.

```
<beans>
  <alternatives>
    <class>de.ordix.weld.LoginServerConnectionMock</class>
  </alternatives>
</beans>
```

Abb. 8: Eintragen der Mock-Implementierung.

dieser Spezifikation ist, lässt sie glücklicherweise auch die Verwendung in Nicht-EE-Applikationen zu, was das Einsatzgebiet dieses Framework beträchtlich erweitert.

In Verbindung mit dem Thema Dependency Injection und Java gibt es noch den JSR-330. In diesem werden nur die grundlegenden Annotationen definiert, während JSR-299 weit darüber hinaus geht und den JSR-330 als Basis verwendet.

Die Lösung mit Weld

Schauen wir uns nun die Implementierung des `LoginController` mit Hilfe von Weld an (siehe Abbildung 6).

Die Annotation `@Inject` weist Weld an, die Instanzvariable `connection` mit der vorgesehenen Implementierung zu belegen. Was

die vorgesehene Implementierung ist, kann durch die Konfiguration festgelegt werden. Sämtliche Abhängigkeiten zu den Implementierungsklassen sind nun verschwunden, verwendet wird hier nur noch das Interface.

Um die Instanzvariable (hier auch Injection Point genannt) mit dem konkreten Wert zu belegen, kennt Weld neben der Injection einer Instanzvariablen noch folgende Möglichkeiten:

- Injection der Parameter des Konstruktors:


```
@Inject
public LoginController(
    LoginServerConnection connection){
    this.connection = connection;
}
```
- Injection der Parameter einer Initialisierungsmethode:


```
@Inject
public setConnection(
    LoginServerConnection connection){
    this.connection = connection;
}
```
- Injection durch Produktionsmethoden:


```
@Produces
public LoginServerConnection
createConnection(){
    return new LoginServerConnectionImpl();
}
```

Die andere Seite

Wo eine Injection stattfinden soll, wird durch die `@Inject`-Annotation festgelegt. Was injiziert werden kann, wird durch eine der folgenden Annotationen bestimmt:

- `@RequestScoped`
- `@SessionScoped`
- `@ConversationScoped`
- `@ApplicationScoped`
- `@Singleton`
- `@Dependent`

Nur die drei letztgenannten Möglichkeiten können in einer SE-Applikation verwendet werden.

Durch `@ApplicationScoped` `public class LoginServerConnectionImpl` wird Weld mitgeteilt, dass diese Bean in der gesamten Applikation gültig ist. Unter Umständen werden mehrere Instanzen erzeugt. Soll oder darf in der gesamten Applikation nur eine Instanz erzeugt werden, so muss die Annotation `@Singleton` verwendet werden. Die letzte

Möglichkeit (`@Dependent`) definiert, dass je Injection Point eine Instanz erzeugt wird. Wird keine der genannten Annotationen verwendet, so nimmt Weld implizit `@Dependent`.

Ungewöhnlich ist der Start der Client-Applikation. Normalerweise würde man eine Methode `main` schreiben, doch im Zusammenhang mit Weld muss die Startklasse mit `@ApplicationScoped` annotiert sein. Die Startmethode muss dabei den Parameter `@Observes ContainerInitialized` `init` besitzen (siehe Abbildung 7).

Beim Start des Java Programms wird hierbei nicht die Klasse `TestClient` als Startklasse verwendet, sondern die Klasse `org.jboss.weld.environment.se.StartMain` des Weld Framework. Diese sucht nach den benutzerdefinierten Startklassen und ruft die Startmethoden auf.

Alternativen

Ist neben der `Impl`-Klasse noch eine Mock-Klasse vorhanden, so reicht es nicht aus, diese mit `@ApplicationScoped` zu annotieren, denn dann erhält man die Fehlermeldung *ambiguous dependencies*. Eine Lösungsmöglichkeit besteht darin, die Mock-Implementierung zusätzlich mit der Annotation `@Alternative` zu versehen. Im Normalfall wird die Klasse verwendet, die nicht als Alternative gekennzeichnet ist. Erst wenn in der Konfigurationsdatei `META-INF/beans.xml` die Mock-Implementierung eingetragen ist, wird diese auch verwendet (siehe Abbildung 8).

So kann also von außen durch eine Konfiguration das Gesamtsystem an den jeweiligen Anwendungsfall angepasst werden, ohne dass der Source Code verändert werden muss.

Interceptoren

Neben den Möglichkeiten, abhängige Objekte zu injizieren, können mit Weld Interceptoren und Dekoratoren definiert werden. Beides sind recht ähnliche Konzepte wobei wir uns im Folgenden auf die Interceptoren konzentrieren wollen. Weld unterstützt folgende Interceptoren:

- **Bean-Methoden-Interceptoren**
Interceptor-Methoden, die mit `@AroundInvoke` annotiert sind, werden aktiviert, wenn Bean-Methoden aufgerufen werden. Dies wird später noch in einem Beispiel dargestellt.

```
@InterceptorBinding
@Retention(RetentionPolicy.RUNTIME)
@Target({ ElementType.METHOD, ElementType.TYPE })
public @interface Log {}
```

Abb. 9: Definition der Anotation `@Log`.

```
@Interceptor
@Log
public class LogInterceptor {
    @AroundInvoke
    public Object logInvocation(InvocationContext ctx)
        throws Exception {
        long start = System.currentTimeMillis();
        Object result = ctx.proceed();
        System.out.println(ctx.getMethod() + ": "
            + (System.currentTimeMillis() - start) + „ms“);
        return result;
    }
}
```

Abb. 10: Implementierung der Protokollierung.

- **Lifecycle-Callback-Interceptoren**
Diese werden vom Container aufgerufen, wenn eine Bean einen bestimmten Zustand in seinem Lifecycle einnimmt (zum Beispiel `@PostConstruct`).
- **Timeout-Interceptoren**
Diese werden vom EJB-Container im Timeout-Fall aufgerufen.

Sollen in unserem Beispiel bestimmte Bean-Methodenaufrufe protokolliert werden, um z.B. Zeitmessungen durchzuführen, so müssen diese Methoden mit einer selbst zu definierenden Annotation gekennzeichnet werden. In unserem Fall definieren wir eine Annotation namens `@Log` (siehe Abbildung 9).

Diese Annotation dient dazu, alle Bean-Methoden, die geloggt werden sollen, zu markieren. Folgende Zeilen weisen z.B. Weld an, die Methode `login` zu protokollieren:

```
@Log
public boolean login(String
    userName, String password) {...}
```

Wie diese Protokollierung aussieht, definiert der zugehörige Interceptor, der wie in Abbildung 10 implementiert ist.

Wird nun in der Klasse `LoginServerConnection` die Methode `login` aufgerufen,

```

<beans>
  <interceptors>
    <class>de.ordix.weld.LoginInterceptor</class>
  </interceptors>
</beans>

```

Abb. 11: Eintragung der Interceptoren in der META-INF/beans.xml.

Glossar

Annotation	Anmerkungen im Java Source Code, die zur Laufzeit des Programms ausgewertet werden können. Annotations sind seit Java 5 Bestandteil von Java. Als geistigen Vater der Annotations kann man das Open-Source-Projekt XDoclet ansehen. XDoclet erlaubte es, auch schon in früheren Java-Versionen mit Annotations zu programmieren.
Bean	Eine Bean ist eine Java-Klasse bzw. Komponente, die bestimmten Konventionen entspricht. Die wichtigste Konvention definierte die Benennung der Zugriffsmethoden auf Attribute.
EE-Applikation	Ein Programm, das eine Java Enterprise Edition (EE) benötigt. Dies ist dann der Fall, wenn Entity Beans, Session Beans, Queues oder andere Techniken eingesetzt werden, die einen Application Server voraussetzen.
JSR	Java Specification Request (JSR) - Kennzeichnet eine Anforderung für eine Änderung der Programmiersprache Java, die vom Standardisierungskomitee, dem Java Community Process, eingebracht wird.
SE-Applikation	Ein Programm, das sich im Gegensatz zu einer EE-Applikation, auch mit einer Java Standard Edition (SE) begnügt. Es kann überall dort gestartet werden, wo eine SE installiert ist. Meistens sind dies Programme mit einer GUI (z.B. Swing) oder einer Steuerung über die Kommandozeile.

Links

- ▶ Download und Informationen zu Weld:
<http://seamframework.org/Weld>
- ▶ Informationen zu der Spezifikation JSR-299:
<http://jcp.org/en/jsr/detail?id=299>
- ▶ Informationen zu einem ähnlichen Produkt von Google:
<http://code.google.com/p/google-guice/>

sorgt das Framework dafür, dass stattdessen `logInvocation` des Interceptor aufgerufen wird. Diese Methode merkt sich die aktuelle Systemzeit und aktiviert mit dem Befehl `proceed` die eigentliche Methode `login`. Anschließend wird die Dauer berechnet und ausgegeben. Wie die Alternativen (s.o.) müssen auch die Interceptoren in der `META-INF/beans.xml` eingetragen werden, um diese zu aktivieren (siehe Abbildung 11).

Interceptoren können beliebig geschachtelt werden. So können z.B. für Sicherheitsüberprüfungen, Transaktionssteuerung, Logging etc. Interceptoren erstellt und den einzelnen Bean-Methoden beliebig kombiniert zugeordnet werden.

Fazit

Weld bietet die Möglichkeit auch SE-Applikationen in den Genuss der Dependency Injection kommen zu lassen und trotzdem kompatibel zum Standard (definiert durch JSR-299) zu bleiben.

Neben den normalen Funktionalitäten der Dependency Injection hat der Entwickler darüber hinaus noch die Möglichkeiten Interceptoren, Dekoratoren, Eventsteuerungen und noch Einiges mehr zu verwenden. Dies alles macht Weld zu einem recht mächtigen Werkzeug, auch in einer SE-Applikation.

Jedoch funktionieren die oben vorgestellten Verfahren nur, wenn alle Objekte, die injiziert werden sollen, in einer lückenlosen, Weld-gesteuerten Erzeugungskette stehen. Wird in der Kette nur ein Objekt mit `new` erzeugt, so wird dieses Objekt von Weld nicht mehr verwaltet und anschließende Injections können von Weld nicht mehr durchgeführt werden.



Andreas Massmann
(info@ordix.de).

ETL im Data Warehouse am Beispiel IBM DataStage (Teil III)

Stage für Stage zum Ziel

Im vorherigen Artikel dieser Reihe [1] wurde die grundlegende Architektur von IBM DataStage erläutert. Dieser Artikel geht nun auf die einzelnen Komponenten, die DataStage bietet, ein. Anhand eines Beispiels werden der Aufbau und die Funktionsweisen eines DataStage Job erläutert.

Dieser Artikel richtet sich an Data Warehouse Entwickler und Architekten.

Grundlegender Aufbau eines DataStage Job

Der Aufbau einer Verarbeitung mit DataStage wird durch die Anforderungen festgelegt. Ein ETL-Prozess kann sowohl in einem Job als auch in mehreren nacheinander bzw. parallel laufenden Jobs umgesetzt werden.

Als Beispiel zeigen wir einen kleinen Job, der in der Praxis nur einen Teil einer ETL-Jobkette darstellen würde. In unserem Beispiel sollen Daten aus einer Semikolon-separierten Datei mit bereits bestehenden Daten aus einer Oracle Datenbank verknüpft werden. In den nächsten Schritten werden diese Daten aggregiert und in eine DB2 Zieldatenbank eingespielt (siehe Abbildung 1).

Die Erstellung eines solchen DataStage Job erfolgt auf einer grafischen Entwicklungsoberfläche vom DataStage Designer. Die einzelnen Stages werden darauf platziert und parametrisiert. Die Stages sind untereinander mit Links verbunden, welche den Job-Verlauf bestimmen.

Durch die Möglichkeit des Imports von Metadaten, die Informationen über die Quell- und Zielsysteme enthalten, wird es dem Entwickler erleichtert, die Struktur beispielsweise einer Datenbanktabelle in einer Stage zu hinterlegen.

Diese Metadaten werden einmalig importiert und in einem Repository hinterlegt. Ab diesem Zeitpunkt können sie von allen Entwicklern benutzt werden.

Stages

Bei der Vorstellung der Funktionalitäten der einzelnen Stages werden wir uns an dem Beispiel-Job orientieren und uns auf die wichtigsten Stages der DataStage Server Edition beschränken.

Sequential File

Mittels der Sequential File Stage werden Textdateien vom System eingelesen und geschrieben. Dazu müssen in den Eigenschaften der Stage Speicherort, Struktur und Format hinterlegt werden (siehe Abbildung 2). Ob alles richtig eingestellt wurde kann über die Schaltfläche „Daten anzeigen“ überprüft werden.

Hashed File

Diese Stage dient dazu, Daten auf dem Datenträger mittels eines Hashing-Algorithmus, der durch eine Vielzahl von Parametern

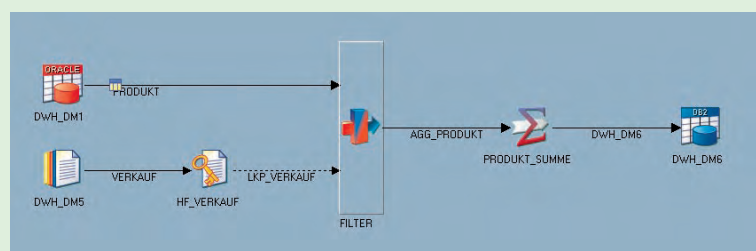


Abb. 1: Job-Übersicht.

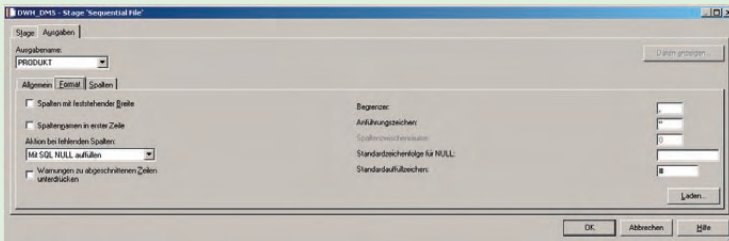


Abb. 2: Sequential File.

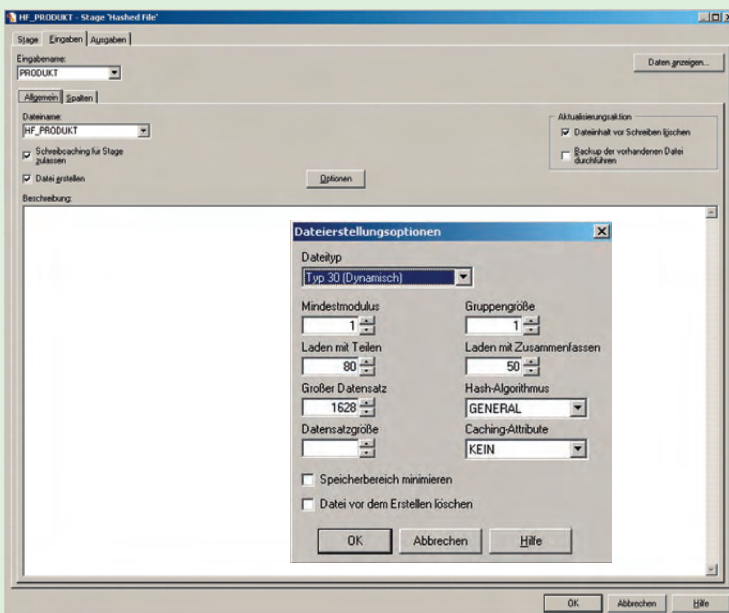


Abb. 3: Hashed File.

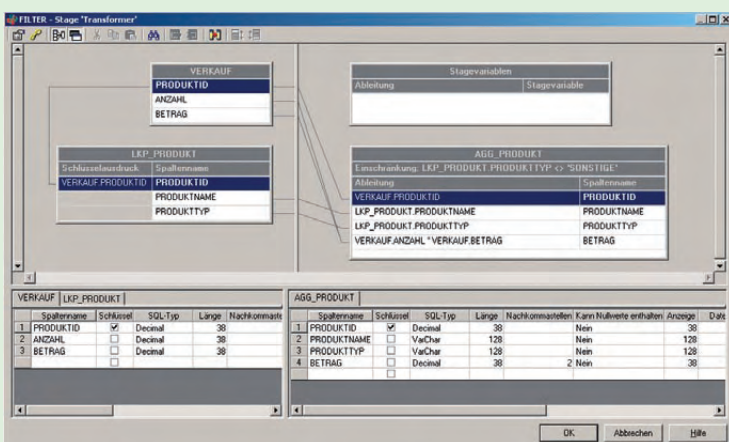


Abb. 4: Transformer Stage.

gesteuert und an konkrete Anforderungen angepasst werden kann, zu verteilen (siehe Abbildung 3). Durch die Möglichkeit der Definition eines Schlüssels, der aus einer oder mehreren Spalten bestehen kann und die Zugriffszeiten auf die einzelnen Datensätze enorm verkürzt, wird diese Stage meistens als eine Referenz- oder Bezugstabelle benutzt.

Transformer Stage

Dies ist die wohl wichtigste Stage, ohne die kaum eine Verarbeitung auskommt. Hier werden verschiedene Daten miteinander verknüpft, wie der Name schon sagt transformiert und in der veränderten Form weitergeleitet (siehe Abbildung 4). Durch die unzähligen Einstellmöglichkeiten und die Verwendung von sowohl bereits integrierten als auch selbstentwickelten Funktionen, stehen einem Entwickler hier alle Wege der Transformation offen.

Datenbankzugriff

DataStage bietet uns eine Reihe an Stages, über die der Zugriff auf eine Datenbank implementiert werden kann. Zusätzlich zu den spezifischen Stages für verschiedene Datenbanken steht auch eine ODBC Stage zur Verfügung, die uns erlaubt auf nahezu jede Datenbank zuzugreifen, sofern ein ODBC-Treiber existiert.

In unseren Beispiel-Job wird die Oracle Stage benutzt (siehe Abbildung 5). Diese erlaubt uns, über Oracle Net und eine `Tnsnames.ora`-Konfigurationsdatei, die Verbindung zur Datenbank herzustellen. Diese Stage dient jedoch nicht alleine der Datenbankverbindung. Hier werden ebenso Aktionen hinterlegt, die ausgeführt werden sollen. Dabei handelt es sich um DML- und DDL-Befehle in Form von SQL-Anweisungen.

Aggregation

Eine Aggregator Stage bietet die Möglichkeit, Datenzeilen in Gruppen zusammen zu fassen und so auch Gruppenfunktionen wie Summe und Durchschnitt auf jede Gruppe anzuwenden (beispielweise die Summenbildung aller Verkaufserlöse pro Produkt, siehe Abbildung 6).

Fazit

Mit DataStage sind wir in der Lage schnell auf veränderte Anforderungen an die ETL-Verarbeitung zu reagieren. Auch der Aufbau eines einfachen Job, wie in unserem Beispiel, benötigt nicht viel Zeit. Die Möglichkeit, Parameter der einzelnen Stages wie Daten-

bank-Login oder Dateipfade über Variablen zu setzen, vereinfacht den Entwicklungsprozess deutlich. So kann der Job an die jeweilige Umgebung angepasst werden, ohne dass Veränderungen an dem Job selbst vorgenommen werden müssen.

In diesem Artikel konnten wir nur einen kleinen Teil von DataStage vorstellen. Der Aufbau eines Jobs sieht auf der grafischen Oberfläche recht einfach aus, jedoch steckt hier der „Teufel im Detail“.

Die große Anzahl an Einstellungen innerhalb der einzelnen Stages erfordert viel Erfahrung mit DataStage, damit ein Job performant und effizient gestaltet werden kann.

Dieser Artikel bildet den Abschluss dieser Artikelreihe. Wir werden in einer der nächsten Ausgaben das Thema ETL wieder aufgreifen. Sollten Sie Fragen zu diesem Thema haben, wenden Sie sich gerne an uns.



Andreas Baier
(info@ordix.de).

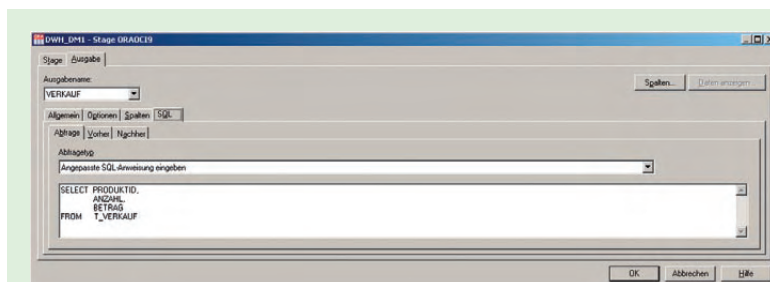


Abb. 5: Datenbankzugriff Oracle.

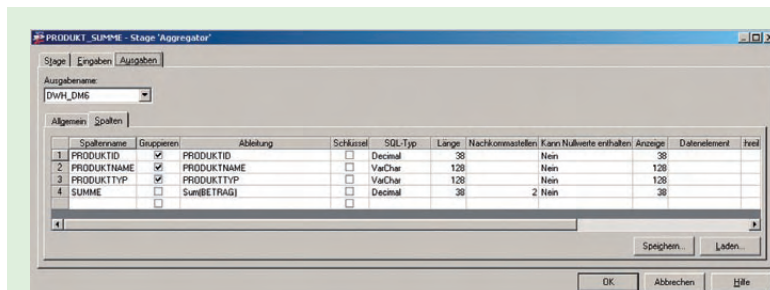


Abb. 6: Aggregation.

Glossar

- DDL** Data Definition Language – Über DDL-Kommandos werden Datenstrukturen gepflegt (z.B. Tabellen anlegen oder löschen).
- DML** Data Modification Language – Über DML-Kommandos werden Daten (Zeilen) in Tabellen eingefügt, gelöscht oder geändert.
- ODBC** Open Database Connectivity – Standardisierte Datenbank-schnittstelle, unabhängig vom Datenbankmanagement-system.
- Oracle Net** Kommunikationssoftware, welche eine Netzwerkverbindung von einer Client-Applikation zu einem Oracle Datenbank-server ermöglicht.
- Stage** Grafische Entwicklungsumgebung von Teilen einer Verarbeitung. Hier vorgenommene Einstellungen werden von DataStage interpretiert und in Programmcode umgewandelt.
- Tnsnames.ora** Oracle Konfigurationsdatei für Verbindungen zu Instanzen.

Links

- ▶ [1] ORDIX News Artikel 1/2011: „Die Architektur einer ETL-Software“: http://www.ordix.de/ORDIXNews/1_2011/etl_datawarehouse_datastage.html
- ▶ [2] IBM Websphere DataStage Homepage: <http://www-142.ibm.com/software/products/de/de/ibminfodata/>
- ▶ [3] DataStage-Forum: <http://www.dsxchange.com/>



DOAG Konferenz und Schulungstag 2011

Wissentransfer auf hohem Niveau

2011
DOAG
Konferenz + Ausstellung

Jedes Jahr im November trifft sich die Oracle Community im CongressCenter Nürnberg Ost um Neuigkeiten und praxisnahe Erfahrungsberichte auszutauschen. In diesem Jahr trifft man sich vom 15. bis zum 17. November. In rund 400 Vorträgen bietet die DOAG Konferenz + Ausstellung 2011 wieder viel Wissen rund um das Thema Oracle. Die ORDIX AG ist in diesem Jahr mit 4 Fachvorträgen und einer eintägigen Schulung dabei.

Wissentransfer seit 11 Jahren

Die ORDIX AG nimmt in diesem Jahr zum 11. Mal an der DOAG Konferenz + Ausstellung teil. Vier Fachvorträge bilden dabei den Rahmen des 3-tägigen Messegeschehens. Abgerundet wird die diesjährige Konferenz mit einem Schulungstag zum Thema „Oracle PL/SQL Tuning“.

Flashback mal sieben

Den Anfang macht am ersten Konferenztag Klaus Reimers mit seinem Vortrag „Flashback mal sieben“. Flashback ist mit Oracle 9i etabliert worden und wird mittlerweile bei vielen Kunden eingesetzt. Allerdings ist Vielen nicht klar, wie vielfältig dieses Thema ist und welche Möglichkeiten hiermit verbunden sind.

Klaus Reimers stellt in seinem Vortrag die sieben Varianten des Flashback vor, zeigt mögliche Praxiseinsätze auf und wägt Vor- und Nachteile gegeneinander ab.

MySQL-Replikationen

Am zweiten Konferenztag erläutert Matthias Jung die generelle Funktionsweise und Konfiguration von MySQL-Replikationen und erörtert, wie und wo diese Funktion im täglichen Betrieb sinnvoll eingesetzt werden kann.

MySQL unterstützt bereits seit der Version 3.23 die Möglichkeit, Daten zwischen mehreren Systemen zu replizieren. Oftmals wird dieses Konstrukt in Zusammenhang mit Backup-, Verfügbarkeits- und/oder Scale-out-Konzepten (Performance-Optimierungen) eingesetzt.

Features des Oracle Warehouse Builder 11.2

Am gleichen Tag findet der Vortrag „Features des Oracle Warehouse Builder 11.2“ statt. ETL-Tools wurden in den letzten Jahren zu einem unumstößlichen Instrument, um Daten aus verschiedenen heterogenen Quellen in ein Data Warehouse zu importieren. Durch das Anwachsen der Datenstrukturen und -quellen werden immer größere Anforderungen an das Performancetuning des ETL-Prozesses, das Design des Workflow-Prozesses und das Troubleshooting von ETL-Prozessen gestellt.

Der Vortrag von Franz von Sales Hohenberg wird die einfache und sichere Handhabung des Oracle Warehouse Builder im Bereich ETL-Performance, Data-Profiling und Einhaltung der Datenqualität darstellen. Mehrere Beispiele aus aktuellen Projekten werden den Vortrag untermauern.

The Good, the Bad and the Ugly

Zu guter Letzt stellt Martin Hoermann am dritten Konferenztag die Index-Zugriffsstrategien von Oracle vor und untersucht wesentliche Kennziffern, die auf diese Zugriffsstrategien Einfluss haben. Hierzu gehören beispielsweise die Höhe des Index-Baums, die Anzahl der Blätter (**leaf-nodes**) und der Clustering-Faktor. Aber auch weniger beachtete Kennziffern, wie der Speicherparameter **PCTFREE** und die Anzahl gelöschter Blätter finden Beachtung.

DOAG Schulungstag

Direkt im Anschluss an die DOAG Konferenz lädt Sie die ORDIX AG am 18. November 2011 von 9 bis 16 Uhr zu einer eintägigen Schulung ebenfalls in das CCN Ost ein. Der Referent, Markus Fiegler, stellt in seinem Seminar das Thema „Oracle PL/SQL Tuning“ vor.

Wir freuen uns auf Sie bei der DOAG 2011 in Nürnberg!

Unseren Messestand finden Sie auf Etage 3, Stand 324.

Die Vorträge im Überblick



Flashback mal sieben

Dienstag, 15.11.2011
10:00 – 10:45 Uhr, Raum Tokio
Referent: Klaus Reimers,
ORDIX AG



MySQL-Replikationen

Mittwoch, 16.11.2011
09:00 – 09:45 Uhr, Raum Singapur
Referent: Matthias Jung,
ORDIX AG



Features des Oracle Warehouse Builder 11.2

Mittwoch, 16.11.2011
12:00 – 12:45 Uhr, Raum Stockholm
Referent: Franz von Sales Hohenberg,
ORDIX AG



Index-Rebuild - The Good, the Bad and the Ugly

Donnerstag, 17.11.2011
12:00 – 12:45 Uhr, Raum Tokio
Referent: Martin Hoermann,
ORDIX AG



*Evelyn Ernst
(info@ordix.de).*

DOAG-Schulungstag: Oracle PL/SQL Tuning

Inhalte:

- Datentypen
- Coding
- XMLType-Speicherungsformen
- SQL-Tuning
- Tools zur Performance-Analyse
- Tipps & Tricks

Zeit/Ort:

18. November 2011
09:00 - 16.00 Uhr
CongressCenter Ost, Nürnberg
Referent: Markus Fiegler

Preis:

450 € zuzügl. MwSt.
Melden Sie sich gleich an unter:
<http://www.doag.org/konferenz/doag/2011/>

Weitere Seminarangebote finden Sie im ORDIX-Trainingsshop unter: <http://training.ordix.de>.





Agiles Projektmanagement mit Scrum: Interview mit Lars Eisenblatt

Agil im Betriebsprojekt!

Scrum als agiles Projektmanagementverfahren wird häufig mit dem Bereich der Softwareentwicklung verbunden. Das Verfahren ist aber hervorragend für alle Projektarten geeignet, bei denen die Anforderungen starken Änderungen ausgesetzt sind oder zu Beginn nur unvollständig vorliegen. Im folgenden Interview zwischen der ORDIX news und Lars Eisenblatt, dem Vorstand der coniatos AG, werden die Erfahrungen bei der Einführung von Scrum dargestellt.

ORDIX news: In welchem Umfeld haben Sie Scrum eingesetzt?

Lars Eisenblatt: Im Rahmen eines neuen Auftrages für einen Outsourcer von IT-Dienstleistungen war es notwendig geworden, die Organisation des Datenbankbetriebes zu optimieren. Dabei musste die Anzahl der zu verwaltenden Datenbanken im 7 x 24 Std.-Betrieb verdreifacht und ein zusätzliches drittes DB-Produkt, parallel zu den beiden bereits bestehenden, eingeführt werden. Die Projektziele umfassten die Aufwandsreduktion unter Beibehaltung der Servicequalität und die Verbesserung der Compliance.

Das Unternehmen verfügte über ein Vorgehensmodell (VGM) für Entwicklungs- und Einführungsprojekte, sowie eine iterativ-inkrementelle Variante. Aufgrund der zu Projektbeginn unbekanntem Detailanforderungen wurde das Vorgehensmodell für Schlüsselprojekte um Scrum erweitert.

ORDIX news: Welche Vorteile ergaben sich dadurch?

Lars Eisenblatt: Durch die Agilität erreichen wir eine höhere Flexibilität. Die Anforderungen können ständig erweitert oder verändert werden, während die wichtigsten Anforderungen bereits umgesetzt werden.

Die formellen und informellen Anforderungen werden in einem Product Backlog vom Product Owner in enger Abstimmung mit dem Anforderer/Auftraggeber gesammelt. Die jeweils wichtigsten Anforderungen werden dann in Zyklen (Sprints) abgearbeitet und führen am Sprint-Ende immer zu „auslieferbaren“ Ergebnissen.

Nach jedem Sprint erfolgt eine neue Priorisierung und Planung, wodurch eine optimale Anpassung an das Ziel möglich ist. Es wird maximal der Aufwand eines Sprints riskiert: Bekommt man nicht das gewünschte Ergebnis, steuert man mit neuen Anforderungen gegen.

ORDIX news: Wer legt fest, was das Wichtigste ist?

Lars Eisenblatt: Dies ist die Aufgabe des Auftraggebers in Verbindung mit dem Product Owner. Er legt jeweils für den nächsten Sprint fest, welches die wichtigsten Anforderungen sind, die in diesem Sprint umgesetzt werden sollen. Hierbei priorisiert er die Themen zu Sprint-Beginn in der Sprint-Planungsphase.

ORDIX news: Wie wird ein Sprint geplant?

Lars Eisenblatt: Jeder Sprint beginnt mit einer Planungsphase. Die Planung findet in einem Teammeeting statt. Der Product Owner stellt dem Team die umzusetzenden Anforderungen aus dem Product Backlog vor. Die Anforderungen sind dabei bereits grob geschätzt.

Diese Grobschätzungen werden anschließend von dem Team verifiziert. Dazu ist es notwendig, dass jedes Teammitglied die Anforderung versteht. Für Rückfragen steht der Product Owner bzw. der Auftraggeber bereit. Das Team wählt aus den priorisierten Anforderungen diejenigen als Arbeitsaufgabe aus, die es glaubt mit der gegebenen Zusammensetzung in der Sprint-Länge umsetzen zu können. Das Team übernimmt die ausgewählten Anforderungen in ihr Sprint Backlog. Darin erfolgt eine Zerlegung in Features und eine detailliertere Schätzung.

Zu beachten ist, dass die Schätzung nicht besonders genau sein muss, da maximal der Arbeitsumfang der nächsten 30 Tage geschätzt wird. Eine Sprint-übergreifende Planung ist das Release Planning. Dabei werden Aufgaben grob geschätzt, die in einer bestimmten Reihenfolge über mehrere Sprints hinweg durchgeführt werden müssen.

ORDIX news: Wie erfolgt die Überwachung des Fortschritts?

Lars Eisenblatt: Während der Abarbeitung wird in den Daily Scrums jeweils abgefragt, welche Anforderungen inwieweit fertig gestellt worden sind, was als nächstes geplant ist und welche Hindernisse die effiziente Arbeit schwierig machen. Die jeweils abgearbeiteten Anforderungen bzw. verbrauchte Stunden werden in ein Tableau eingetragen. Zur Visualisierung wird im Teamraum ein so genanntes Burndown-Chart fortgeschrieben, das den idealen sowie den aktuellen Fortschritt in Verbindung zueinander setzt. Damit erhält jedes Teammitglied jederzeit

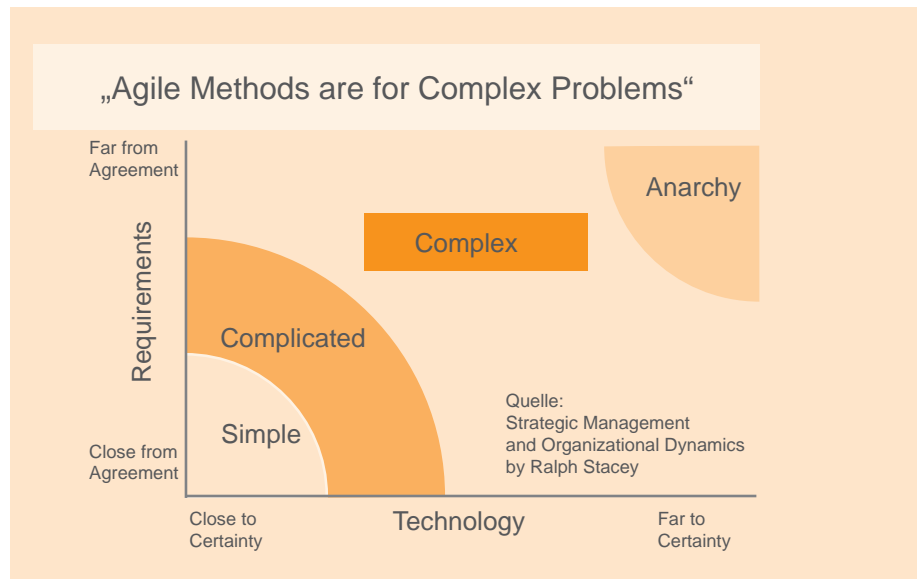


Abb. 1: Sprint Planungsphasen.

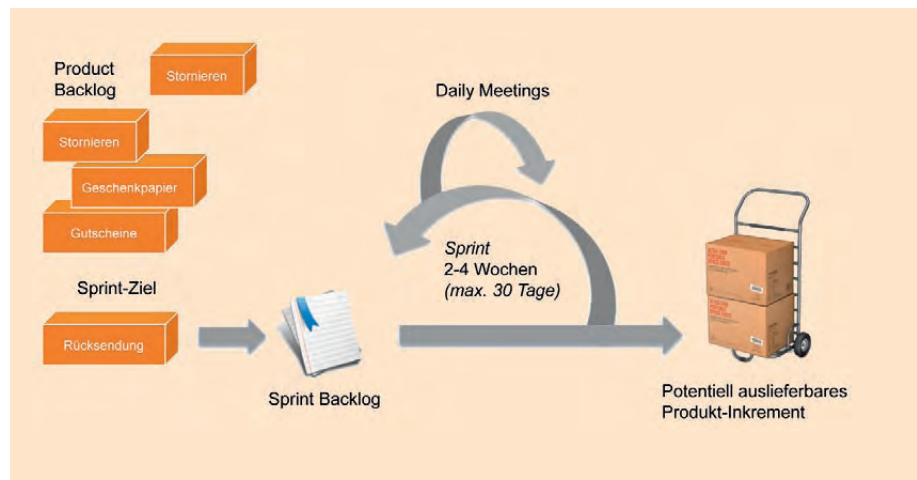


Abb. 2: Scrum Überblick.

einen Eindruck über den aktuellen Stand des Fortschritts.

ORDIX news: Was ist der Scrum Master?

Lars Eisenblatt: Der Scrum Master unterstützt das autonome Team in ihrer Arbeit, dabei ist er insbesondere verantwortlich für die Beseitigung von Hindernissen. Er agiert als Coach und Change Agent, er steuert den Scrum-Prozess und sorgt für eine direkte

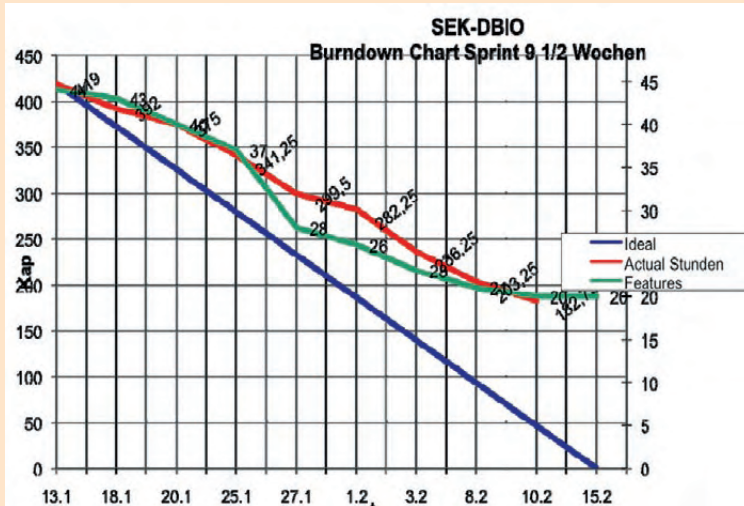


Abb. 3: Burndown-Chart.

Kommunikation des Teams mit dem Product Owner. Er hat keine Personalverantwortung und ist nicht für die Leistungsbeurteilung verantwortlich.

Er moderiert die Meetings für die Sprint-Planung und den Product Review, sorgt für die regelmäßige Durchführung von Daily Scrums und der Retrospective. Er wird typischerweise aus dem Team heraus bestimmt oder auch zusammen mit dem Team benannt.

ORDIX news: Was ist gemeint mit „auslieferbaren“ Ergebnissen?

Lars Eisenblatt: Die auslieferbaren Ergebnisse werden am Ende der Sprints im Sprint Review Meeting mit dem ganzen Team dem Product Owner vorgestellt, der sie abnehmen muss.

Dabei ist es wichtig, dass die Abnahmekriterien aus dem Product Backlog eindeutig erreicht werden. Ist eine Arbeitsaufgabe, die aus dem Product Backlog übernommen wurde nicht vollständig fertig geworden, gilt sie als nicht erledigt. Sie kann dann, bei gegebener Priorität durch den Product Owner in einem der nächsten Sprints fertiggestellt werden.

Nach dem Sprint Review findet eine Retrospective statt. Diese Retrospective dient der Prozessverbesserung des Teams. Dabei werden Vorschläge zur Effizienzsteigerung des Teams durch Tools, Veränderungen der Parameter oder einfach Konfliktlösungen diskutiert und anschließend durch das Team und den Scrum Master implementiert.

ORDIX news: Wie haben Sie die typischen Parameter in Scrum gewählt?

Lars Eisenblatt: Unser Team bestand aus 12 teilweise sehr erfahrenen Mitarbeitern, die aber jeweils nur 5 - 40 % für das Projekt zur Verfügung standen. Des Weiteren gingen die Betriebssicherheit und Betriebsproblembeseitigung immer vor. Um den Verwaltungsaufwand für die Meetings klein zu halten, haben wir uns zu einer Sprint-Länge von 30 Tagen entschlossen. Wir haben mit einer Geschwindigkeit von 20 idealisierten Tagen (IT) begonnen.

Die letzten drei Sprints wurden aufgrund des Feedbacks aus der Retrospective, sogar auf 35 Tage erweitert, wobei die Geschwindigkeit im gleichen Maße erhöht wurde. Dadurch wurde eine gleichmäßigere Zielerreichung gewährleistet.

Die Daily Scrums wurden nur Dienstags und Donnerstags, jeweils auf 30 Minuten begrenzt durchgeführt. Die Meetings zur Sprint-Planung 1 und 2 wurden auf jeweils 4 Stunden festgelegt. Sowohl der Sprint Review als auch die Retrospective wurden mit jeweils maximal 2 Stunden festgesetzt. Es wurde, wie in allen agilen Verfahren, besonders darauf geachtet keine Zeiten zu überschreiten, bedarfsweise wurden daher Folgemeetings einberufen.

Auch die Sprint-Planung wurde adaptiert: Um eine fachlich orientierte präzisere Schätzung durchführen zu können, wurde zwischen den Meetings zur Sprint-Planung 1 und 2 eine Gruppenarbeit gesetzt. Damit konnten die Themen eher fachspezifisch zerlegt und geschätzt werden. Die zusammengeführten Ergebnisse wurden dann aber mit dem gesamten Team abgestimmt und beschlossen.

ORDIX news: Wie ist Scrum bei den Mitarbeitern angekommen?

Lars Eisenblatt: Nach einer Eingewöhnungsphase waren die Mitarbeiter sehr zufrieden. Sie wussten jederzeit, wie weit sie waren, was noch offen war und konnten sich flexibel einbringen. Einzelne Mitarbeiter haben sich zu im Team anerkannten Spezialisten für bestimmte Werkzeuge oder Gebiete entwickelt. Durch die gemeinsame Planungsarbeit konnten Erfahrungen effizient ausgetauscht und von bestehenden Plattformen einfach an neue weitergeben werden.

Umgekehrt sind viele neue Ideen auch zur Verbesserung bestehender Plattformen genutzt worden. Durch das gemeinsame Schätzen von Anforderungen stieg das Verständ-

nis für die umzusetzende Aufgabe. Durch die Daily Scrums kam es zu einem kurzfristigen Ideen- und Informationsaustausch zwischen verschiedenen Aufgaben.

Viele Mitarbeiter mussten zuerst das Zerlegen und Schätzen komplexer Aufgaben lernen, insbesondere wenn dabei das Ergebnis bereits definiert werden musste. Zu Beginn gab es aber auch Findungsprobleme der Gesamtverantwortung: das Team in Scrum ist autonom, der Scrum Master unterstützt es nur. So liegt die Verantwortung für alle geplanten Aufgaben bei allen Mitgliedern, d.h. wenn etwas nicht fertig wird, sind alle dafür verantwortlich!

ORDIX news: Wie wurde Scrum vom Management wahrgenommen?

Lars Eisenblatt: Es ist als ein Verfahren, das schnell und flexibel hochwertige Ergebnisse liefert wahrgenommen worden. Die Kommunikation mit dem klassischen Projektcontrolling über Projektstatusberichte und Meilensteine sowie deren Termingenauigkeit musste angepasst werden. Die Überwachung und Bewertung des Projektfortschritts und der Zielerreichung wurde durch veränderte Leistungskennzahlen (KPIs = Key Performance Indicators) angepasst. So wurde anhand der auslieferbaren Ergebnisse der Mehrwert in Bezug auf die Projektziele ermittelt und überwacht.

Das Projektcontrolling empfand die hohe Transparenz durch das Product Backlog, Sprint Backlog, Sprint Burndown, Product Burndown u.ä. Werkzeuge als besonders vorteilhaft.

ORDIX news: War Scrum ein Erfolg ?

Lars Eisenblatt: Ich denke, dass der Einsatz von Scrum ein Erfolg war. Zum Startzeitpunkt waren die Anforderungen nur schemenhaft bekannt. Die umzusetzenden Maßnahmen entwickelten sich im Laufe des Projektes sehr schnell weiter. Durch Scrum war es uns aber möglich, schnell und flexibel darauf zu reagieren.

Sicherlich hätte das Projekt auch mit einem klassischen Vorgehensmodell erfolgreich zu Ende geführt werden können. Scrum ist eine Best-Practice-Entwicklung aus dem Krisenmanagement von Projekten. Wäre das Projekt aufgrund der vielen Änderungen vom Projektcontrolling in den Krisenstatus erhoben worden, hätten ähnlich flexible Verfahren gegriffen.

Glossar

Daily Scrum	Häufiges, meist tägliches Teammeeting von ca. 15 Minuten Länge, in dem der aktuelle Stand, die weitere Planung und aufgetretene Hindernisse besprochen werden.
Product Backlog	Katalog der Produktmerkmale, die im aktuellen Sprint umgesetzt werden.
Product Owner	Der Product Owner ist verantwortlich für die Anforderungen und deren Priorisierung in Zusammenarbeit mit dem Auftraggeber und Know-how-Träger.
Sprint	Iteration fester Länge zur Umsetzung von Anforderungen.
Scrum Master	Der Scrum Master unterstützt das Team im Scrum-Prozess bei der Umsetzung der Anforderungen.
Change Agent	Dieser Agent sorgt für geeignete Rahmenbedingungen, die ein natürliches Heranreifen quasi-autonomer Entscheidungen und ihrer Umsetzungen fördern.
Retrospective	Feedback des Teams zum letzten Sprint bezüglich des Prozesses.
Idealisierte Tage	Mögliche Maßeinheit zum Schätzen der Aufwände.
Key Performance Indicator (KPI)	Metrik um die Effizienz eines Prozesses zu beurteilen.

Links

- ▶ [1] ORDIX News Artikelreihe „IT-Management“: http://www.ordix.de/ORDIXNews/artikelreihen.html#it_management
- ▶ [2] Webseite der coniatos AG: <http://www.coniatos.de>
- ▶ [3] Seminarempfehlung: Agiles Projektmanagement mit SCRUM <http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Projektmanagement/nr/1014/index.html>

ORDIX news: Gibt es noch andere agile Verfahren außer Scrum?

Lars Eisenblatt: Ja, im Rahmen des Betriebes haben wir ebenfalls sehr gute Erfahrungen mit Kanban zur Arbeitsplanung in der Linie gemacht.



Lars Eisenblatt
(info@coniatos.de).

ORDIX news: Wie können Sie Kunden beim Einsatz agiler Verfahren unterstützen?

Lars Eisenblatt: Wir unterstützen gerne mit Coaching und Beratung zur Einführung von Scrum, wir stehen als Scrum Master oder Product Owner zur Verfügung und führen nach Bedarf gerne Scrum Projekt Audits durch.

ORDIX news: Wir bedanken uns für das Gespräch!

Zertifizierte Servervirtualisierung aus dem Hause Oracle (Teil II)

Virtualisierung und Hochverfügbarkeit mit Oracle VM

Dieser Artikel richtet sich an Administratoren, die ihre Infrastruktur auf virtuellen Maschinen betreuen und verwalten.

Während sich der erste Artikel dieser Reihe [1] auf die grundlegende Vorstellung des Produktes konzentrierte, richtet sich das Augenmerk in diesem Artikel auf die technische Umsetzung. Darüber hinaus werden wichtige Schlüsselfunktionalitäten auf ihre Praxistauglichkeit getestet und bewertet.

Aufbau Oracle VM Manager

Der VM Manager dient der Verwaltung und Kontrolle von virtuellen Maschinen. Diese laufen auf Servern, die in Oracle VM zu sogenannten Server-Pools zusammengefasst werden. Um den VM Manager zu installieren wird als Basisbetriebssystem wahlweise ein Oracle Enterprise oder ein Red Hat Enterprise Linux der Version 4.5 (oder höher) benötigt.

Ein ISO Image der Software steht bei Oracle kostenlos zum Download bereit. Nach dem Download kann es mit folgendem Befehl eingebunden werden:

```
mount -o loop /tmp/OracleVM-Manager-2.2.0.iso /ovmman
```

Im eingebundenen Verzeichnis kann der interaktive `runInstaller` aufgerufen werden. Zunächst werden die Hard- und Softwarevoraussetzungen geprüft. Als Repository-Instanz für den Manager kann eine bestehende Oracle Datenbank genutzt oder die im Installationspaket enthaltene Oracle 10g Express verwendet werden.

Anschließend werden die Passwörter für den Admin User und den Webserver definiert. Nach der erfolgreichen Installation kann man sich per Browser über den entsprechenden Host-Namen oder die IP-Adresse über den Port 4443 beim Manager anmelden.

Installation Oracle VM Server

Die Installation der einzelnen Server funktioniert ohne ein vorinstalliertes Basisbetriebssystem. Das zum kostenlosen Download angebotene ISO Image zum Aufsetzen von VM Servern ist bootfähig. Es beinhaltet einen Xen Hypervisor, einen Linux-Kernel und einen Agenten, der für die Kommunikation mit dem Manager zuständig ist. Die so installierten Server werden einem Server-Pool zugewiesen und erhalten eine bestimmte Rolle.

Einrichtung des Shared Storage im Server-Pool

Um VMs hochverfügbar aufzusetzen, sind gewisse Voraussetzungen notwendig. So müssen die VM Server beispielsweise über einen gemeinsamen Speicherbereich (Shared Storage) verfügen. Sofern diese Anforderung erfüllt ist, kann eine VM unterbrechungsfrei von einem auf den anderen VM Server migriert werden.

Zur Identifizierung des Shared Storage eines Server-Pool wird der UUID (Universally Unique Identifier) verwendet. Dieser muss für alle VM Server identisch sein, wenn man den Server-Pool im Hochverfügbarkeitsmodus einrichten möchte. Zur Verwaltung der UUIDs wird das Python-Skript `repos.py` verwendet (siehe Abbildung 1).

Zur Initialisierung des Storage Repository müssen zunächst die Cluster-Konfigurationsdateien auf jedem Server angepasst werden (siehe Abbildung 2).

Anlegen von Server-Pools

Ein Server-Pool besteht aus mindestens einem VM Server. Hochverfügbare Server-Pools bestehen mindestens aus zwei Maschinen, die auf den gemeinsamen Speicherbereich zugreifen können. Zur Erstellung eines hochverfügbaren Pool muss der Parameter „High Availability Mode“ (siehe Abbildung 3) gewählt werden.

Jeder Server übernimmt in seinem Verbund eine oder mehrere Rollen:

- Server-Pool Master
- Utility Server
- VM Server

Während der Erstellung eines jeden Pool wird der Shared Storage initialisiert. Daraufhin steht im Dateisystem unter dem Verzeichnis /ovs, auch als Cluster Root bezeichnet, die folgende Verzeichnisstruktur zur Verfügung:

- **iso_pool**
zur Aufnahme von ISO Images
- **publish_pool**
beinhaltet jedem zugängliche guest virtual machine
- **seed_pool**
Speicherung von Templates
- **shared_disks**
geteilte Speicherbereiche laufender virtueller Maschinen
- **running_pool**
VM Images und Konfigurationsdateien

Verwaltung von virtuellen Maschinen

Der Oracle VM Manager unterstützt drei verschiedene Methoden zur Erstellung virtueller Maschinen:

- Templates
- Installationsmedien
- Netzwerk-Boot (PXE Boot)

Zur Einrichtung einer hochverfügbaren VM muss bei jeder Erstellungsart die Option High Availability explizit eingeschaltet werden.

Auf allen Servern des Server-Pool:

```
Cd /opt/ovs-agent-2.3/utlils
./repos.py
./repos.py -n /dev/sdd1
[ NEW ] 363fb538-3a3b-4693-ab2f-9884aa86f3fa => /dev/sdd1
./repos.py -r UUID
[ R ] 363fb538-3a3b-4693-ab2f-9884aa86f3fa => /dev/sdd1
```

Abb.1: Agent Repository einrichten.

```
node:
  ip_port      = 7777
  ip_address   = 172.17.95.104
  number      = 0
  name        = oravmsr1
  cluster     = ocfs2

node:
  ip_port      = 7777
  ip_address   = 172.17.95.105
  number      = 1
  name        = oravmsr2
  cluster     = ocfs2

cluster:
  node_count  = 2
  name        = ocfs2
```

Abb.2: Cluster-Konfigurationsdatei zur Initialisierung des Storage Repository.

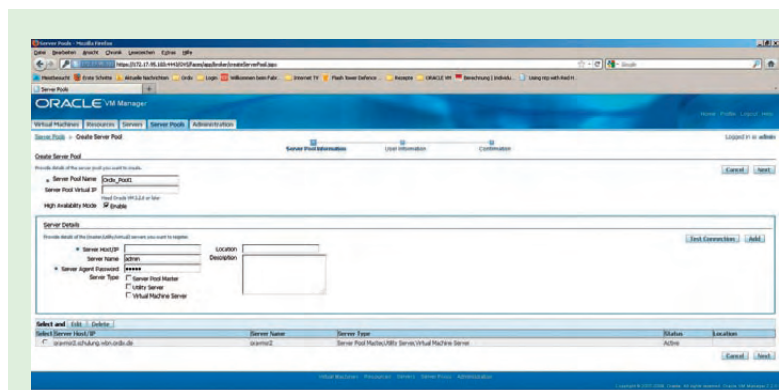


Abb. 3: Hochverfügbarkeit des Server-Pool.

Beim Herunterfahren eines Oracle VM Server können die dort betriebenen hochverfügbaren virtuellen Maschinen automatisch auf eine andere Maschine des Server-Pool verschoben werden. Die Anwender können unterbrechungsfrei weiterarbeiten. Die Hochverfügbarkeitseinstellung bleibt ohne Bedeutung, sollte der Server-Pool nicht hochverfügbar sein.

Zur vereinfachten Verteilung der einzelnen virtuellen Maschinen und zur besseren Lastverteilung kann die Zuweisung einer VM auf die im Server-Pool verbleibenden Server automatisch erfolgen.

Erstellung mit Templates

Oracle stellt vordefinierte Templates zum Download bereit. Diese reichen von einfachen Betriebssystemen wie zum Beispiel einem Oracle Enterprise Linux über installierte Datenbanken bis hin zu komplexen RAC-Architekturen. Die Erstellung von eigenen Templates aus bestehenden virtuellen Maschinen ist ebenso möglich. Die im Shared Storage abgelegten Vorlagen müssen dem VM Manager als Ressource bekannt gemacht und vom Administrator freigeschaltet werden. Im Dateisystem liegt unter `/OVS/seed_pool` ein Ordner mit dem Namen des Template.

Virtuelle Maschinen aus ISO Images

Analog zu der Template-Verwaltung verhält es sich auch mit den ISO Images. Die hinterlegten ISO-Image-Dateien werden im ISO Pool des Shared Storage hinterlegt. Die Dateien können direkt im Dateisystem abgelegt oder von einer externen Ressource via HTTP oder FTP geladen und dem Manager bekannt gemacht werden.

Für jede erstellte VM wird im Dateisystem im Verzeichnis `/OVS/running_pool` ein Ordner, der den Maschinennamen beinhaltet, angelegt. Neben dem eigentlichen Image wird hier eine Konfigurationsdatei generiert.

Operationen einer laufenden VM

Während des Lebenszyklus einer virtuellen Maschine, können verschiedene Ereignisse auftreten, die eine Statusänderung erforderlich machen. Oracle VM ermöglicht es, diese Aufgaben über die Administrationsoberfläche des Manager vorzunehmen.

- **Pausieren einer VM**

Die virtuelle Maschine kann jederzeit eingefroren werden. Dabei laufen sämtliche Prozesse, die keine Netzwerkkommunikation zu anderen Maschinen benötigen, nach dem Pausieren ungestört weiter. Die verteilten Prozesse müssen neu angestoßen werden. Sollte die VM z.B. als Webserver fungieren, erscheint es dem Client als wäre dieser Server heruntergefahren.

- **Status „Suspend“**

Der aktuelle Zustand einer Maschine wird im Status Suspend zum jeweiligen Zeitpunkt auf der Platte abgelegt. Diese Funktion tritt auch per Voreinstellung beim Herunterfahren eines VM Server in Kraft. Der Hauptspeicher der DomUs wird so gesichert, dass die virtuellen Maschinen beim Neustart des Server nahtlos fortgeführt werden können (ein Boot-Vorgang entfällt). Netzwerkverbindungen werden in diesem Status natürlich aufgelöst.

Duplizierung von Maschinen

Für den schnellen Aufbau gleichartiger virtueller Maschinen bietet Oracle VM mehrere Möglichkeiten an. Die Methoden Deploying, Klonen und die Abspeicherung als Template erfordern es, dass die VM im ausgeschalteten Zustand (Powered Off) ist. Ein Starten der Maschinen ist während dieser Vorgänge unmöglich.

Die Verfahren Deployen und Klonen sind sich sehr ähnlich. Mit ihnen können einzelne Maschinen kopiert werden. Diese virtuellen Maschinen können ebenfalls in anderen Server-Pools ausgerollt werden. Der einzige Unterschied dieser Methoden besteht darin, dass man mit dem Klonen mehrere Kopien auf einmal generieren kann. In diesem Fall wird für den Namen der zu klonenden Maschinen lediglich ein Präfix angegeben, der um eine fortlaufende Nummer erweitert wird.

Die Ablage als Template ist ein gutes Verfahren, um zunächst ein Betriebssystem mit den grundlegenden Programmen zu erstellen und zu konfigurieren. Dieses Template kann nun für den Aufbau mehrerer identischer Maschinen herangezogen werden. So erstellte Templates müssen nicht mehr vom Oracle VM Administrator freigegeben werden.

Die Methode der Duplikation beruht auf Dateikopien. Für das Deployment- und Klonverfahren werden die Dateien der VMs zwischen den Ordnern des `running_pool` und bei der Erstellung von Templates, zwischen dem `publish_pool` und `seed_pool` kopiert.

Migration während des Betriebes

Der unterbrechungsfreie Transfer einer VM zwischen zwei Servern (Live Migration) ist wohl eine der spannendsten Funktionen. Wartungsfenster führen so nicht mehr zu

Unterbrechungen der eigentlichen Anwendung.

Allerdings bringt dieses Funktionen auch einige Einschränkungen mit sich. Server-Migrationen funktionieren nur zwischen identisch konfigurierten Maschinen innerhalb eines Server-Pool.

Fazit

Der Einsatz von Oracle VM setzt eine sorgsame Planungsphase voraus. Neben der Planung des Shared Storage, muss die Zusammensetzung vom Server-Pool und die Rollenverteilung innerhalb des Pool festgelegt werden.

Ein Einsatz von Oracle VM kann vor allem in recht homogenen IT-Landschaften sinnvoll sein. Die schnelle und einfache Erstellung identischer virtueller Maschinen aus Templates oder ISO Images sorgt für eine Zeitersparnis und gewährleistet eine standardisierte Umgebung.

Eine weitere Grundüberlegung, die für den Einsatz von Oracle VM spricht, ist die Realisierung von Hochverfügbarkeitslösungen (Live Migration) ohne dabei RAC oder Dataguard einsetzen zu müssen.

Glossar

**DomU
(Domain U) (Gast)**

Virtuelles System, das auf derselben Hardware wie der Wirt läuft.

UUID

Universally Unique Identifier – Er dient dazu, Informationen in verteilten Systemen ohne zentrale Koordination eindeutig kennzeichnen zu können.

Links

- ▶ [1] ORDIX News Artikel „Blick hinter die Kulissen: Oracle VM für x86“:
http://www.ordix.de/ORDIXNews/1_2011/oracle_vm.html
- ▶ [2] Download des Konsolen Plug-In:
<http://oss.oracle.com/oraclevm/manager/RPMS/>



*Ole Breimann
(info@ordix.de).*

Dieser Platz ist frei! Für Sie?

Werden Sie Teil unseres TEAMS von Datenbank-Spezialisten als

- Oracle Senior Consultant (m/w)
- DB2 Consultant (m/w)

Besuchen Sie unser Bewerberportal:

www.ich-will-ins-ordix-team.de



IBM Informix 11.70 (Teil III) - Neue Hochverfügbarkeitsfunktion

Transaction survival – der Panther überlebt

Dieser Artikel richtet sich an Administratoren und Entscheider, die eine Skalierung der Applikation anstreben.

Mit der Version 11.70.xC2 hat IBM eine neue Funktion mit dem Namen „Transaction survival“ in Informix eingebaut. Transaktionen überleben von nun an auf UPDATABLE Secondary Servern in hochverfügbaren Datenbankumgebungen. In dem dritten Artikel dieser Reihe gehen wir auf diese neue Funktion ein.

Hochverfügbarkeitslösungen mit Informix Version 11.70

Die aktuellen Hochverfügbarkeitslösungen sind unter dem Codenamen MACH11 (Multinode Active Cluster for High Availability) bekannt.

Diese Lösungen sind direkt in dem Informix-Kernel integriert und umfassen eine ganze Reihe an Möglichkeiten, die die Verfügbarkeit und Ausfallsicherheit von Datenbanken verbessern.

Folgende Funktionalitäten sind in beliebiger Kombination möglich:

- **HDR (High Availability Data Replication)**
Diese Funktion ist eine 1:1-Primary-/Secondary-Replikation. Sie ermöglicht

einen automatisierten Failover und Lesezugriff auf den Secondary Server.

- **RSS (Remote Standalone Secondary)**
Die RSS-Funktion ist eine 1:n-Primary-/Secondary-Replikation mit Lesezugriff auf den Secondary Server.
- **SDS (Shared Disk Secondary)**
Die sekundären SDS-Server laufen auf separaten Rechnern. Sie verfügen über eigene Buffer-Pools, halten jedoch keine lokalen Daten vor und greifen lesend direkt auf die Plattenbereiche der primären Instanz zu.
- **ER (Enterprise Replication)**
Mit der Enterprise Replication können einzelne Tabellen auf verschiedene Tabellen von an der Replikation teilnehmenden Datenbank-Servern verteilt werden.

UPDATABLE Secondary

Mit der Version 11.50 hat IBM in den Informix Server bereits die Funktion des UPDATABLE Secondary integriert. Applikationen können auf sämtlichen sekundären Servern DML Statements absetzen. Jedoch bestand in der Version 11.50 das Problem, dass Transaktionen auf den sekundären Servern bei einem Absturz des primären Server unweigerlich verloren waren.

Diesem Problem wurde in der neuen Version 11.70 Abhilfe geschaffen. Die Arbeitsweise des UPDATABLE Secondary wird durch die Abbildung 1 veranschaulicht. Diese Funktion kommuniziert jede Änderung mit dem aktuellen Primary Server, der die Schreibaktivitäten erledigt. Die Transaktionen werden in den logical logs protokolliert, um sie im Falle eines Ausfalls wiederherstellen oder zu rückfahren zu können.

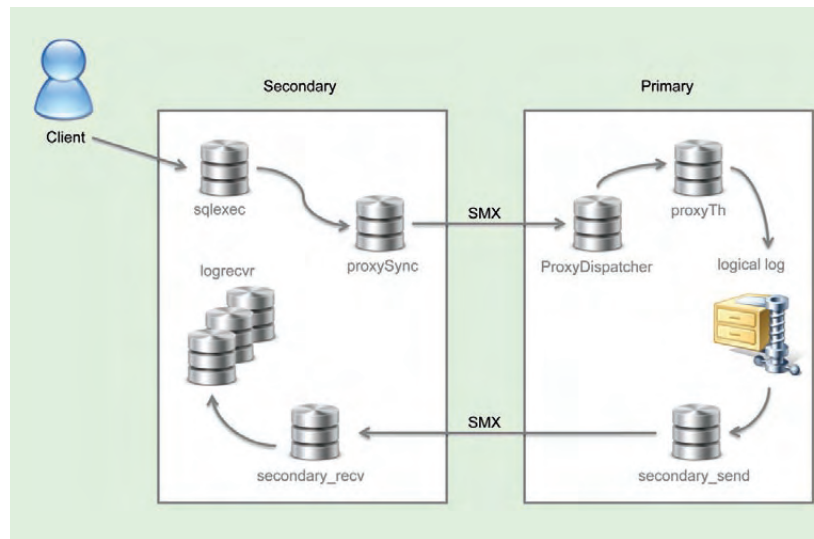


Abb. 1: Arbeitsweise UPDATABLE Secondary.

Transaction survival

Die neue Funktion „Transaction survival“ wurde mit der Version 11.70.xC2 in Informix integriert. Diese Funktion erlaubt es, dass gestartete Transaktionen auf einem UPDATABLE Secondary Server nach einem Ausfall eines Informix Primary Server übernommen und auf einem neuen Server weiter ausgeführt werden.

Ein Beispiel für die Funktionsweise wird in Abbildung 2 dargestellt. Sämtliche Server-Typen verbinden sich erneut mit dem neuen Primary Server. Dabei ist es völlig irrelevant, um welche HV-Lösung es sich handelt (HDR, RSS, SDS).

```
10:10:16 SMX thread is exiting because of network error code -25582
10:10:17 Updates from secondary currently not allowed
10:10:17 Updates from secondary currently not allowed
10:10:17 SDS: Lost connection to psds
10:11:28 Switching primary to sds2
10:11:28 DR: Reservation of the last logical log for log backup turned on
10:11:28 SDS Source Node Alias Changed from (psds) to (sds2)
10:11:28 SDS: Reconnected to sds2
10:11:28 Updates from secondary allowed
10:11:28 Transaction Survival was Successful
10:11:28 Transaction Survival:Completed successfully.
```

Abb. 2: Ausgabe des MSGPATH auf UPDATABLE Secondary.

```
onmode -wf FAILOVER_TX_TIMEOUT=10
```

Abb. 3: Beispiel für eine Änderung des Parameters FAILOVER_TX_TIMEOUT.

Konfiguration von Transaction survival

Mit Hilfe des Konfigurationsparameters `FAILOVER_TX_TIMEOUT` können die Server in einem hochverfügbaren Cluster so konfiguriert werden, dass Transaktionen nach einer Funktionsübernahme (Failover) beendet werden. Dieser Parameter wird mit folgendem Eintrag in der `onconfig`-Datei aktiviert:

- `FAILOVER_TX_TIMEOUT 1`

Der Wert des Konfigurationsparameters beschreibt das maximale Zeitfenster in Sekunden, welches zwischen einem Ausfall des Primary Server und der Konvertierung eines UPDATABLE Secondary Server in den neuen Primary Server bleibt. Sollte innerhalb des vorgegebenen Wertes keine Konvertierung

```
10:11:28 Starting BldNotification
10:11:28 SDS Source Node Alias Changed from (psds) to (sds2)
10:11:28 SCHAPI: Started dbScheduler thread.
10:11:28 Transaction Survival:Starting.
10:11:28 Transaction Survival:Starting recovery threads.
10:11:28 Transaction Survival:Getting secondary server state.
10:11:28 Transaction Survival:Advancing to a common LSN.
10:11:28 Transaction Survival:Advancing to a common LSN.
10:11:28 Transaction Survival:Retrieving completed transaction lists.
10:11:28 Transaction Survival:Rebinding transactions.
10:11:28 I've got 1 txn to bind
10:11:28 I've got 0 XA txn to bind
10:11:28 Got 1 txn to bind
10:11:28 Transaction Survival:Processing completed transactions.
10:11:28 Transaction Survival:Resuming interrupted transactions.
10:11:28 Transaction Survival:Resuming normal processing.
10:11:28 Transaction Survival was Successful
10:11:28 Transaction Survival:Completed successfully.
```

Abb. 4: Ausgabe des MSGPATH (neuer Primary Server).

```
CREATE PROCEDURE insertcommit ( )
DEFINE i, a integer;
  LET i = 1;
  LET a = 1;
BEGIN;
  FOR i = 1 to 5000
    INSERT INTO cluster values (i, a);
    LET a = a+1;
  END FOR;
  COMMIT;
END PROCEDURE;
```

Abb. 5: Quellcode Prozedur insertcommit.

```
CREATE PROCEDURE insertohnebegin ( )
DEFINE i, a integer;
  LET i = 1;
  LET a = 1;
  FOR i = 1 to 5000
    INSERT INTO cluster values (i, a);
    LET a = a+1;
  END FOR;
END PROCEDURE;
```

Abb. 6: Quellcode Prozedur insertohnebegin.

Server	SDS-Secondary	SDS-Secondary	SDS-Secondary
Datenbank	Cluster (Buffered)	Cluster (Unbuffered)	Cluster (ANSI)
Prozedur:	Datensätze	Datensätze	Datensätze
insertCommit	5000	5000	5000
insertohneBegin	~4875	5000	5000

Abb.7: Ergebnisse der Prozeduren.

```
FOC sds1+hdrs1+SDS,20 (Standard Policy SDS+HDR+RSS,0)
```

Abb. 8: Beispiel für eine Failover-Konfiguration.

erfolgen, so werden alle offen stehenden Transaktionen zurückgefahren. Der Parameter sollte auf allen Servern innerhalb des Cluster auf denselben Wert gesetzt sein. Der Wert ist bei einer Neuinstallation standardmäßig auf 0 gesetzt und damit deaktiviert. Er lässt sich aber flexibel zur Laufzeit mit dem Befehl `onmode -wf` ändern (siehe Abbildung 3).

Um einen vernünftigen Wert für die hochverfügbare Umgebung zu finden empfiehlt es sich im Vorfeld einige Failover-Tests durchzuführen. Für einen erfolgreichen Failover sollte der von Informix mitgelieferte Connection Manager eingerichtet werden. Dieser wird seit der CSDK-Version 3.50 standardmäßig mit ausgeliefert. Der Connection Manager ermöglicht das (lastabhängige) Routing zu weiterhin arbeitenden Servern innerhalb des Cluster und übernimmt die Konvertierung eines UPDATABLE Secondary Server in einen Primary Server.

Was „Transaction survival“ nicht leistet

Die Funktion „Transaction survival“ stellt keine Hilfe für Transaktionen von Clients, die mit dem Primary verbunden sind, dar. Die Transaktionen auf einem Primary Server sind unweigerlich verloren. Es besteht auch nicht die Möglichkeit den Primary Server neu zu starten und die Transaktionen von den Secondary Servern weiter arbeiten zu lassen.

In der Praxis

Transaktionen waren in der Version 11.50 unweigerlich verloren, daher erhält diese Funktion in der Praxis eine große Bedeutung. Durch das „Überleben“ der Transaktionen auf sekundären Servern sind diese nun nicht nur für Reporting-Abfragen, sondern auch für normale DML-Arbeiten geeignet. Die Abbildung 4 zeigt eine erfolgreiche Anwendung des „Transaction survival“ und die Ausgabe der Log-Datei bei einem Ausfall eines Primary Server. Dort ist erkennbar, dass Informix eine Transaktion mit Hilfe des „Transaction survival“ erkennt und übernimmt.

Für einen Test wurden zwei Prozeduren (siehe Abbildung 5 und 6) angelegt. Diese Prozeduren wurden innerhalb einer hochverfügbaren SDS-Cluster-Umgebung gegen einen UPDATABLE Secondary Server ausgeführt. Die Ergebnisse dieses Tests sind in Abbildung 7 dargestellt. Es ist zu erkennen, dass die Prozedur `insertcommit` auf den verschiedenen Datenbanken immer die volle Anzahl an Datensätzen in die Datenbank einfügt. Der Logging-Modus ist dabei nicht relevant.

Die Prozedur `insertohnebegin` zeigt ein unvoreilhaftes Verhalten bei den Datensätzen in der Datenbank „Cluster (Buffered)“. Diese Datenbank arbeitet im Buffered

Logging und verarbeitet die Datensätze vorerst im Speicher. Während eines Failover durch das „Transaction survival“ gehen die noch nicht im Logfile gespeicherten Transaktionen verloren. Daher ist dieser Logging-Modus aus Sicherheitsaspekten in einer hochverfügbaren Datenbankumgebung nicht zu empfehlen.

Empfehlung für die Benutzung

Für sämtliche Server innerhalb des Cluster wird empfohlen, den gleichen Wert für den Parameter `FAILOVER_TX_TIMEOUT` zu verwenden. Als Reihenfolge für die Wahl bei einem Failover wird SDS, HDR und dann RSS empfohlen. Eine Konfiguration für eine Failover-Richtlinie wird in Abbildung 8 gezeigt. Diese Einstellung ist in der Konfigurationsdatei des Connection Manager vorzunehmen. Wenn die Funktion „Transaction survival“ im Zusammenhang mit HDR benutzt wird, wird empfohlen, den Parameter `DRINTERVAL` auf `-1` zu setzen. Dadurch werden die Logs synchron übertragen.

In einer produktiven Umgebung empfiehlt sich ausschließlich die Verwendung von Unbuffered oder ANSI als Logging-Modus für die Datenbank. Die Verwendung von Buffered Logging ist nicht akzeptabel, wenn die Übernahme der Transaktionen gewährleistet werden soll. Zudem empfiehlt sich die Verwendung eines Connection Manager in Verbindung mit dieser neuen Funktion. Dadurch werden Failover-Konfigurationen (siehe Abbildung 9) effektiv und effizient umgesetzt und der Administrator muss lediglich den Vorgang überwachen.

Fazit

Die Einrichtung der neuen Funktion „Transaction survival“ geht wie gewohnt einfach und unkompliziert. Die Funktion kann problemlos im laufenden Betrieb ein- und ausgeschaltet werden. Es darf mit Spannung erwartet werden, ob IBM die kommenden Informix Versionen um weitere Hochverfügbarkeitsfunktionen erweitern wird und den Konkurrenzkampf zu den Mitbewerbern weiter aufleben lässt.



Carsten Hummel
(info@ordix.de).

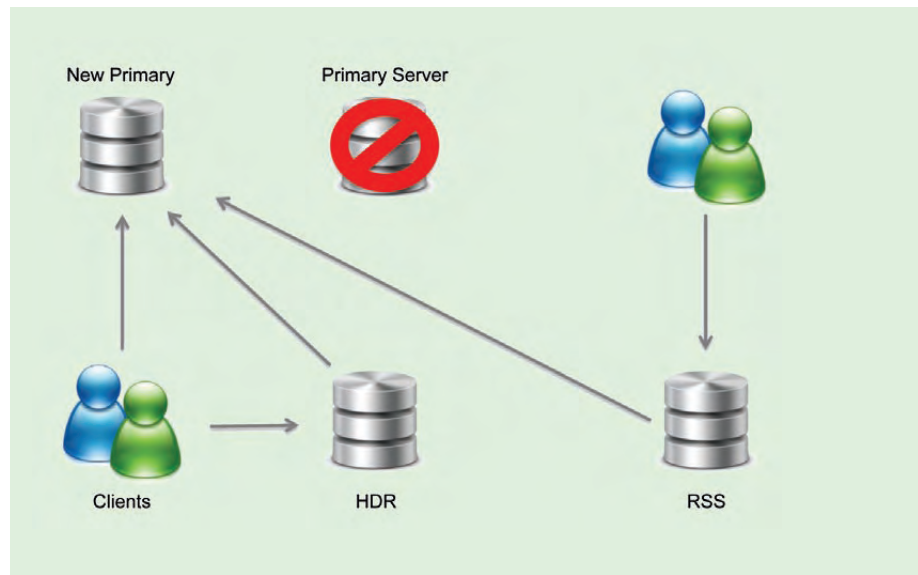


Abb. 9: Beispiel eines Failover in einer HV-Umgebung.

Links

- ▶ [1] Dokumentation Transaction survival:
http://publib.boulder.ibm.com/infocenter/idshelp/v117/index.jsp?topic=%2Fcom.ibm.admin.doc%2Fids_admin_1321.htm
- ▶ [2] IBM Informix – New Features in Informix 11.70:
<http://www.ibm.com/developerworks/wikis/download/attachments/137528140/IFMX-Infomix1170-Features-TechWhitePaper-v20.pdf?version=1>
- ▶ [3] ORDIX News Artikel 2/2009: „IBM Informix 11 (Teil IV): Shared Disc Secondary“
http://www.ordix.de/ORDIXNews/2_2009/Datenbanken/share_disc_secondary.html
- ▶ [4] IBM Redbook Extending Availability and Replication:
<http://www.redbooks.ibm.com/redbooks/pdfs/sg247488.pdf>

Glossar

CSDK	Client Software Development Kit – Es stellt einzelne Pakete bereit, die dafür optimiert sind, Anwendungen für Informix Server zu entwickeln.
ER	Enterprise Replication – Eine spezielle Technologie, die der Datenreplikation dient. Beliebig viele Datenbank-Server können als Replikat eingebunden werden.
HDR	High Availability Data Replication – Eine weitere Replikationsart unter Informix.
MACH 11	Multinode Active Cluster for High Availability – Eine äußerst flexible Hochverfügbarkeitstechnologie, die eine beliebige Kombination der Funktionalitäten HDR, RSS und SDS ermöglicht.
RSS	Remote Standalone Secondary – Eine Erweiterung des HDR-Verfahrens auf mehrere Standby-Datenbank-Server.
SDS	Shared Disc Secondary – Hierbei haben alle teilnehmenden Datenbank-Server Zugriff auf einen gemeinsamen Plattenbereich.

Wir geben DENKANSTÖSSE auf der DOAG Konferenz 2011!



DOAG Schulungstag: Oracle PL/SQL Tuning

In dieser eintägigen Schulung wird aufgezeigt, wie die Laufzeiten von PL/SQL-Applikationen deutlich verkürzt werden können. Dabei werden zum einen nützliche Features und zum anderen Tipps und Tricks aus der Praxis zur Performancesteigerung von PL/SQL-Applikationen dargestellt.

Seminarinhalte:

- Datentypen
- Coding
- Compiler-Optionen
- XMLType-Speicherungsformen
- SQL-Tuning
- Tools zur Performanceanalyse
- Tipps & Tricks

Zeit/Ort:

18. November 2011
09:00 - 16.00 Uhr
CongressCenter Ost, Nürnberg
Referent: Markus Fiegler

Preis:

450 € zuzügl. MwSt.

Melden Sie sich gleich an unter:
<http://www.doag.org/konferenz/doag/2011/>

2011
DOAG
Konferenz + Ausstellung