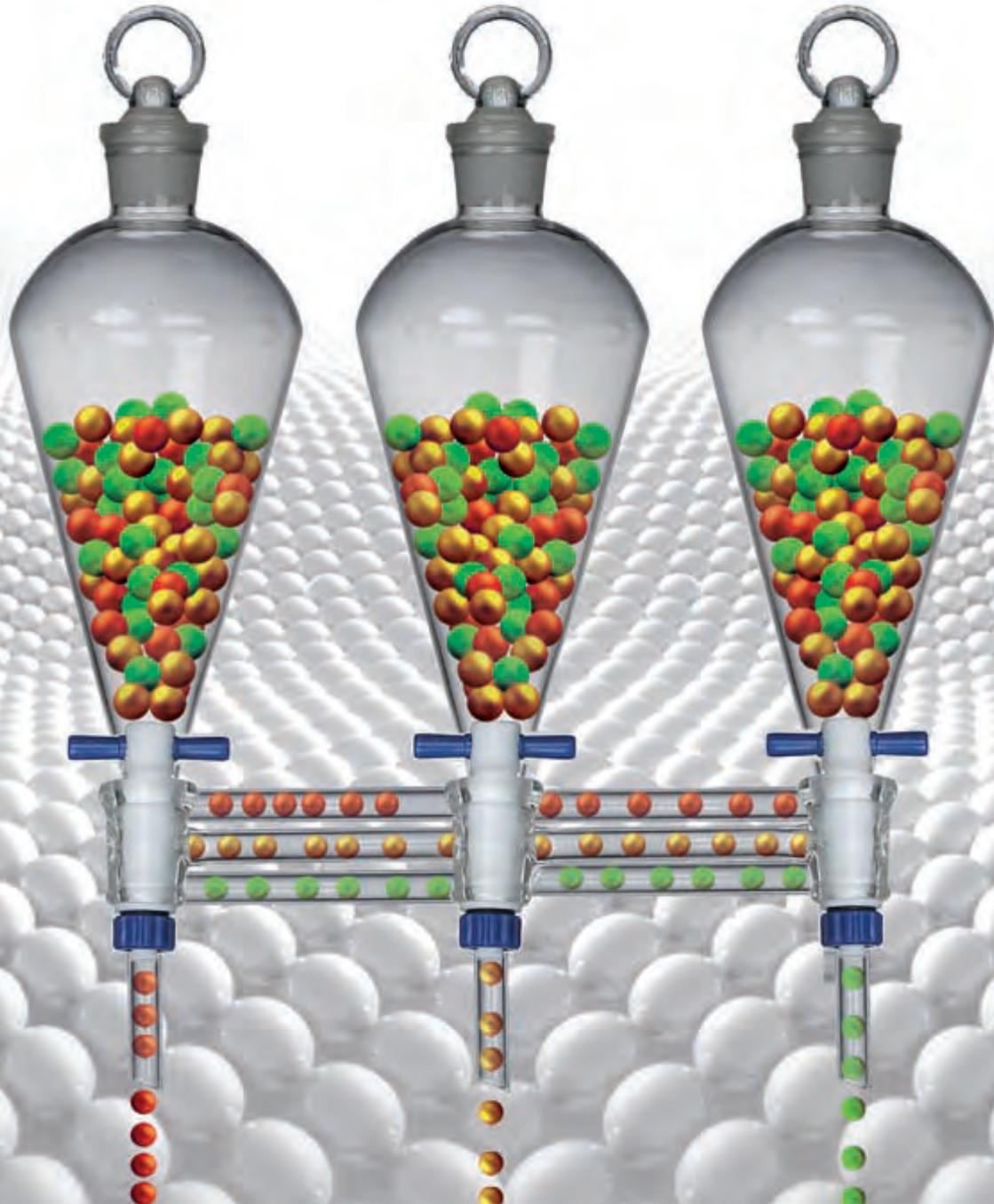


# ORDIX news

Das IT-Magazin der ORDIX AG



## Geordneter Einzug dank ETL-Prozessen

24 | ETL im Data Warehouse Umfeld

- 6 | Transient Logical Standby  
Getarntes Rolling Upgrade
- 36 | Concurrency - Lästige Konkurrenz  
Multi-Threading Programmierung mit Java

- 30 | Prozessoptimierung mit CMMI  
Steigerung der Produkt- & Servicequalität
- 41 | Informix: Administration auf grafisch  
Performance-Analyse mit dem OAT

# IT-Management ERFOLGREICH GESTALTEN

Moderne IT-Technologien sind bereits heute das Fundament für Ihren Unternehmenserfolg. Über die strategische Ausrichtung Ihrer IT für die Zukunft entscheidet immer mehr die Qualität Ihres IT-Managements.

Wir unterstützen Sie in allen wichtigen Bereichen des IT-Managements mit umfassenden Dienstleistungen unseres Partners der coniatos AG.

## Unsere IT-Management-Seminare

Wir zeigen Ihnen mit unseren Seminaren passende Wege auf bei der Entwicklung Ihres IT-Managements:

- IT-Architekturen (3 Tage)
- E-Business (2 Tage)
- IT-Strategien (2 Tage)
- Systemische Strategieentwicklung (3 Tage)
- IT-Management (5 Tage)
- IT-Risikomanagement (3 Tage)
- IT-Prozessmanagement (5 Tage)

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

Fordern Sie die Seminarbroschüre an unter [training@ordix.de](mailto:training@ordix.de)

# Weihnachtszeit – Geschenkezeit

Paderborn, Dezember 2010

Nun erhoffen sie sich wieder kurz vor Weihnachten alle das große Geschäft mit i-, androiden oder anderen (Smart-) Phones. Mich hat es vor einem Jahr erwischt. Wie Sie ja wissen, laufe ich oft wie ein Hubschrauberpilot <sup>1</sup> durch die Gegend. Nein eigentlich nicht um zu telefonieren, sondern nur um meinem Hobby „Musik hören“ zu frönen.

Aber dies nahm meine Frau letztes Jahr zum Anlass, meinen iPod gegen ein iPhone zu tauschen. Hier nun mein 12-Monate Erfahrungsbericht: Während die Handys früher immer kleiner wurden und die Dauertelefonierer in den ICE und IC Zügen dieser Republik deshalb meinten, die größere Entfernung zwischen Mund und Mikrofon mit deutlich höherer Lautstärke auszugleichen, setzen Smartphones einen neuen Trend. Wollte jemand zu diesem Zeitpunkt Industriespionage betreiben, so brauchte er eigentlich nur ständig von Hamburg nach München und zurück zu fahren und die Ohren spitzen, jetzt muss er nur ein ungeschütztes Smartphone klauen, um an unternehmenswichtige Daten zu geraten.

Nun war ich gezwungen ein ausgewachsenes Smartphone von der Größe eines Siemens Handys von 1996 in die Taschen meiner Hosen, Anzüge oder Hemden zu packen. Meistens steckte ich es in die Brusttasche meines Hemdes, was dazu führte, dass es mir binnen der ersten 3 Wochen mindestens fünfmal aus ca. einem Meter Höhe vorzugsweise auf Steinboden, Parkett oder Beton herabfiel.

Test bestanden oder Glück gehabt? Das iPhone blieb jedenfalls unbeschadet. Als dann begann ich trotzdem nach Schutzhüllen für das kostbare Stück zu suchen und stieß auf die Variante Schreiner: Individuelle Holzschale. Seither ist das Handy noch größer und mir noch öfter herunter gefallen, aber dank der inzwischen zweiten, auch schon leicht ramponierten Hülle, immer noch ohne echten Schaden. Selbst beim Stolpern über einen unmotiviert am Boden angebrachten Türstopper und einen 4-fach Looping aus der erwähnten Brusttasche blieb das Ding heile.

Inzwischen hatte mich das App-Fieber gepackt. Allerdings ergaben sich dadurch Situationen, in denen ich länger nach der App suchte, als es gebraucht hätte, die Information ohne App herauszufinden. Bei 9 Bildschirmseiten mit jeweils 16 Minifenstern übersieht man schon mal das gesuchte Objekt der Begierde und immer wenn ich die Bildschirmseiten nach Themen organisiert hatte, schwupp fand ich die 17. App für diese Seite.

Auf dieses Problem waren anscheinend auch die Apple Entwickler gestoßen und sie erfanden den Ordner und das Pseudo Multitasking. Welch gigantische Neuerfindung der Apple Ingenieure, hatte nicht ca. 1991 eine Firma namens Microsoft mit Windows 3.0 ähnlich geniale „Erfindungen“ gemacht? Aber immerhin seit iOS4.0 komme ich mit immerhin 5 übersichtlichen Seiten und einigen Ordnern aus.

Ach so telefonieren, ja also am häufigsten telefoniere ich mit meiner Mailbox, nicht weil dort ständig Anrufe eingehen, sondern vielmehr weil meine Hände auch für dieses Phone noch etwas zu groß sind und ich versehentlich beim Umklammern des Handys die Voicebox Taste auslöse. Glücklicherweise ist dieser Anruf bei meinem Tarif umsonst.

Auch das normale Telefonieren fällt mir nicht so leicht, mag es nun an der Inkompatibilität meiner Finger zur Tastaturgröße liegen oder der Blickwinkel mit der Gleitsichtbrille mich irre leiten. Auf jeden Fall hätte ich meinen Bekanntenkreis mindestens verdoppeln können, hätte ich nur die Hälfte meiner fehlgeleiteten Anrufe genutzt, um neue Bekanntschaften zu knüpfen.

Daran, dass ich das Handy oft zwei- bis dreimal pro Tag aufladen muss, habe ich mich auch wieder gewöhnt, das war bei meinem Siemens Handy von 1996 auch nicht viel anders. Dafür habe ich mir einen Akkuschnelllader gekauft, der nur doppelt so dick aber dafür halb so groß ist wie mein iPhone. Der Vorteil ist, beide passen nur noch in die Hosentasche aber nicht mehr in die Brusttasche, sieht zwar nicht ganz so schick aus, aber besser als mitten im wichtigen Gespräch einen Abbruch zu haben und darauf zu warten bis der Akku wieder bei 5% Ladekapazität ist. Vorher geht nämlich das smarte Phone nicht an.

Ich sehe, ich muss einen News Artikel über das iPhone schreiben, denn der Platz fürs Editorial reicht nicht aus. Bieten die Artikel dieser News zwar hervorragende Parallelen zu meinen Smartphone Erfahrungen, bleibt mir dennoch nur der Platz für die besten Wünsche zur Weihnachtszeit und für 2011. In der letzten News 2009 habe ich Ihnen ein weniger krisengesätteltes 2010 gewünscht, den eingetretenen Aufschwung konnte man ja kaum verkraften. Vielleicht geht es so weiter, egal ob Sie nun ein iPhone zu Weihnachten bekommen oder nicht, wir werden die News b.a.w auch ohne App anbieten.

Ihr



Wolfgang Kögler

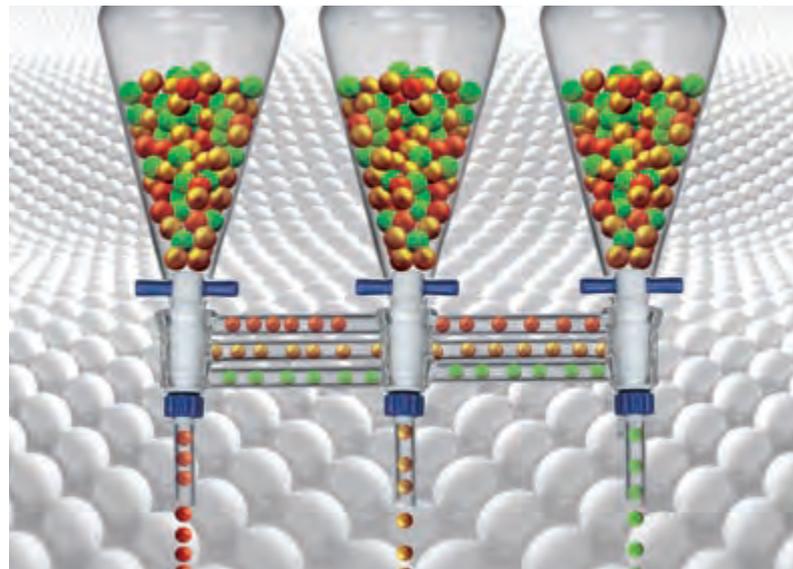


Wir wünschen  
allen Lesern ein  
geruhames  
Weihnachtsfest  
und einen guten  
Start ins Jahr 2011!

<sup>1</sup> Siehe Editorial in der ORDIX news 2/2007



6 | Transient Logical Standby - Getarntes Rolling Upgrade



24 | Geordneter Einzug dank ETL-Prozessen

## Java/JEE

- 27..... **Integration mit Launch4j - der letzte Schliff**  
Das Starten von Java-Anwendungen unter Windows Systemen verursacht sehr oft Probleme. Das OpenSource Werkzeug Launch4j bietet hier Abhilfe.
- 36..... **Java Best Practice (Teil VIII): Concurrency - Lästige Konkurrenz (I)**  
Parallele Zugriffe auf Objekte verursachen sehr oft Probleme. Dieser Artikel aus der Java Best Practice-Reihe liefert Ihnen einen ersten Einblick in die Multi-Threading Programmierung.

## Open Source

- 10..... **Synchrone Server mit Csync2 - Cluster und Server IN-Sync**  
Dieser Artikel beleuchtet die Möglichkeiten, die das Open Source Werkzeug Csync2 bietet um Änderungen auf mehreren Systemen zu vollziehen.
- 14..... **Einfache Anbindung mit dem Nagios::Plugin-Modul**  
Die freie Monitoring-Lösung Nagios wird immer populärer. Der Artikel beschreibt, wie mit diesem Modul nicht nur Basisinformationen geliefert werden können.

## IT-Strategie

- 30..... **IT-Management (Teil IV): Prozessoptimierung mit CMMI Sind Ihre Prozesse schon reif?**  
Der vierte Teil der Reihe erläutert, wie mit Hilfe von Capability Maturity Model Integration, die Qualität Ihrer Prozesse verbessert werden kann.

## Datenbanken

- 6..... **Minimierung der Ausfallzeiten beim Upgrade von Oracle Datenbanken (Teil II): Transient Logical Standby - getarntes Rolling Upgrade**  
Logical oder Physical Data Guard? Dieser Artikel stellt das Upgrade mit Hilfe des Physical Data Guard vor und vergleicht es mit dem Upgrade des Logical Data Guard.
- 17..... **Oracle Audit Vault – Das zentrale Data Warehouse für Audit-Daten**  
Sensible Audit-Daten sollen zentral konsolidiert, geschützt und sinnvoll ausgewertete werden. Audit Vault ist hierfür das geeignete Produkt.
- 24..... **ETL im Data Warehouse am Beispiel IBM DataStage (Teil I): Geordneter Einzug dank ETL-Prozessen**  
Der erste Teil der neuen Reihe erläutert die Definition der ETL-Prozesse in dem IBM Produkt.
- 41..... **Informix IDS V11.5 Das OpenAdmin Tool - IDS-Administration auf grafisch!**  
Am Beispiel der Performance-Analyse wird das web-basierte Tool zur Verwaltung, Überwachung und Analyse vorgestellt.





30 | Prozessoptimierung mit CMMI



36 | Concurrency - Lästige Konkurrenz

## Aktuell

- 21..... Treffpunkt RheinMain IT**  
Was haben Projektmanager und Manager gemeinsam?
- 21..... Rückblick auf den 55. IUG Workshop**
- 34..... DOAG Konferenz und Schulungstag 2010**  
Rückblick auf das Event des Jahres für die Oracle Community

## Standards

- 03..... Editorial**
- 04..... Inhalt**
- 05..... Impressum**
- 13..... Larry Ratlos**
- 22..... Seminarübersicht:**  
Dezember 2010 bis Mai 2011

## 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:** 9.900

**Druck:** Media-Print Informationstechnologie GmbH, Paderborn

### Bildnachweis:

- © piqs.de / Tamung ist alles © Cindi
- © sxc.hu / ball waves 2 © gerard79
- © aboutpixel.de / House Building © alexsl
- © sxc.hu / Cup 1 black cup © bingss
- © sxc.hu / red apple © lockstockb
- © sxc.hu / Glove Love © alfredo-9
- © sxc.hu / Traffic light for bikes © boroda003
- © sxc.hu / Sometimes you just have to shine © m3 fs
- © istockphoto.com / Closeup of Golfer Sinking a Putt © Cardston

### Autoren dieser Ausgabe:

Andreas Beyer, Evelyn Ernst, Christian Fertsch, Kathleen Hock, Dr. Stefan Koch, Wolfgang Kögler, Norbert Munkel, Jens Pothmann, Vanessa Prior, Matthias Schneider, Werner Wellmann

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

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

### Copyright:

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

### Haftung:

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

### Warenzeichen:

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



Minimierung der Ausfallzeiten beim Upgrade von Oracle Datenbanken (Teil II)

# Transient Logical Standby - getarntes Rolling Upgrade

Dieser Artikel richtet sich an Datenbankadministratoren, die ein Upgrade-Verfahren mit minimaler Ausfallzeit suchen.

In der ORDIX News 1/2010 [1] haben wir das Rolling Upgrade mit Hilfe des Logical Data Guard vorgestellt. Im vorliegenden Artikel betrachten wir nun das Rolling Upgrade mit Hilfe eines Physical Data Guard. Mit der Oracle Version 11gR1 wurde diese Funktionalität verfeinert. Wir erläutern, welche Einschränkungen es beim Rolling Upgrade mit Physical Data Guard gibt und weshalb der Logical Data Guard immer noch Anwendung findet.

## Grobkonzept des Rolling Upgrade mit Physical Data Guard

Wie ist ein Rolling Upgrade mit einem Physical Data Guard überhaupt möglich? Kann es funktionieren, dass der Physical Data Guard kurzzeitig mit einem anderen Release gefahren wird als die Primärdatenbank?

Nein! Denn ein Physical Data Guard ist eine blockidentische Kopie der Primärdatenbank. Das Data Dictionary spielt hier die wichtigste Rolle. Dieses ist im Konstrukt des Physical Data Guard immer identisch zu dem der Primärdatenbank.

Während eines Rolling Upgrade besitzt jedoch die Standby-Datenbank aufgrund der

Aktualisierung kurzzeitig ein anderes Data Dictionary als die Primärdatenbank. Somit scheidet ein direktes Rolling Upgrade mit Physical Data Guard aus.

Oracle bietet eine elegante Lösung, indem der Physical Data Guard temporär in einen Logical Data Guard konvertiert wird. Danach kann das eigentliche Rolling Upgrade durchgeführt werden. Schließlich wird der Logical Data Guard zurück in einen Physical Data Guard konvertiert. Dieses Konzept wird Transient Logical Standby genannt. Beim Rolling Upgrade mit Logical Data Guard können allerdings nicht alle Datentypen (z. B. XMLType) übertragen werden.

Somit müssen wir dieselben Einschränkungen einplanen, die auch mit dem Logical Data Guard gegeben sind.

### Vorgeschichte

Ein Rolling Upgrade mit Hilfe des Physical Data Guard ist nicht erst seit Oracle 11g möglich. Vor der Version 11g war die Umsetzung des Rolling Upgrade mit Physical Data Guard jedoch erheblich aufwändiger. Nachdem der Physical Data Guard in einen Logical Data Guard konvertiert wurde, war der Weg zurück nur mit vielen Zusatzschritten möglich. Mit Oracle 11g hat sich dieser Rückweg immens vereinfacht.

### Transient Logical Standby en detail

Das eigentliche Rolling Upgrade mit dem Logical Data Guard wurde bereits im Artikel „Rolling Upgrade mit Logical Data Guard“ [1] betrachtet. Daher erfolgt in diesem Artikel nur der detaillierte Ablauf des Rolling Upgrade mit dem Physical Data Guard.

Das Konzept der Transient Logical Standby lässt sich grob in fünf Schritte unterteilen (siehe Abbildung 1).

1. Zu Beginn wird die Primärdatenbank für das Rolling Upgrade mit Transient Logical Standby vorbereitet.
2. Daraufhin wird der Physical Data Guard zum Logical Data Guard konvertiert.
3. Das Rolling Upgrade mit anschließendem Rollentausch wird durchgeführt.
4. Danach wird ein Flashback Database des neuen Logical Data Guard durchgeführt.
5. Schlussendlich kann die Rückkonvertierung zum Physical Data Guard erfolgen.

Diese fünf Schritte beschreiben wir nun detailliert:

### Vorbereitungen auf der Primärdatenbank

Um die Primärdatenbank auf ein Rolling Upgrade mit Transient Logical Standby vorzubereiten, bedarf es einiger Vorkehrungen.

Zusätzlich zum ArchiveLog-Modus, der bei einer Standby-Datenbank aktiviert werden muss, muss die Funktionalität Flashback Database eingeschaltet werden.

Dazu definieren wir die beiden Parameter DB\_RECOVERY\_FILE\_DEST\_SIZE und DB\_RECOVERY\_FILE\_DEST.

Mit diesen Parametern legen wir das Verzeichnis und die maximale Größe der Recovery-File-

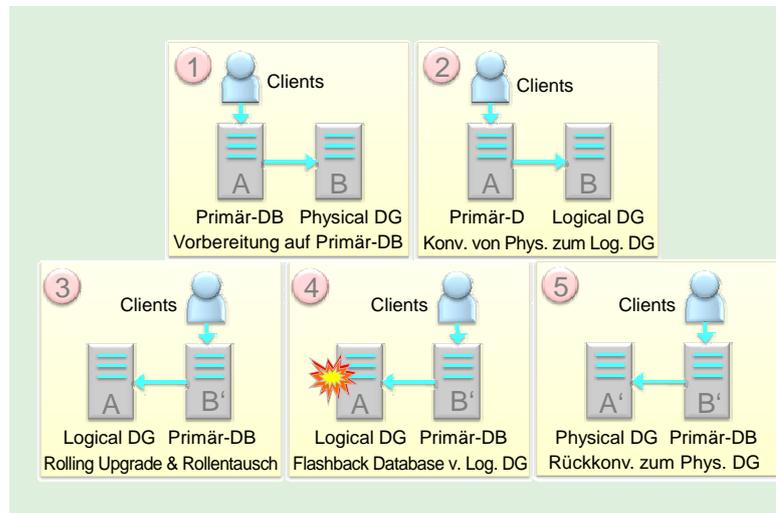


Abb. 1: Ablauf Transient Logical Standby.

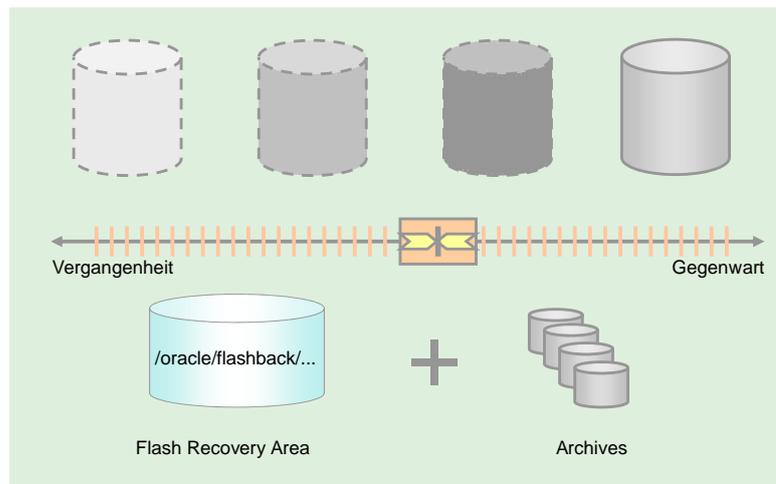


Abb. 2: Flashback-Datenbase.

Destination fest, in das die so genannten FlashbackLogs geschrieben werden.

Mit Hilfe der RedoLog- bzw. ArchiveLog-Dateien und FlashbackLog-Dateien ist es möglich, die Datenbank auf jeden beliebigen Zeitpunkt in einem definierten Zeitfenster in die Vergangenheit zurück- und wieder in die Gegenwart vorzufahren (siehe Abbildung 2).

Der Parameter DB\_FLASHBACK\_RETENTION\_TARGET gibt den Zeitraum an, bis zu dem die Datenbank in die Vergangenheit zurückgefahren werden kann. Der Standardwert beträgt 1440 Minuten, dieses entspricht einem Tag. Dieser Zeitraum definiert das Zeitfenster, bis zu dem wir die Datenbank zurück in die

```
ALTER DATABASE FLASHBACK ON;
```

Abb. 3: Befehl zur Aktivierung der Flashback Funktionalität.

```
CREATE RESTORE POINT vorher GUARANTEE FLASHBACK DATABASE;
```

Abb. 4: Befehl zum Erstellen eines Guaranteed Restore Point.

```
ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/tmp/stby.ctl';
```

Abb. 5: Befehl zum Erstellen eines Standby-Controlfiles.

```
ALTER DATABASE RECOVER TO LOGICAL STANDBY KEEP IDENTITY;
```

Abb. 6: Konvertierung von Physical Data Guard zu Logical Data Guard.

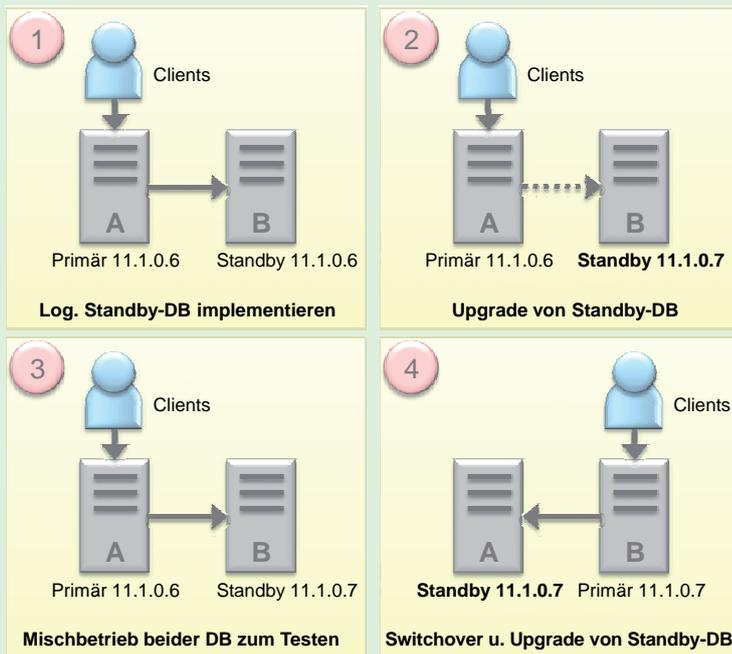


Abb. 7: Ablauf des Rolling Upgrade mit Logical Data Guard.

```
FLASHBACK DATABASE TO RESTORE POINT vorher;
```

Abb. 8: Flashback des Logical Data Guard.

Nachdem wir Flashback Database aktiviert haben, können wir ein Guaranteed Restore Point erzeugen (siehe Abbildung 4).

Mit Hilfe dieses Guaranteed Restore Point ist gewährleistet, dass ein Zurückfahren der Datenbank bis zu diesem Zeitpunkt möglich ist.

Wenn wir einen normalen Restore Point anstelle eines Guaranteed Restore Point verwenden, wäre nicht sichergestellt, dass dieser Restore Point zum Zeitpunkt des Flashback Database noch existiert. Sobald das Verzeichnis DB\_RECOVERY\_FILE\_DEST vollgeschrieben ist, werden alte FlashbackLog-Dateien gelöscht. Wurde ein Guaranteed Restore Point erzeugt, werden die alten FlashbackLog-Dateien nicht gelöscht, sondern die Datenbank bleibt stehen. Um dieser Situation vorzubeugen, ist es wichtig, dass die Flash Recovery Area groß genug dimensioniert ist.

Des Weiteren wird für die spätere Rückkonvertierung zum Physical Data Guard ein Standby-Controlfile benötigt. Dieses wird mit dem Befehl in Abbildung 5 erzeugt.

### Konvertierung des Physical Data Guard zum Logical Data Guard

Sobald alle Vorbereitungen auf der Primärdatenbank durchgeführt worden sind, kann der Physical Data Guard zu einem Logical Data Guard konvertiert werden (siehe Abbildung 6).

Die KEEP IDENTITY-Klausel ist erst mit Oracle 11g verfügbar und sorgt dafür, dass der DB\_NAME und die DBID identisch zur Primärdatenbank bleiben. Dieser Mechanismus wurde speziell für die Situation der Transient Logical Standby entwickelt und lässt nur einen Rollentausch zu.

Sobald die Konvertierung abgeschlossen ist, kann der SQL-Apply und somit die Übertragung der Informationen von der Primärdatenbank an die logische Standby Datenbank gestartet werden.

### Rolling Upgrade des Logical Data Guard mit Rollentausch

Das eigentliche Rolling Upgrade mit dem Logical Data Guard kann jetzt durchgeführt werden. Hierzu wird zuerst der SQL-Apply gestoppt, der Logical Data Guard auf das neue Oracle Release aktualisiert und das SQL-Apply wieder gestartet. Daraufhin wird

Vergangenheit fahren können. Über die Kommandozeile wird Flashback Database im Zustand Mount (siehe Abbildung 3) aktiviert.

ein Switchover durchgeführt und die Rollen werden getauscht.

Der Logical Data Guard wird nun zur Primärdatenbank und die Primärdatenbank wird zum Logical Data Guard (siehe Abbildung 7). Dieses Verfahren wurde bereits im ORDIX News Artikel 1/2010 [1] näher vorgestellt.

### Flashback des neuen Logical Data Guard

Um zurück zur Ausgangssituation zu gelangen und den Logical Data Guard zurück in einen Physical Data Guard zu konvertieren, wird der neue Logical Data Guard auf den Zeitpunkt vor dem Rolling Upgrade zurückgesetzt (siehe Abbildung 8).

Die Datenbank wird auf den Guaranteed Restore Point zurückgesetzt. Dazu wird die Datenbank auf den letzten Checkpoint vor dem Guaranteed Restore Point zurückgerollt. Da der Stand der Datenbank jedoch zu weit in der Vergangenheit liegt, wird die Datenbank mit Hilfe der DO und UNDO-Informationen aus den ArchiveLog-Dateien exakt zum gewünschten Zeitpunkt (dem Guaranteed Restore Point) vorgerollt.

### Rückkonvertierung zum Physical Data Guard

Jetzt wird der Logical Data Guard zu einem Physical Data Guard zurückkonvertiert. Hierzu wird das Standby-Controlfile aus Schritt 1 genutzt. Dieses wird auf das Standby-System kopiert. Sinnvollerweise wurde zuvor die neue Software, die auch auf der aktuellen Primärdatenbank eingespielt wurde, auf dem Server des aktuellen Logical Data Guard installiert.

Das \$ORACLE\_HOME der Datenbank setzen wir jetzt auf das neue \$ORACLE\_HOME um, in dem die Software installiert wurde. Die Standby-Datenbank wird eingebunden. Der Recovery-Prozess, der die Informationen von der Primärdatenbank an die Standby-Datenbank liefert, wird neu gestartet (siehe Abbildung 9).

Die Aktualisierung der Standby-Datenbank erfolgt jetzt automatisch mit dem Start des Recovery über den Redo-Apply der Primärdatenbank an die Standby-Datenbank. Alle Informationen der Primärdatenbank werden an den Physical Data Guard gesendet, sodass beide Datenbanken wieder ein identisches Data Dictionary besitzen. Somit wurde der neue Physical Data Guard implizit aktualisiert.

```
RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT;
```

Abb. 9: Recovery starten.

#### Glossar

<b>Flashback Database</b>	Eine Technik, ein Point in Time Recovery einer Oracle Datenbank ohne das Einspielen eines Backups durchzuführen.
<b>SQL-Apply</b>	Die SQL-Statements, die aus den RedoLog-Dateien der Primärdatenbank gewonnen werden, werden auf dem Logical Data Guard nachgezogen und neu ausgeführt. Dieses Verfahren dient der Informationsübertragung im Data Guard Umfeld.

#### Links

- ▶ [1] ORDIX News Artikel 1/2010 „Rolling Upgrade mit Logical Data Guard“:  
[http://www.ordix.de/ORDIXNews/1\\_2010/Datenbanken/rolling\\_upgrade\\_logical\\_data\\_guard.html](http://www.ordix.de/ORDIXNews/1_2010/Datenbanken/rolling_upgrade_logical_data_guard.html)
- ▶ [2] ORDIX News Artikelreihe „Oracle Data Guard“:  
<http://www.ordix.de/ORDIXNews/artikelreihen.html#dataguard>
- ▶ [3] Seminarempfehlung „Oracle Migration und Patching“:  
<http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/840/index.html>

### Zusammenfassung

Ein Rolling Upgrade mit Physical Data Guard bringt nicht weniger Einschränkungen mit sich als das Rolling Upgrade mit Logical Data Guard. Der Logical Data Guard hält sich nämlich im Kern des Rolling Upgrade mit Physical Data Guard versteckt. Es liegen somit dieselben Einschränkungen (siehe ORDIX News 1/2010 [1]) und Möglichkeiten (die Ausfallzeit der Datenbank wird für den Anwender auf ein Minimum reduziert) vor, die auch mit dem Logical Data Guard gegeben sind.

#### Fazit

Das Gesamtkonstrukt der Funktionalität des Transient Logical Standby ist sehr praktisch, da sich die häufiger vorkommenden Physical Data Guard Umgebungen einfacher mit Hilfe des Rolling Upgrade aktualisieren lassen. Oracle bietet uns damit einen eleganten, schnellen Weg, um wieder zurück zur Ausgangssituation – dem Physical Data Guard – zu gelangen.

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



Vanessa Prior  
([info@ordix.de](mailto:info@ordix.de)).

# Cluster und Server IN-Sync

Dieser Artikel richtet sich an Administratoren die mehrere Server betreuen auf denen Dateien und Verzeichnisse mit gleichem Inhalt zu administrieren sind.

In Cluster- oder bestimmten Server-Umgebungen ist es oft notwendig die Konfigurationsänderungen nicht nur auf einem System, sondern auf allen beteiligten Systemen durchgeführt werden. Häufig stellt man in der Praxis fest, dass bei einem Cluster-Failover der Benutzer, der Cron-Job, die Konfigurationsdatei nicht auf allen beteiligten Systemen angelegt oder verändert wurde. Im allerschlimmsten Fall führt dies zu einem Totalausfall. In diesem Artikel möchten wir detailliert auf die Open Source Synchronisationslösung Csync2 eingehen, mit der oben genannte Probleme der Vergangenheit angehören.

## Csync2 – das Synchronisationswerkzeug

Werkzeuge um Dateien und Verzeichnisse zwischen mehreren Servern zu kopieren und aktuell zu halten gibt es wie Sand am Meer. Bekannte Vertreter hierfür sind zum Beispiel `rsync` oder `scp`. Aber eines haben die meisten Lösungen gemeinsam, man muss die Änderungen auf einem ganz bestimmten System ausführen und von dort aus findet im Anschluss die Synchronisation der Daten statt. Ändert man unwissentlich die Dateien auf einem der Zielservers, werden die durchgeführten Änderungen wieder gnadenlos überschrieben.

Die Lösung wäre ein Werkzeug, mit dem auf einem beliebigen System Änderungen durchgeführt werden könnten und diese immer auf die anderen beteiligten Systeme synchronisiert werden. Csync2 [1] beherrscht dies. Viele Synchronisationswerkzeuge haben Probleme das Löschen von Dateien korrekt zu erkennen, für Csync2 stellt dies keine Hürde dar. Csync2 erstellt auf jedem System unter dem Pfad `/var/lib/csync2/hostname.db` eine SQLite-Datenbank, in der Informationen über die lokalen von Csync2 kontrollierten Dateien enthalten sind. Die Datenbank enthält beispielsweise Informationen über Größe und Änderungszeitstempel der lokalen Dateien und wird zum Erkennen von Veränderungen herangezogen. Die Synchronisation mit den

Partnersystemen erfolgt standardmäßig über den TCP/IP Port 30865, mit Hilfe eines `inetd/xinetd`-basierten Dienstes. Abbildung 1 zeigt eine Beispielkonfiguration für den `xinetd`. Wichtig für die Konfiguration ist noch ein Eintrag in der `/etc/services`, in der Form `csync2 30865/tcp`, welcher beim Installieren von Csync2 mit RPM/DEB-Paketen automatisch eingetragen wird. Bei einer manuellen Installation muss dies selbst durchgeführt werden.

## Konfiguration von Csync2

Normalerweise erwartet Csync2 seine Konfiguration in der Datei `/etc/csync2.cfg`, es kann aber später beim Aufruf der Befehle auch die Option `-c config-name` mitgegeben werden, wobei dann eine Datei mit dem Namen `/etc/csync_config-name.cfg` erwartet wird. So können bequem für individuelle Zwecke unterschiedliche Konfigurationen abgelegt werden. Meist ist dies aber nicht notwendig, da in der Datei `/etc/csync2.cfg` auch schon das Erstellen unterschiedlicher Gruppen vorgesehen ist. Abbildung 2 zeigt eine Konfigurationsdatei.

Innerhalb der Konfigurationsdatei können beliebig viele Synchronisationsgruppen erstellt werden, wobei für einen Server nur Gruppen genutzt werden, bei denen er auch mit einem `host`-Eintrag aufgelistet ist. Mit dem

Eintrag `host` werden alle Hosts aufgelistet die in dieser Gruppe bei der Synchronisation eingeschlossen sind. Hierbei können alle Systeme in einer Zeile aufgelistet werden, oder für jedes System eine separate Zeile. Ist ein Server unter mehreren Namen erreichbar, muss hier zwingend der Name der mit dem Befehl `hostname` ausgegeben wird, angegeben werden. Die IP-Adresse, die sich hinter dem Hostnamen verbirgt wird dann auch für die Synchronisation der Daten verwendet.

Will man eine andere Netzwerkschnittstelle dafür nutzen, muss wie beim Server `suse2` der Eintrag konfiguriert werden. Der Eintrag `suse2@suse2-eth0` bedeutet: Nutze die Schnittstelle `suse2-eth0` auf dem System `suse2`, wobei zu diesem Namen wiederum ein Eintrag in der `/etc/hosts` gesucht wird. Der Server `suse3` ist ein sogenannter „Slave Host“, er bekommt zwar Änderungen übertragen, kann selbst aber keine Änderungen durchführen. Stößt man von `suse3` eine Synchronisation an, endet dies mit der Fehlermeldung: `ERROR from peer suse1; Permission denied for slave!`.

### Aber nur mit Authentifizierung und Sicherheit

Die Authentifizierung der Systeme bei der Synchronisation erfolgt zum einen mit der IP-Adresse, zum anderen mit dem Inhalt der mit dem `key`-Eintrag hinterlegten Datei. Jede Gruppe muss zwingend einen Pre-Shared-Key hinterlegt haben, der auf allen beteiligten Systemen gleich sein muss. Empfehlenswert ist die Erzeugung eines eigenen Schlüssels für jede Gruppe.

Der Schlüssel kann hierbei mit dem Befehl `csync2 -k /etc/csync2.key_apache2_group` erstellt werden.

Csync2 ist von Haus aus auf Sicherheit ausgelegt und erwartet, dass die Synchronisation zwischen den Systemen mit SSL verschlüsselt wird.

Das funktioniert aber nur wenn man im `/etc`-Verzeichnis einen SSL-Key und ein SSL-Zertifikat mit dem Namen `csync2_ssl_cert.pem` und `csync2_ssl_key.pem` ablegt. Hat man den Quellcode von Csync2, kann mit dem Befehl `make cert` beide Dateien generiert werden.

Alternativ kann man mit den in Abbildung 3 dargestellten Befehlen die notwendigen Dateien erstellen. Selbstverständlich kann auch der firmeninterne Zertifikatsserver zur Erstellung ge-

```
service csync2
{
    flags      = REUSE
    socket_type = stream
    wait       = no
    user       = root
    group      = root
    server     = /usr/sbin/csync2
    server_args = -i
    port       = 30865
    type       = UNLISTED
    only_from = 192.168.0.1 192.168.0.2
}
```

Abb. 1: Konfiguration des `xinetd` für `csync2` Netzwerksynchronisation.

```
group apache2_group
{
    host suse1 (suse3);
    host suse2@suse2-eth0;

    key /etc/csync2.key_apache2_group;

    include /etc/hosts;
    include /etc/apache2;
    include %apachehome%/wwwrun;

    backup-directory /var/backups/csync2;
    backup-generations 3;

    auto none;
}

prefix apachehome
{
    on suse[12]: /home;
    on *:       /var/lib;
}

noss1 suse1 suse2;
```

Abb. 2: Konfigurationsdatei `/etc/csync2.cfg`.

nutzt werden. Legt man auf eine verschlüsselte Übertragung der Daten keinen Wert, kann dieses mit dem `noss1`-Statement in der Konfiguration erreicht werden. Der erste Wert hinter dem `noss1`-Statement ist der Quellserver der zweite Wert der Zielsystem. Arbeitet man ohne den Platzhalter `*`, der für jedes System steht, muss für jede unverschlüsselte Verbindung ein eigener Eintrag gemacht werden.

### Include und Exclude von Dateien

Mit den Schlüsselwörtern `include` und `exclude` werden die zu synchronisierenden Dateien und Verzeichnisse angegeben. Bei der Angabe eines Dateinamens, wird nur diese eine Datei synchronisiert, wird ein Verzeichnis angegeben wird rekursiv der ganze Verzeich-

```
openssl genrsa -out /etc/csync2_ssl_key.pem 1024
openssl req -new -key /etc/csync2_ssl_key.pem -out /etc/
csync2_ssl_cert.csr
openssl x509 -req -days 365 -in /etc/csync2_ssl_cert.csr
-signkey
/etc/csync2_ssl_key.pem -out /etc/csync2_ssl_cert.pem
```

Abb. 3: Erstellung eines selbstsignierten SSL-Zertifikats für Csync2.

```
action
{
    pattern /etc/apache2/httpd.conf;
    pattern /etc/apache2/vhosts.d/*;
    exec "/etc/init.d/apache2 reload";
    logfile "/var/log/csync2_action.log";
    do-local;
}
```

Abb. 4: Action innerhalb einer Csync2-Gruppe.

nisinhalt übertragen. Mit `exclude` können tiefer gelegene Dateien wieder ausgeschlossen werden. Werden auf den Servern für die Dateien unterschiedliche Pfade als Speicherort der Dateien benutzt, kann dies mit der Definition eines `prefix` definiert werden. Über die Variable `%apachehome%` kann dieser dann beim Ein- und Ausschließen von Dateien, genutzt werden.

### Backups und Konfliktbewältigung

Wohl oder übel werden bei der Synchronisation Daten auf den Zielsevernen überschrieben. Über den Eintrag `backup-directory` wird ein Verzeichnis angegeben, in dem die alten Versionen abgelegt werden. Um die Datei später wieder zuordnen zu können, werden die Dateien so umbenannt, dass die genaue Herkunft der Datei ersichtlich bleibt. So wird z. B. aus der Datei `/etc/hosts` im Backupverzeichnis die Datei `etc_hosts.0`.

Mit dem Eintrag `backup-generations` wird die die Anzahl alter Dateien definiert, die aufgehoben werden sollen. Wenn der Eintrag `backup-generations` nicht angegeben ist, wird automatisch der Standardwert 3 benutzt.

Bei Änderungen zwischen einer Synchronisation der Dateien auf verschiedenen Servern, stellt Csync2 beim nächsten Aufruf einen Konflikt fest. Mit dem Parameter `auto` innerhalb der Konfigurationsdatei kann der Administrator bestimmen, wie sich Csync2 bei der Konflikterkennung verhalten soll.

Folgende Einträge sind möglich.

- **none**  
Konflikte werden von Csync2 nicht automatisch gelöst. Der Administrator muss manuell bestimmen, welche Version der Datei die aktuellste ist.
- **first**  
Der Server, auf dem der Abgleich der Daten mit Csync2 zuerst gestartet wird, überschreibt die Daten auf den anderen Servern.
- **younger/older**  
Der Server mit den jüngeren, bzw. mit den älteren Dateien wird für den Abgleich genutzt.
- **bigger/smaller**  
Der Server mit der größeren, bzw. kleineren Datei bestimmt die Synchronisationsrichtung.
- **left/right**  
Der Server der „links“ in der `host`-Liste steht, bzw. derjenige der „rechts“ in der Liste steht.

Möchte man keine unvorhergesehenen Überraschungen erleben, ist der Wert `none` die sicherste Variante. Üblicherweise wird der Abgleich der Daten mit Cron-Jobs automatisiert. Niedrige und auf den Servern unterschiedliche Intervalle, reduzieren die Gefahr eines Konfliktes.

### Neustart des Dienstes notwendig

Die meisten UNIX-Dienste benötigen nach Änderungen an den Konfigurationsdateien, ein Reload oder Neustart des Dienstes. Mit Csync2 ist es möglich nach der Synchronisation Befehle auszuführen. Hierzu kann in einer Gruppe ein Befehl `action` definiert werden. Abbildung 4 zeigt einen möglichen Eintrag. Werden bei der Synchronisation Dateien mit einem bestimmten `pattern` übertragen, wird anschließend der unter `exec` definierte Befehl ausgeführt. Der Befehl wird immer genau einmal ausgeführt, selbst wenn sich mehrere Dateien auf die das `pattern` zutrifft geändert haben.

Ausgaben, die bei der Befehlsausführung entstehen, landen in der mit `logfile` angegebenen Datei. Wird die `logfile`-Datei nicht angegeben, werden Ausgaben automatisch in den ewigen Papierkorb `/dev/null` umgeleitet. Mit dem Befehl `do-local` wird der `exec`-Befehl auch auf dem Server ausgeführt, auf dem die Änderungen ursprünglich durchgeführt worden sind.

## Csync2 – Der Befehl

Für die Synchronisation der Server wird nun der Kommandozeilenbefehl `csync2` genutzt. Mit dem Befehl `csync2 -xv` werden alle Dateien geprüft und Änderungen mit den Partnersystemen synchronisiert. Im Regelfall hat man schon beim allerersten Aufruf einen Konflikt. Wurde Csync2 nicht direkt nach der Installation des Servers aktiviert, unterscheiden sich wahrscheinlich die Dateien auf den Servern. Die Meldung „`While syncing file /etc/hosts:; ERROR from peer suse1: File is also marked dirty here!`“ weist auf das Problem hin.

Mit dem Befehl `csync2 -f /etc/hosts` auf einem bestimmten Server, wird diese Version als die zu Nutzende erklärt. Beim nächsten Befehlsaufruf klappt dann die Synchronisation. Hat man einen ganzen Verzeichnisbaum, der zum aktuellen Stand erklärt werden soll, ist der Befehl `csync2 -fr` notwendig. Die Entwickler haben für alle Eventualitäten Optionen eingebaut. So sind verschiedene `DryRun`-Methoden, Testsznarien oder nur das Ausführen bestimmter Gruppen möglich.

### Links

- ▶ [1] Projektseite von Csync2: <http://oss.linbit.com/csync2>
- ▶ [2] Seminarempfehlung: Linux Hochverfügbarkeits-Cluster <http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Betriebssysteme/nr/363/index.html>

### Fazit

Csync2 ist ein praktisches Kommandozeilenwerkzeug mit dem nichtsynchrone Cluster der Vergangenheit angehören. Da es keine Limitierung auf die Anzahl der Server oder Anzahl der Dateien gibt, können selbst viele Server, bzw. große Verzeichnisstrukturen zwischen Servern synchronisiert werden. Csync2 kann sowohl an der Kommandozeile als auch automatisiert per Cron-Job genutzt werden. Durch eine einfach zu erlernende Konfigurationsdatei sind Synchronisationsjobs schnell eingerichtet.

Die gleiche Konfiguration bei Clustern ist das A und O, daher haben wir unseren Kurs „Linux Hochverfügbarkeits-Cluster“ (siehe unten) um dieses praktische Werkzeug ergänzt.



Christian Fertsch  
([info@ordix.de](mailto:info@ordix.de)).

## Larry Ratlos

# Larry und das Problem mit der Namensliste

Larry bekam heute eine Mail mit dem Auftrag, aus einer XML-Nachricht eine Namensliste der Geschäftsstelle Paderborn zu erstellen. Diese Liste soll bis zum 16. Dezember dem Geschäftsführer vorliegen. Jetzt ist Stress vorprogrammiert oder können Sie weiterhelfen?

### Namen über Namen

Abteilungsleiter Bruus hatte ihm die Anforderung zusammen mit einer XML-Datei wortlos übermittelt.

Die Zeichenkette, die in Abbildung 1 gezeigt wird, hat er als einzigen Anhaltspunkt.

Er überlegt, in welcher Sprache werden Suchen in dieser Weise formuliert?

### Können Sie Larry helfen?

Schicken Sie Larry Ihre richtige Lösung, damit er die Namensliste erzeugen kann. Auf Ihren



Lösungsvorschlag freut sich Larry **bis zum 16. Dezember 2010 an [kniffel@ordix.de](mailto:kniffel@ordix.de)**.

Die Lösung des Rätsels aus der Ausgabe 3/2010 finden Sie unter:

[http://www.ordix.de/ORDIXNews/verzeichnis\\_42010.html](http://www.ordix.de/ORDIXNews/verzeichnis_42010.html)

```
„//Adresse[Ort='Paderborn']/../Nachname/text()“
```

**Abb. 1: Welche Sprache ist das?**

# Einfache Anbindung mit dem Nagios::Plugin-Modul

Der Artikel richtet sich an Administratoren, die mit geringem Aufwand Nagios-Plugins erstellen möchten.

Nagios gibt es mittlerweile als freie Monitoring-Lösung in Rechenzentren aller Größenklassen. Eine genial einfache Schnittstelle erlaubt es, mit wenigen Schritten eigene Plugins in beliebigen Programmiersprachen zu erstellen. Möchte man allerdings neben den Basisinformationen auch Performance-Daten (z. B. zur Auswertung mit PNP oder NagVis) liefern, eine Online-Hilfe anbieten oder gar das eigene Plugin der wachsenden Nagios Community zur Verfügung stellen, führt kein Weg um das Modul Nagios::Plugin herum. Wir zeigen Ihnen, wie Sie schnell zu einem überzeugenden Ergebnis kommen.

## Die Plugin-Schnittstelle

Die grundsätzlichen Anforderungen an ein Nagios-Plugin sind denkbar gering. Ein gültiges Skript muss lediglich einen entsprechenden Return Code (siehe Abbildung 1) zurückliefern, gefolgt von einer Zeile Text (ab Nagios 3 kann dieser auch mehrzeilig sein).

Nachteile dieser Anforderung:

- Keine Standard-Optionen (wie z. B. `--help`)
- Keine Online-Dokumentation,
- Keine Performance-Daten
- Keine Überprüfung der Eingaben.

Dies lässt sich natürlich auch alles mit Bordmitteln herstellen. Der Aufwand steht hier aber für den Einzelnen oft nicht im Verhältnis zum Ergebnis und wird deshalb meist gescheut.

Numerischer Wert	Status
0	OK
1	WARNING
2	CRITICAL
3	UNKNOWN
4	DEPENDENT

Abb. 1: Gültige Return-Codes für Nagios.

## Das Modul Nagios::Plugin

Das Modul `Nagios::Plugin` aus dem CPAN-Archiv wird von den Nagios-Entwicklern gepflegt. Leider ist die Dokumentation noch recht dürftig und selbstverständlich nur auf Englisch verfügbar. Die Installation erfolgt am besten über die CPAN-Shell. Einige Linux-Distributoren bieten allerdings auch fertige Installationspakete an.

### Plugin initialisieren

Nach dem Einbinden des Moduls folgt zunächst der Aufruf des Konstruktors `new()`. Abbildung 2 zeigt die beispielhafte Initialisierung (Zeilen 4-9). Wie bei Perl-Objekten üblich, werden die Argumente in einem Hash übergeben. Eine besondere Bedeutung kommt hier dem Argument `usage` zu. Wird dieses gesetzt, wird automatisch ein Objekt vom Typ `Nagios::Plugin::Getopt` erzeugt. Bereits vorbelegt und sofort verwendbar sind die Schalter `--help`, `--version`, `--usage`, `--timeout`, `--verbose` sowie die jeweiligen Kurzformen (`-h`, `-V`, `-u`, `-t`, `-v`).

### Eigene Optionen definieren

In den Zeilen 10-25 von Abbildung 2 werden drei zusätzliche Optionen eingeführt (jeweils

in Lang- und Kurzform) und dann in Zeile 26 zur Verfügung gestellt. Zu jeder Option kann ein Hilfe-Text mit angegeben werden, der bei der Verwendung der Option `--help` ausgegeben wird. Die Optionen `warning` und `critical` erwarten hier einen Integer-Wert (i), während die Variante `file` einen Dateinamen erwartet (s).

Bereits an dieser Stelle liefert das Skript beim Aufruf ohne Optionen eine Usage-Meldung, da alle drei Möglichkeiten mit dem Argument `required=>1` versehen wurden (siehe Abbildung 3).

### Automatische Online-Hilfe

Die Option `--help` liefert auch jetzt schon eine vollständige Hilfe (siehe Abbildung 4). Sie sehen, dass für alle Varianten automatisch entsprechende Hilfenmeldungen generiert werden. Eine Besonderheit stellt hier der Schalter `critical` dar. Hier wurde mit der Option `label` die automatische Einstellung `INTEGER` überschrieben. Bei `warning` und `file` wird die Art des Arguments aus der Zuweisung in der `spec`-Option interpoliert.

### Optionswerte übernehmen und Thresholds setzen

Die Inhalte der Optionsargumente bzw. der Status der Variante ohne Argumente stehen über die Methode `opts` zur Verfügung. Mit dem Befehl `set_thresholds()` können die Schwellenwerte für die Status `WARNING` und `CRITICAL` gesetzt werden. In Abbildung 5 sehen sie die Zuordnung der Optionen und das Setzen der Schwellenwerte in den Zeilen 28-33.

### Performance-Daten hinzufügen

Zur Überprüfung der Schwellenwerte muss das Beispielskript die Größe der Datei ermitteln. Es liegt also nahe, den jeweils ermittelten Wert an Nagios zu übermitteln, falls dort Performance-Daten ausgewertet werden sollen. Die Methode `add_perfdata()` in (Abbildung 5, Zeile 39-44) stellt dies automatisch zur Verfügung.

### Status ermitteln

Statt sich mit ewigen `If-then-else`-Konstrukten abzumühen, können Sie den Vergleich der ermittelten Werte mit den ein-

```
01 #!/usr/bin/perl -w
02 use strict;
03 use Nagios::Plugin;
04 my $np = Nagios::Plugin->new(
05     shortname => "TESTPLUGIN",
06     usage => "Usage: %s [-v|--verbose ] [-H <host>]
[-t <timeout> ] "
07     . " -c|--critical=<threshold> -w|--
warning=<threshold> "
08     . " [-h|--help] -f|--file=<filemonitor> "
09 );
10 $np->add_arg(
11     spec      => "warning|w=i",
12     help      => "Size of File in BYTES",
13     required  => 1
14 );
15 $np->add_arg(
16     spec      => "critical|c=i",
17     label     => ['BYTES'],
18     help      => "size of File in BYTES",
19     required  => 1
20 );
21 $np->add_arg(
22     spec      => "file|f=s",
23     help      => "File to monitor (full qualified)",
24     required  => 1
25 );
26 $np->getopts;
```

Abb. 2: Erste Schritte.

```
./checkfilesize
Usage: checkfilesize [-v|--verbose ] [-H <host>] [-t <timeout> ]
-c|--critical=<threshold> -w|--warning=<threshold> [-h|--help]
-f|--file=<filemonitor>

Missing argument: warning
Missing argument: critical
Missing argument: file
```

Abb. 3: Usage-Meldung bei fehlenden Optionen.

```
./checkfilesize --help
This nagios plugin is free software, and comes with ABSOLUTELY
NO WARRANTY.
It may be used, redistributed and/or modified under the terms of
the GNU
General Public Licence (see http://www.fsf.org/licensing/licen-
ses/gpl.txt).

Usage: checkfilesize [-v|--verbose ] [-H <host>] [-t <timeout> ]
-c|--critical=<threshold> -w|--warning=<threshold> [-h|--help]
-f|--file=<filemonitor>

-?, --usage
    Print usage information
-h, --help
    Print detailed help screen
-V, --version
    Print version information
--extra-opts=[section][@file]
    Read options from an ini file. See http://nagiosplugins.org/
extra-opts for usage
-w, --warning=INTEGER
    Size of File in BYTES
-c, --critical=BYTES
    size of File in BYTES
-f, --file=STRING
    File to monitor (full qualified)
-t, --timeout=INTEGER
    Seconds before plugin times out (default: 15)
-v, --verbose
    Show details for command-line debugging (can repeat up to 3 times)
```

Abb. 4: Online-Hilfe sofort verfügbar.

```

27
28 my $testfile = $np->opts->file;
29 my $warning = $np->opts->warning;
30 my $critical = $np->opts->critical;
31
32 $np->set_thresholds(warning => $warning,
33                   critical => $critical);
34
35 if (-r $testfile)
36 {
37     my @fileinfo = stat($testfile);
38     my $size = $fileinfo[7];
39     $np->add_perfdata(
40         label    => "size",
41         value    => $size,
42         uom      => "B",
43         threshold => $np->threshold()
44     );
45
46     my $result = $np->check_threshold($size);
47     $np->nagios_exit($result, "Groesse der Datei $testfile:
48 $size Bytes");
49 }
50 else { $np->nagios_die("File $testfile not found:$!") }

```

Abb. 5: Teil 2 des Beispielskripts.

```

./checkfilesize -w 100 -c 1000 -f /tmp/testdatei
TESTPLUGIN CRITICAL - Groesse der Datei /tmp/testdatei: 102400
Bytes | size=102400B;100;1000
./checkfilesize -w 100000 -c 1000000 -f /tmp/testdatei
TESTPLUGIN WARNING - Groesse der Datei /tmp/testdatei: 102400
Bytes | size=102400B;100000;1000000
./checkfilesize -w 120000 -c 1000000 -f /tmp/testdatei
TESTPLUGIN OK - Groesse der Datei /tmp/testdatei: 102400 Bytes
| size=102400B;120000;1000000
./checkfilesize -w 120000 -c 1000000 -f /tmp/gibtsnicht
TESTPLUGIN UNKNOWN - File /tmp/gibtsnicht not found:No such
file or directory

```

Abb. 6: Testen des Plugins auf der Kommandozeile.

## Links

- ▶ [1] ORDIX News Artikelreihe „DB2 und Perl“:  
<http://www.ordix.de/ORDIXNews/artikelreihen.html#DB2Perl>
- ▶ [2] Download und Dokumentation zu „Nagios::Plugin“:  
<http://search.cpan.org/~tonvoon/Nagios-Plugin-0.34/lib/Nagios/Plugin.pm>
- ▶ [3] Nagios plugin-developer Guidelines:  
<http://nagiosplug.sourceforge.net/developer-guidelines.html>
- ▶ Seminarempfehlung „Systemüberwachung mit Nagios Grundlagen“:  
<http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Systemmanagement/nr/648/index.html>

gestellten Schwellenwerten der Methode `check_threshold()` überlassen (siehe Abbildung 5, Zeile 49). Sie liefert als Ergebnis den Status der Überprüfung (siehe Abbildung 1).

## Skript beenden und Werte an Nagios zurückgeben

Das Modul `Nagios::Plugin` stellt hierfür zwei Möglichkeiten zur Verfügung. Die Methoden `nagios_exit()` und `nagios_die()`. Die Methode `nagios_exit()` verlangt immer sowohl den Status als auch den Nachrichtentext als Argumente, während der Status bei der Methode `nagios_die()` optional ist (Default ist UNKNOWN) (siehe Abbildung 5, Zeilen 47 und 49). Die Performance-Daten werden automatisch im richtigen Format angehängt.

Die Abbildung 6 zeigt den Test des vollständigen Skripts auf der Kommandozeile. Die Logik dieses Beispiels wurde natürlich betont schlicht gehalten, um die `Nagios::Plugin`-Funktionalität in den Vordergrund zu stellen.

Das Ergebnis kann sich für deutlich weniger als 50 Zeilen Quelltext sehen lassen. Dieser Artikel zeigt nur die wesentlichen Methoden und Optionen der `Nagios::Plugin`-Familie auf.

Weitere Einstellungen und Verwendungsmöglichkeiten finden Sie in der Online-Hilfe des Moduls.



Norbert Munkel  
([info@ordix.de](mailto:info@ordix.de))

## Zentrales Auditing mit Oracle Security

# Oracle Audit Vault – Das zentrale Data Warehouse für Audit-Daten

Viele unserer Kunden legen großen Wert auf das Thema „Zentrales Auditing“ und fragen nach einer entsprechenden Umsetzungsmöglichkeit. Aufgrund von Forderungen von Wirtschaftsprüfern oder Revisionen als Schutz vor internen Bedrohungen oder von Vorschriften wie SOX oder PCI müssen Informationen über das, was irgendwo von irgendjemandem in der Datenbank getan wurde, vorgehalten werden. Die hierbei entstehende Flut an sensiblen Audit-Daten soll zentral konsolidiert, geschützt und sinnvoll ausgewertet werden. Oracle trägt dieser Forderung mit seinem Mitte 2007 eingeführten Audit Vault Rechnung.

Dieser Artikel richtet sich sowohl an Datenbankadministratoren als auch an Sicherheitsbeauftragte im Unternehmen.

## Überblick

Mit Oracle Audit Vault bietet Oracle im Bereich Security ein Produkt an, mit dem Audit-Einträge aus unterschiedlichen Quellsystemen automatisiert gesammelt, in einer zentralen Oracle Datenbank (Audit Vault Server) – dem Data Warehouse für Audit-Daten – konsolidiert, geschützt und per „Knopfdruck“ ausgewertet werden können.

## Funktionsumfang und Nutzen

Unmittelbar nachdem die Audit-Daten auf den Quellsystemen anfallen, werden sie als Rohdaten von Audit Vault zeitnah erfasst und über einen nachgelagerten ETL-Prozess in das Open-Data Warehouse von Audit Vault geladen.

Im Lieferumfang von Audit Vault ist eine Web-Oberfläche (Audit Vault Console) enthalten, die eine größere Anzahl von vordefinierten Reports für Auswertungen bereit hält.

Von dieser Konsole aus lassen sich zudem die Audit-Einstellungen für alle beteiligten Datenbanken zentral steuern und ausrollen. Durch die Möglichkeit, Alerts inklusive einer optionalen E-Mail-Benachrichtigung zu definieren, können auffällige Aktivitäten schnell und automatisch gemeldet werden.

Um die Audit-Daten vor Manipulationen zu schützen, verwendet Oracle Audit Vault die Optionen Oracle Advanced Security und Oracle Database Vault. Hierdurch kann zum einen der Transfer der Audit-Daten verschlüsselt werden, zum anderen wird der lesende und schreibende Zugriff auf Audit-Daten durch Datenbankadministratoren und privilegierte Benutzer unterbunden.

## Architektur und Komponenten

Die Architektur von Oracle Audit Vault setzt sich aus folgenden Hauptkomponenten zusammen (siehe Abbildung 1):

- Audit Data Source(s)
- Audit Vault Collection Agent(s)
  - Kollektoren
- Audit Vault Server
  - Audit Data Store (= Secure Audit Data Warehouse)
  - Audit Vault Services
  - Audit Vault Console

## AV-Komponente: Audit Data Source

Unter einer Audit Data Source ist das laufende Quellsystem zu verstehen, aus dessen

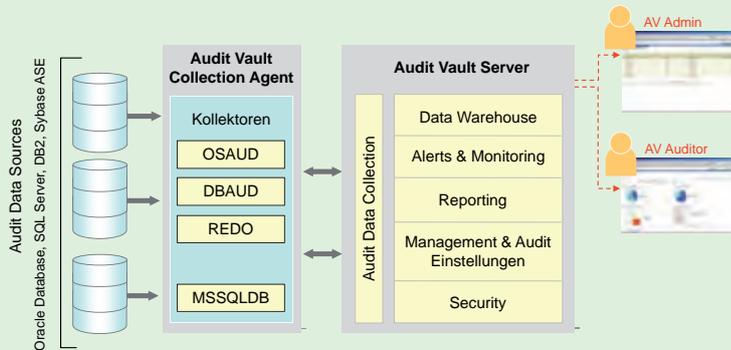


Abb. 1: Oracle Audit Vault Architektur.

Kollektor	Audit Source	Audit Pfad
OSAUD	Oracle Datenbank	Audit Dateien im OS Unix/Linux: *.aud, *.xml, syslog Windows: Event Log, *.xml
DBAUD	Oracle Datenbank	Audit-Tabellen in der Datenbank (SYS.AUD\$, SYS.FGA_LOG\$, DVSYS.AUDIT_TRAIL\$)
REDO	Oracle Datenbank	Redo Logs der Datenbank zur Nachverfolgung von DML-, DDL-Statements inkl. Wertveränderungen
MSSQLDB	MS SQL Server	C2 Audit Logs, Server-side Trace Logs, Windows Event Log
SYBDB	Sybase ASE	System Audit Tabellen (sysaudits_01 - sysaudits_08)
DB2DB	IBM DB2	Binary Audit Log (db2audit.log)

Abb. 2: Kollektoren für die Sammlung und Übermittlung von Audit-Daten.



Abb. 3: Auswahlsite der vordefinierten Reports in der AV Console.

Audit-Pfaden die Audit-Einträge extrahiert und gesammelt werden.

Audit Vault hält in seiner aktuellen Version 10.2.3.2 Mechanismen bereit, um Audit-Daten aus folgenden Datenbankprodukten zu erfassen:

- Oracle Datenbank 9.2, 10g, 11g auch mit Database Vault Option
- Microsoft SQL Server 2000, 2005 und 2008
- Sybase ASE, Versionen 12.5.4 – 15.0
- IBM DB2, Versionen 8.2 – 9.5

**AV-Komponente: Collection Agent**

In der Regel läuft auf den Quellsystemen, also auf den Systemen, deren Datenbanken auditiert werden sollen, jeweils ein Audit Vault Collection Agent. Dieser Collection Agent ist für die Steuerung der sogenannten Kollektoren verantwortlich. Über ihn findet zudem die Kommunikation mit dem Audit Vault Server statt.

Für den Zugriff der Kollektoren auf die Source-Datenbanken stellt der Agent die entsprechenden Zugangsdaten geschützt in einem „Secure External Password Store“ im AV Wallet bereit.

**AV-Komponente: Kollektoren**

Auf die verschiedenen Audit-Pfade (Audit Trails) der einzelnen Audit Source-Datenbanken kann jeweils ein Kollektor „gesetzt“ werden. Dieser sammelt die Audit-Daten, unmittelbar nachdem sie angefallen sind und übermittelt sie an den AV Server. Abbildung 2 zeigt die Kollektoren, die in der aktuellen Version zur Verfügung stehen.

**AV-Komponente: Audit Vault Server**

Der AV Server hält ein zentrales Audit Repository bereit, in dem die von den Kollektoren gesammelten Audit-Daten abgelegt und verwaltet werden. Hier werden Data Warehouse Strukturen genutzt, die das Erzeugen von Auswertungen vereinfachen und performant gestalten sollen. Da Oracle das Schemadesign seines Data Warehouses offen gelegt hat, ist der Zugriff durch weitere Reporting-Werkzeuge (Oracle Application Express, Business Intelligence Tools) möglich.



**Glossar**

<b>APEX</b>	Oracle Application Express ist eine vollständige, mit der Oracle-Datenbank integrierte Entwicklungs- und Laufzeitumgebung, mit der datenbankgestützte Web-Anwendungen erstellt werden können.
<b>DDL</b>	Data Definition Language. Über DDL-Kommandos werden Datenstrukturen gepflegt (z. B. Tabellen anlegen oder löschen).
<b>DML</b>	Data Modification Language. Über DML-Kommandos werden Daten (Zeilen) in Tabellen eingefügt, gelöscht oder geändert.
<b>ETL</b>	Extract, Transform, Load. ETL bezeichnet einen Prozess in einer Data Warehouse-Umgebung, um Daten von einem Quell- in ein Zielsystem zu kopieren.
<b>FGA</b>	Fine Grained Auditing. Fein granuliertes Auditing.
<b>PCI-DSS</b>	Payment Card Industry Data Security Standard. Regelwerk für Kreditkartendaten verarbeitende Unternehmen.
<b>SOX</b>	Sarbanes-Oxley Act ist ein US-Bundesgesetz, das sich auf die Verbesserung der Berichterstattung von Unternehmen am US-Kapitalmarkt konzentriert. Durch strenge Richtlinien wird gewährleistet, dass die von Unternehmen veröffentlichten Finanzdaten sicher und verlässlich sind.

**Links**

- ▶ [1] Buch „HOWTO Secure and Audit Oracle 10g and 11g“, Autor: Ron Ben Natan, ISBN-10: 1420084127, ISBN-13: 978-1420084122
- ▶ [2] Oracle Audit Vault Documentation: <http://www.oracle.com/technetwork/database/audit-vault/documentation/index.html>
- ▶ [3] Seminarempfehlung „Oracle Security“ <http://training.ordix.de/siteengine/action/load/kategorie/Datenbanken/nr/837/index.html>

detaillierten Informationen des ausgewählten Audit-Ergebnisses zu verzweigen (siehe Abbildung 4). Format und Ergebnismenge der Reports können zudem angepasst werden und als benutzerdefinierte Reports dauerhaft im Audit Vault hinterlegt werden.

Bisher konnten Reports immer nur „auf Knopfdruck“ erstellt werden. Erst mit der aktuellen AV-Version 10.2.3.2 ist es möglich, Ablaufpläne zur planmäßigen und regelmäßigen Ausführung von Berichten zu definieren. Die so generierten Berichte werden im PDF-Format in der Datenbank abgelegt und können sowohl über die Konsole abgerufen als auch an bestimmte Empfänger per Mail gesendet werden.

**Schutz der Audit-Daten**

Um die gesammelten und gespeicherten Daten zu schützen, verwendet Audit Vault zwei Optionen:

- **Oracle Advanced Security**  
Zum Verschlüsseln des Transfers der Audit-Daten zwischen Collection Agent und AV Server.
- **Oracle Database Vault**  
Zum Unterbinden des lesenden und schreibenden Zugriffs auf die Audit-Daten im AV Server durch Datenbankadministratoren und privilegierte Benutzer.

Dies wird erreicht durch eine strikte Aufgabentrennung („Separation of duties“) im Data Warehouse mittels spezieller Admin-Accounts, die während der Installation des AV Servers in der Repository-Datenbank angelegt werden. Diesen Accounts werden Audit- und Database Vault Rollen zugeordnet, die ganz bestimmte Tätigkeiten erlauben wie z. B. das Anlegen und Berechtigen von Benutzern, das Lesen von Daten zur Berichterstellung oder die Administration des Audit und Database Vault im Allgemeinen.

Um den „Audit-Tresor“ vor unautorisiertem Zutritt verschlossen zu halten, ist es wichtig diese Rollenverteilung verstanden zu haben und sie in der Praxis durch die Verteilung an verschiedene Personen umzusetzen.

**Fazit**

Gerade wer schon einmal ein zentrales Auditing „zu Fuß“ aufgesetzt hat, weiß welche zahlreichen und aufwändigen Implementierungsschritte eine solche Lösung erfordert. Auf umso größeres Interesse dürften also die Automatismen und Möglichkeiten stoßen, die uns Audit Vault in diesem Bereich bietet.

Auch für die Belange eines manipulationsgeschützten Data Warehouses, das Audit-Daten aus unterschiedlichen Datenbanksystemen zentral zusammenführen soll, ist das Audit Vault ein durchweg funktionsfähiges und empfehlenswertes Produkt.



*Kathleen Hock  
(info@ordix.de).*



Rückblick Treffpunkt RheinMain IT:

## Was haben Manager und Projektmanager gemeinsam?

Diese und andere Fragen rund um das Thema Projektmanagement beantwortete **Benedikt Georgi, Vorstand der ORDIX AG**, in seinem Vortrag am 09. November 2010 im ORDIX Trainingszentrum Wiesbaden.

### Projektmanagement ist Management

In seinem Vortrag forderte Benedikt Georgi zum Perspektivwechsel auf. Projektmanagement wird häufig unterbewertet und als Tätigkeit betrachtet, die man „nebenbei“ erledigen kann.

Dass Projektmanagement deutlich mehr Respekt verdient und es viele Analogien zu eigentlichen Management-Aufgaben gibt, bewies der erfahrene Manager bei der Gegenüberstellung der Aufgaben, der Rollen und der Skills, über die sowohl Manager als auch Projektmanager gleichermaßen verfügen müssen.

Zudem reicherte der langjährige Projektmanager und Coach den Vortrag mit lebensnahen Praxisbeispielen und Anekdoten aus seinen über 20 Jahren Projekt- und Management-erfahrung an.

Dieser Vortrag bildete den Abschluss der diesjährigen Veranstaltungsreihe. Die Termine und Themen für das Jahr 2011 stehen bereits fest. Informieren Sie sich jetzt und sichern Sie sich Ihre Teilnahme für das nächste Jahr.

Informationen, Termine, Themen unter: [www.RheinMainIT.de](http://www.RheinMainIT.de)

Treffpunkt  
RheinMain **IT**

Rückblick 55. IUG Workshop:

## Informix-Features im Data Warehouse

Die Informix-User-Group traf sich am 26. Oktober 2010 in Dortmund zu Ihrem 55. Workshop. Im Anschluss an die Begrüßung durch Wolfgang Kögler, Beiratsmitglied der IUG und Vorstandsvorsitzender der ORDIX AG, folgten acht interessante Vorträge. Wolfgang Kögler referierte in seinem Vortrag zu dem Thema „Legenden rund um das Data Warehouse“.

### Informix-Funktionen bringen Gewinn bei Data Warehouses

Wolfgang Kögler gab in seinem Vortrag einen Überblick über eine Reihe von gewinnbringenden Funktionen, die Informix bei Data Warehouses bietet.

Er erläuterte Informix-Funktionen wie zum Beispiel Partitionierung und High Performance Loader. Der Informix-Experte stellte in seinem Data Warehouse-Beispiel die Maßnahmen eines großen deutschen Versicherers vor. Durch den Einsatz verschiedener Metho-

den konnte das Unternehmen eine erheblich Performance-Verbesserung bei Abfragen, Lade- und Löschvorgängen erzielen.

### Wissensweitergabe als Philosophie

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

**IUG**

Dezember 2010 - Mai 2011

	Dauer	Preis	Dezember		Januar			Februar			März			April			Mai										
			KW 49	KW 50	KW 51	KW 52	KW 1	KW 2	KW 3	KW 4	KW 5	KW 6	KW 7	KW 8	KW 9	KW 10	KW 11	KW 12	KW 13	KW 14	KW 15	KW 16	KW 17	KW 18	KW 19	KW 20	KW 21
<b>Datenbanken</b>																											
Datenbank-Hochverfügbarkeitslösungen für Entscheider	1 Tag	550,00 €																									
Datenbank-Modellierung	2 Tage	890,00 €																									
Oracle SQL	5 Tage	1.890,00 €																									
Oracle SQL für Experten	3 Tage	1.290,00 €																									
Oracle Datenbankprogrammierung mit PL/SQL Grundlagen	5 Tage	1.890,00 €																									
Oracle Datenbankprogrammierung mit PL/SQL Aufbau	5 Tage	1.890,00 €																									
Oracle PL/SQL Tuning	3 Tage	1.890,00 €																									
Oracle APEX Programmierung	3 Tage	1.290,00 €																									
Objektorientierung in Oracle	5 Tage	1.990,00 €																									
Oracle Datenbankadministration Grundlagen	5 Tage	1.990,00 €																									
Oracle Datenbankadministration Aufbau	5 Tage	1.990,00 €																									
Manuelles Oracle Backup und Recovery & weitere Konzepte	5 Tage	1.990,00 €																									
Oracle Backup und Recovery mit RMAN	5 Tage	1.990,00 €																									
Oracle Tuning und Monitoring	5 Tage	1.990,00 €																									
Oracle AWR und ASH Analyse und Interpretation	3 Tage	1.290,00 €																									
Oracle Troubleshooting Workshop	5 Tage	1.990,00 €																									
Oracle 11gR2 RAC und Grid Infrastructure	5 Tage	1.990,00 €																									
Oracle 11g Neuheiten	5 Tage	1.990,00 €																									
Oracle Security	4 Tage	1.590,00 €																									
Oracle Data Guard	4 Tage	1.590,00 €																									
Oracle Grid Control	3 Tage	1.290,00 €																									
Oracle Streams Advanced Queuing	3 Tage	1.290,00 €																									
Oracle Migration und Patching	3 Tage	1.290,00 €																									
IBM Informix SQL	5 Tage	1.790,00 €																									
IBM Informix Dynamic Server Administration	5 Tage	1.990,00 €																									
IBM Informix Dynamic Server Tuning und Monitoring	5 Tage	1.990,00 €																									
IBM Informix Dynamic Server Backup und Recovery	3 Tage	1.290,00 €																									
IBM Informix Dynamic Server HV-Technologien unter Unix	4 Tage	1.590,00 €																									
IBM DB2 für Linux/Unix/Windows SQL Grundlagen	5 Tage	1.890,00 €																									
IBM DB2 für Linux/Unix/Windows Administration	5 Tage	1.990,00 €																									
IBM DB2 für Linux/Unix/Windows Monitoring und Tuning	3 Tage	1.290,00 €																									
MySQL Administration	3 Tage	1.190,00 €																									
Microsoft SQL Server Administration	5 Tage	1.790,00 €																									
<b>Programmierung</b>																											
PHP Programmierung Grundlagen	5 Tage	1.690,00 €																									
PHP Programmierung Aufbau	3 Tage	1.190,00 €																									
Perl Programmierung Grundlagen	5 Tage	1.690,00 €																									
Perl Programmierung Aufbau	5 Tage	1.690,00 €																									
Shell, Awk und Sed	5 Tage	1.690,00 €																									
AWK Intensiv-Workshop	3 Tage	1.190,00 €																									
<b>Web- und Applikations-Server</b>																											
Apache Web-Server Installation und Administration	3 Tage	1.190,00 €																									
Tomcat Konfiguration und Administration	3 Tage	1.190,00 €																									
WebSphere Application Server Installation/Administration	3 Tage	1.390,00 €																									
Administration und Konfiguration für JBoss	3 Tage	1.190,00 €																									
<b>Systemmanagement</b>																											
Systemüberwachung mit Nagios Grundlagen	3 Tage	1.190,00 €																									
Systemüberwachung mit Nagios Aufbau	2 Tage	890,00 €																									

Informationen und Anmeldung

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

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

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

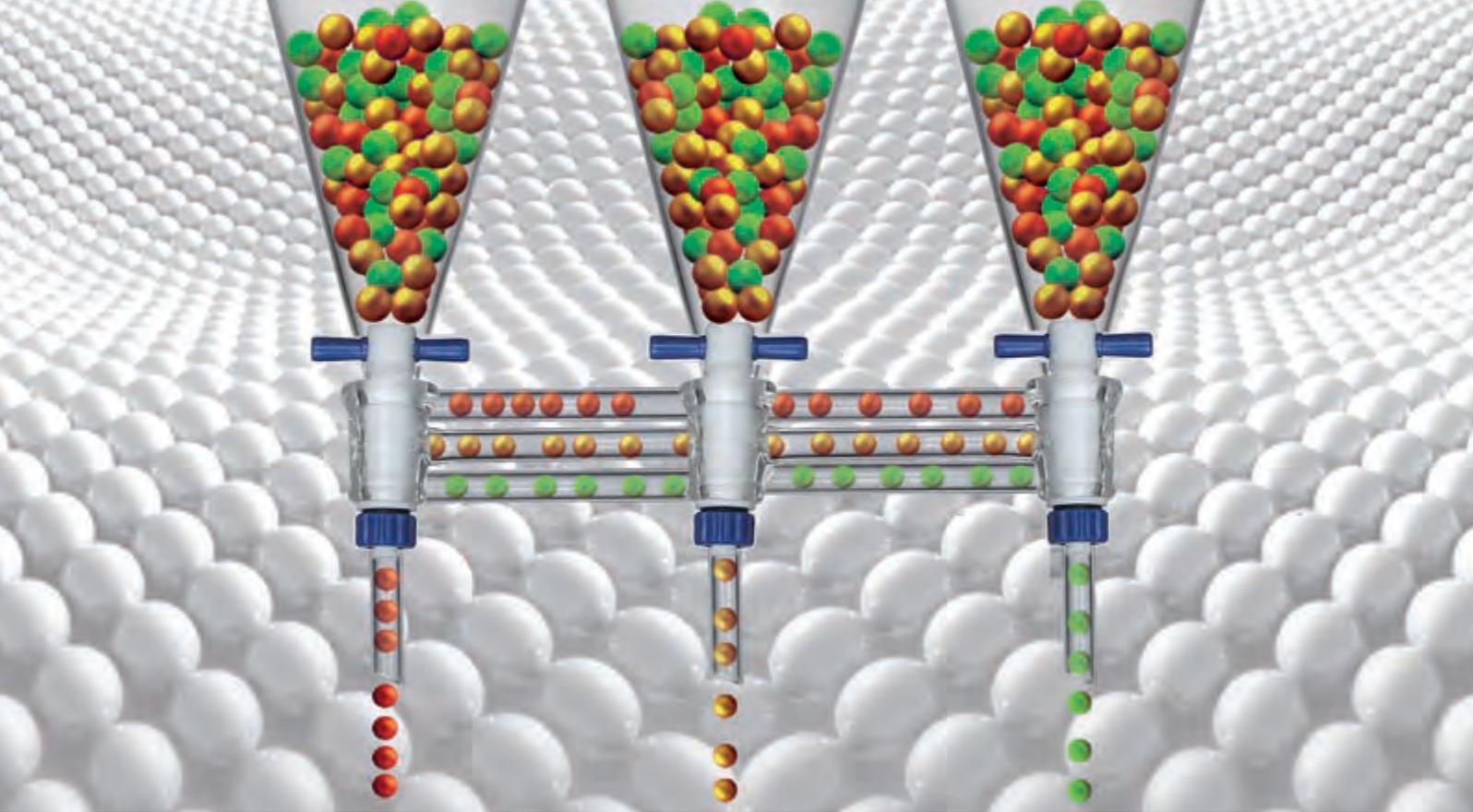
Online-Anmeldung und stets aktuelle Seminarinhalte und Termine unter: <http://training.ordix.de>

	Dauer	Preis	Dezember		Januar				Februar				März			April			Mai								
			KW 49	KW 50	KW 51	KW 52	KW 1	KW 2	KW 3	KW 4	KW 5	KW 6	KW 7	KW 8	KW 9	KW 10	KW 11	KW 12	KW 13	KW 14	KW 15	KW 16	KW 17	KW 18	KW 19	KW 20	KW 21
<b>Java-JEE</b>																											
Einführung in die objektorientierte Programmierung	3 Tage	1.190,00 €																									
Java Programmierung Grundlagen	5 Tage	1.690,00 €																									
Java Programmierung Aufbau	5 Tage	1.690,00 €																									
Java GUI Entwicklung mit Swing	5 Tage	1.690,00 €																									
JEE für Entscheider	1 Tag	590,00 €																									
Einführung in JEE	3 Tage	1.290,00 €																									
JSP und Servlet Programmierung	5 Tage	1.590,00 €																									
EJB Programmierung	5 Tage	1.590,00 €																									
Web-Anwendungen mit JavaServer Faces (JSF)	5 Tage	1.590,00 €																									
Entwickeln mit dem Spring-Framework	3 Tage	1.190,00 €																									
Java Web Services	3 Tage	1.190,00 €																									
Hibernate und die Java Persistence API	5 Tage	1.690,00 €																									
Java Performance Tuning	3 Tage	1.290,00 €																									
Anwendungsentwicklung für Android mit Java	5 Tage	1.690,00 €																									
Entw. v. Rich-Internet-Applikationen mit Google Web Toolkit	5 Tage	1.690,00 €																									
<b>Betriebssysteme</b>																											
Unix/Linux Grundlagen für Einsteiger	5 Tage	1.690,00 €																									
Linux Systemadministration	5 Tage	1.690,00 €																									
Linux Hochverfügbarkeits-Cluster	3 Tage	1.290,00 €																									
OpenLDAP - Praxiseinsatz im Netzwerk	4 Tage	1.490,00 €																									
Solaris Systemadministration Grundlagen	5 Tage	1.990,00 €																									
Solaris Systemadministration Aufbau	5 Tage	1.990,00 €																									
Solaris kompakt für erfahrene Unix-Administratoren	5 Tage	1.990,00 €																									
Solaris 10 für erfahrene Systemadministratoren	5 Tage	1.990,00 €																									
Solaris Containers	3 Tage	1.290,00 €																									
Solaris Virtualisierung mit LDOM	3 Tage	1.290,00 €																									
Zettabyte Filesystem (ZFS) Workshop	2 Tage	890,00 €																									
IBM AIX Systemadministration Grundlagen	5 Tage	1.990,00 €																									
IBM AIX Installation, Backup und Recovery	3 Tage	1.290,00 €																									
Analyse v. komplexen AIX Performance Problemen	5 Tage	2.300,00 €																									
IPv6 für Entscheider	1 Tag	590,00 €																									
IPv6 für Systemadministratoren	3 Tage	1.190,00 €																									
<b>Projektmanagement</b>																											
IT-Projektmanagement	5 Tage	1.990,00 €																									
Systemisches Projektmanagement	4 Tage	1.850,00 €																									
Agiles Projektmanagement mit SCRUM	2 Tage	1.100,00 €																									
Capability Maturity Model Integration (CMMI)	2 Tage	1.100,00 €																									
Kennzahlen der IT	2 Tage	1.100,00 €																									
Krisenmanagement in Projekten	2 Tage	1.100,00 €																									
IT-Projektcontrolling	3 Tage	1.290,00 €																									
Konfliktmanagement	2 Tage	1.100,00 €																									
<b>IT-Management</b>																											
IT-Architekturen	3 Tage	1.650,00 €																									
E-Business	2 Tage	1.100,00 €																									
IT-Strategien	2 Tage	1.100,00 €																									
Systemische Strategieentwicklung	3 Tage	1.650,00 €																									
IT-Management	5 Tage	1.990,00 €																									
IT-Risikomanagement	3 Tage	1.650,00 €																									
IT-Prozessmanagement	3 Tage	1.650,00 €																									

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

 Seminar in Wiesbaden

 Seminar in Bielefeld (IPS)



Neue Reihe: ETL im Data Warehouse am Beispiel IBM DataStage (Teil I)

# Geordneter Einzug dank ETL-Prozessen

Dieser Artikel richtet sich an Data Warehouse Entwickler und Architekten.

Im Data Warehouse Umfeld wird meist mit ETL-Prozessen gearbeitet. Dies ist eine besondere Art, die Datenversorgung einer Datenbank zu bewerkstelligen, für die verschiedene Hersteller unterschiedliche Ansätze zur visuellen Unterstützung anbieten. In unserer Artikelreihe „ETL im Data Warehouse“ wollen wir dieses Vorgehen anhand des IBM Produktes DataStage erläutern. Beginnend mit der allgemeinen Definition von ETL-Prozessen werden wir unsere Artikelreihe mit der Architektur der Software IBM DataStage sowie mit Technologien und Beispielen fortsetzen. Abschließend werden wir uns in Performance-Vergleichen speziell mit weiteren Technologien und Software-Lösungen befassen.

## Allgemeine Erklärung ETL

Die Abkürzung ETL setzt sich zusammen aus dem jeweils ersten Buchstaben, der drei Komponenten eines Datenbeschaffungsprozesses:

- Extract** Relevante Daten aus verschiedenen Quellen extrahieren.
- Transform** Daten verarbeiten, Anpassung an das Zielsystem (Datenbank).
- Load** Einspielung der Daten in das Zielsystem (Datenbank).

Ein ETL-Prozess kommt immer dann zum Einsatz, wenn Daten aus einem oder mehreren

operativen Systemen konsolidiert und in ein Data Warehouse eingespielt werden müssen. Damit soll einer redundanten Datenhaltung beim Import der Daten vorgebeugt und die Verwendung der Daten zu beispielsweise Auswertungszwecken vereinfacht werden.

Vorwiegend wird ein ETL-Prozess zu einem lastenarmen Zeitpunkt angestoßen oder sobald die Daten im Quellsystem konsistent vorliegen (zum Beispiel nach einem Tagesabschluss).

Der Prozess muss aber innerhalb eines bestimmten Zeitfensters beendet sein. Diese Anforderung stellt eine große Herausforderung an die Effizienz und Ablaufsteuerung des Prozesses dar. Bei dieser Aufgabenstellung

kommen wir meistens nicht ohne ein ETL-Werkzeug aus.

### Repository

Ein ETL-Werkzeug hält in der Regel in einem Repository die Transformations- und Datenbereinigungsregeln so wie Schemadaten der Zieldatenbank als Metadaten. Im gesamten ETL-Prozess tragen Metadaten zur Transparenz und Effizienzsteigerung bei. Entwickler können ohne großen Aufwand technische Metadaten nutzen, um Definitionen für Datenbanktabellen oder Datentypen zu hinterlegen.

### Sprachen und Systeme

Grundsätzlich kann der ETL-Prozess mit selbst-erstellten Programmen in allen gängigen Programmiersprachen umgesetzt werden. Beispielsweise mit dem Einsatz von SQL und PL/SQL in einer Oracle Datenbankumgebung. Es wird jedoch immer mehr auf bestehende ETL-Werkzeuge anderer Hersteller zurückgegriffen. Diese bringen viele Funktionalitäten von Haus aus mit, die mit einer Individualentwicklung erst programmiert werden müssten. Hierzu zählen Funktionalitäten wie Jobsteuerung, Fehlerbehandlung, Parametrisierung von Umgebungsvariablen oder Protokollierung.

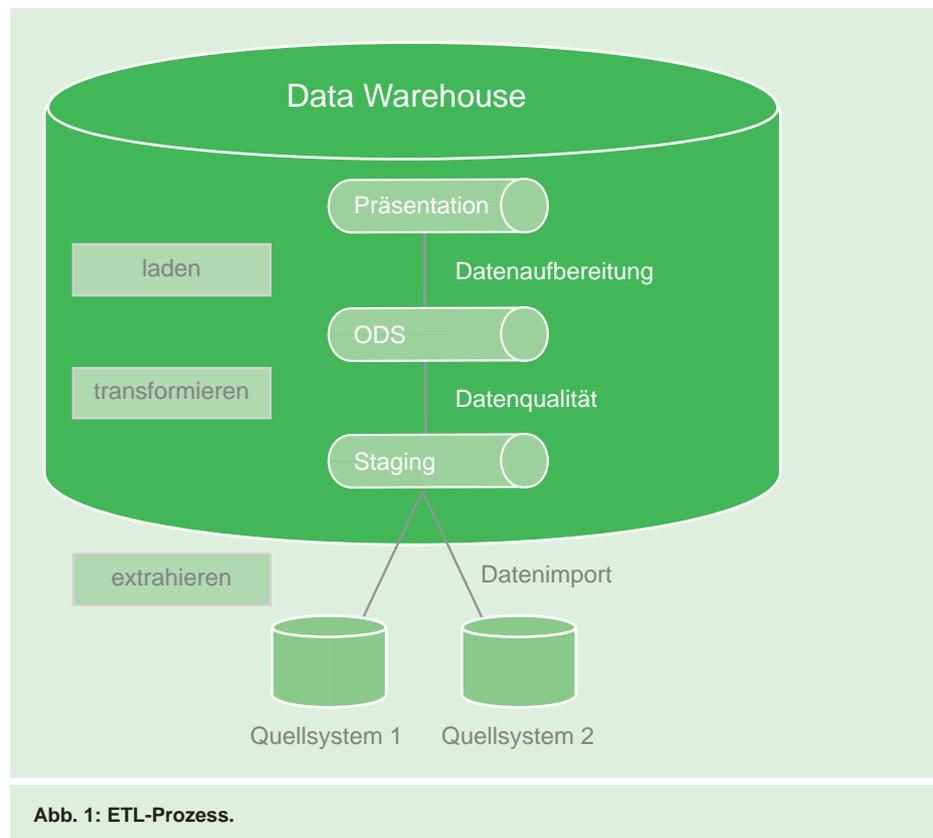
Zu den führenden Anbietern von Kommerziellen ETL-Systemen zählen:

- IBM (Information Server)
- Oracle (Oracle Warehouse Builder, Oracle Data Integrator)
- Informatica (PowerCenter)
- SAP-Business Objects (Business Objects Data Integrator)
- Microsoft (SQL Server Integration Services)

### Ablaufplan von ETL-Prozessen

ETL-Werkzeuge arbeiten sehr unterschiedlich. Bei einigen Produkten werden die Daten zuerst aus dem Quellsystem extrahiert, außerhalb verarbeitet und dann in das Zielsystem eingespielt. Andere Programme arbeiten ausschließlich in der Datenbank, dazu gehört beispielsweise der Oracle Warehouse Builder (OWB).

Im Regelfall läuft ein ETL-Prozess wie in Abbildung 1 ab. Zunächst wird ein Ausschnitt der Daten aus verschiedenen Quellen



extrahiert und der Staging Area bereitgestellt. Dort werden die Daten transformiert und für die weitere Verarbeitung im Operational Data Store (ODS) aufbereitet. Zusätzlich kann hier eine Historisierung angefertigt werden, um auf frühere Versionsstände zugreifen zu können.

Die Präsentationsschicht greift wiederum auf die aufbereiteten Daten zu und stellt diese dar. Als nächstes schauen wir uns die drei Stufen eines ETL-Prozesses (siehe Abbildung 1) genauer an.

### Staging

Ziel dieser meist temporären Stufe ist es, die Quelldaten so vollständig wie möglich in ein Data Warehouse zu laden. Es sollten also keine Veränderungen, gegenüber dem Quellsystem vorgenommen werden (Joins, Mapping, Datentypkonvertierung, o.ä.).

Die Daten können beispielsweise in zeichenaufnehmende Felder geladen werden, um Fehlern bei der Datentypkonvertierung vorzubeugen.

## Glossar

<b>DWH</b>	Data-Warehouse – Datensammlung mit unterschiedlichem Inhalt aus verschiedenen Systemen.
<b>Join</b>	Zusammenführen von Informationen die aufgrund von Normalisierung in unterschiedlichen Tabellen einer Datenbank liegen
<b>Mapping</b>	Zuordnung von Werten zwischen Quell- und Zielsystem.
<b>ODS</b>	Operational Data Store - Bereitstellung der Daten zur zeitnahen Auswertung.

## Historisierung (ODS)

Hier werden die geladenen Daten auf ihre Vollständigkeit, Qualität und Plausibilität geprüft. Auch die Formatierung von Werten sowie die Anreicherung der Datensätze mit weiteren Informationen, die womöglich aus einem anderen Quellsystem kommen, wird hier durchgeführt. Die aufbereiteten Datensätze werden als neue Datensätze angelegt und die Gültigkeit der bereits bestehenden Datensätze zum jetzigen Zeitpunkt abgeschlossen. Erst dadurch wird die Historisierung erreicht, die es ermöglicht, Trends zu erkennen und darauf angemessen zu reagieren.



Andreas Baier  
([info@ordix.de](mailto:info@ordix.de)).

## Präsentations- oder Analyseschicht

Auf dem Weg zu dieser letzten Schicht werden Berechnungen durchgeführt, um dem Nutzer

die gewünschten Kennzahlen zur Verfügung zu stellen. Da die extrahierenden Select Statements hier auf sehr große Datenmengen stoßen, besteht die größte Schwierigkeit darin, zum einen die Statements so effizient wie möglich zu gestalten und zum anderen das Datenmodell entsprechend zu optimieren.

## Fazit

Festzuhalten bleibt, dass ETL-Programme eine effiziente und performante Art der Datenbankbefüllung insbesondere für den Einsatz im Data Warehouse sind. Sie sind ausgelegt, um große Datenmengen möglichst schnell, regelmäßig und mit gleichen logischen Transformationen zu verarbeiten. Angesichts der hohen Beschaffungskosten und des großen Aufwandes für die Einarbeitung in ETL-Produkte, kann die Eigenentwicklung kurzfristig eine günstigere Lösung darstellen. Auf längere Sicht wird jedoch der Mehraufwand durch eine schnellere Umsetzung und bessere Wartung kompensiert.

Auch das Metadatenmanagement gewinnt immer mehr an Bedeutung je komplexer die Verarbeitung ist. Informationen über die Quell- und Zielsysteme, Transformationsregeln und Dokumentationen können automatisch erstellt und verwaltet werden.

Im nächsten Artikel stellen wir Ihnen die grundlegende Architektur von IBM DataStage vor.

- Anzeige -

## Geben Sie Ihrer Karriere einen neuen Kick!

Werden Sie Teil eines TEAMS von IT-Spezialisten als

- Oracle Senior Consultant (m/w)
- Java Senior Software Architekt (m/w)

Starten einer Java-Anwendung als natives Programm

# Integration mit Launch4j – der letzte Schliff

Anwender lieben das, was sie kennen. Also müssen sich Java-Programme nahtlos in die MS Windows Welt einfügen. Das Open Source Werkzeug Launch4j hilft dabei und lässt Java-Anwendungen genauso starten wie native Programme oder wandelt sie in ausführbare Windows-Dateien um. Obwohl Java mit Java Web Start selbst ein mächtiges Konzept zum Starten und Verteilen von Java Software zur Verfügung stellt, wird dabei die Plattformunabhängigkeit angestrebt. Launch4j dagegen ist der Spezialist in Sachen Integration.

Dieser Artikel richtet sich an Java-Entwickler und Systemintegratoren, die Java Software für MS Windows entwickeln möchten.

## Überblick

Unter Windows-Systemen kommt es oft zu Schwierigkeiten, wenn Anwender versuchen, Java-Anwendungen auszuführen. Abhilfe schafft hier das Open Source Werkzeug Launch4j. Nach der Installation erstellt dieses Werkzeug ein Startprogramm für Java-Anwendungen. Dieses hat die typischen Eigenschaften einer nativen Anwendung. Es lässt sich als `exe`-Datei mit Doppelklick starten und wird im Explorer mit eigenem Icon dargestellt. Die Versionseigenschaften eines nativen Programms, wie Dateiversion, Firma, Produktname und `-version` werden im Windows-Explorer angezeigt.

Die durch Launch4j erzeugte `exe`-Datei zeigt bei der Ausführung einen Startbildschirm, die installierte Java-Version wird überprüft. Alternativ kann eine spezielle Java-Installation für den Start genutzt werden. Das Startprogramm kann die `jar`-Datei beinhalten. Der Nutzer sieht nur das ausführbare Programm.

Launch4j ist über die Kommandozeilen-Schnittstelle oder über `ant`-Tasks in den automatischen Build-Prozess integrierbar. Für die Entwicklung der Konfiguration steht eine grafische Benutzeroberfläche zur Verfügung.

## Los geht's: Download und Installation

Launch4j ist über die Launch4j-Webseite [1] kostenlos verfügbar. Nach dem Download wird die `launch4j.exe` gestartet (siehe Abbildung 1).

Zunächst ist der Name des Programms im Reiter "Basic" als Output File einzutragen. Anschließend wird die `jar`-Datei angegeben, die gestartet werden soll. Standardmäßig wird die `jar`-Datei in das Startprogramm eingebettet.

Als letzte erforderliche Information wird im Reiter "JRE" die Java-Version definiert, die für das Programm benötigt wird. Unter `min JRE version` ist die Java-Version einzutragen, die durch `java -version` ausgegeben wird.

Danach wird die Konfiguration abgespeichert. Anschließend muss der Anwender nur noch auf das Zahnrad klicken, um den Build-Prozess zu starten. Schon ist das Startprogramm fertig.

## Neue Icons - darf es etwas bunter sein?

Das so erstellte Startprogramm wird im Windows-Explorer mit dem Standard-Icon für ausführbare Programme dargestellt (siehe Abbildung 2).

Im Reiter "Basic" ist das Feld Icon für die Angabe eines Icons vorgesehen. Es ist sinnvoll, das Startprogramm mit einem Icon zu dekorieren – so kann der Benutzer das Programm leichter identifizieren. In der Regel wird das Icon des Anwendungsfensters verwendet.

Bei Java-Programmen kann je nach Hardware einige Zeit vergehen, bis sich die Anwendung bemerkbar macht. Um dem Anwender unmittelbar zu zeigen, dass das Programm geladen

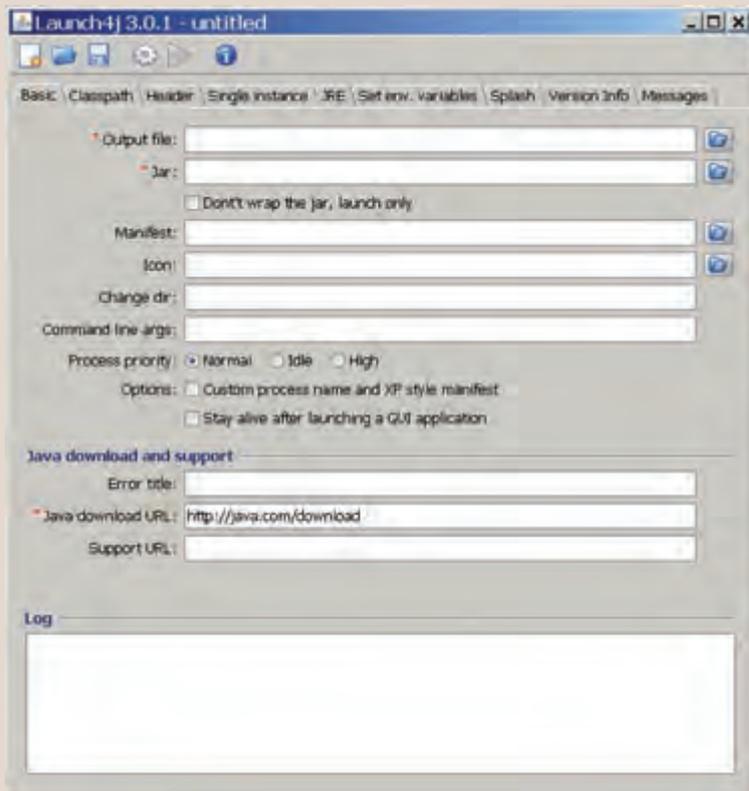


Abb. 1: Benutzeroberfläche von Launch4j.

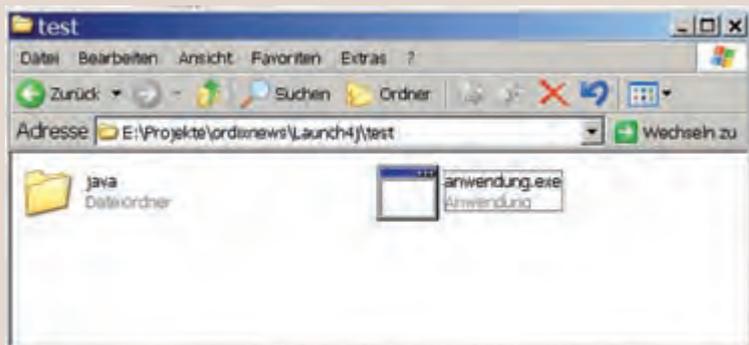


Abb. 2: Explorer-Darstellung einer Anwendung ohne Icon.

wird, ist ein Bild für den Startbildschirm - auch Splash genannt - zu definieren. Das Startprogramm zeigt den Startbildschirm, noch bevor das Java-Programm gestartet wird.

Dazu bietet Launch4j die Checkbox **Enable splash screen** im Reiter "Splash". Dort gibt

der Entwickler eine Bitmap-Grafik als Splash File an.

Nach dem Erstellen des Startprogramms wird das definierte Icon im Windows-Explorer angezeigt (siehe Abbildung 3).

### Java macht abhängig

Möglicherweise hat der Anwender nicht die richtige Java-Version installiert. In diesem Fall zeigt das Startprogramm die Nachricht, die im Reiter "Messages" im Feld **JRE version error** angegeben ist. Darüber hinaus öffnet sich der HTML-Browser mit der Seite, die im Feld Java download URL des Reiters "Basic" eingegeben ist.

Gerade bei Java-Programmen mit Benutzeroberflächen liegt die Überlegung nahe, die Java-Virtual Machine mit der Software auszuliefern, um Inkompatibilitäten zu neueren Java-Versionen auszuschließen. Damit diese Java-Version verwendet wird, muss im Reiter "JRE" das Feld **Bundled JRE path** mit dem Java-Installationspfad angegeben werden.

### Weitere Startparameter

Typischerweise wird die Anwendung Abhängigkeiten zu weiteren Bibliotheken haben. Bei einer überschaubaren Anzahl können die Bibliotheken zusammen mit dem Programm in eine **jar**-Datei gepackt werden. Damit lassen sich externe Abhängigkeiten vermeiden. Alternativ werden die Abhängigkeiten zu externen Bibliotheken im Classpath angegeben. Launch4j spendiert dieser Einstellung eigens einen Reiter namens "Classpath".

Konfigurationsparameter für die Virtuelle Maschine von Java, wie beispielsweise die Größe des Speicherraums, werden im Reiter "JRE" eingetragen. Bemerkenswert ist, dass der Speicherbedarf auch prozentual zum freien Speicherbereich angegeben werden kann.

Weitere Startparameter für das Programm werden im Feld **Command line args** des Reiters "Basic" definiert. Umgebungsvariablen werden im Reiter "Set env. variables" festgelegt.

### Programmeigenschaften

Im Windows-Explorer lassen sich Informationen zu Programmen unter Eigenschaften, Reiter "Version" anzeigen. Die wesentlichen

Attribute sind Beschreibung, Copyright und Dateiversion. Darüber hinaus können unter anderem Angaben zu Firma, Produktname, Produktversion und Sprache angezeigt werden. In Launch4j lassen sich diese Eigenschaften im Reiter "Version Info" definieren.

Launch4j kann auch verhindern, dass eine zweite Instanz des Programms gestartet wird. Im Reiter "Single Instance" muss dazu ein so genannter **Mutex name** angegeben werden.

### Fazit

Launch4j ist bestens geeignet, damit der Anwender ein Java-Programm genauso wie eine native Anwendung nutzen kann. Der Umfang ist gut durchdacht, die Benutzerschnittstelle ist intuitiv bedienbar, ein automatischer Build-Prozess wird unterstützt. Sind Aspekte der Verteilung der Software und die Versionspflege zu berücksichtigen, ist der Einsatz von Java Web Start in Erwägung zu ziehen.

Ob Launch4j, Java Web Start oder andere Werkzeuge: Wir beraten Sie gern beim Verteilen Ihrer Software.



Dr. Stefan Koch  
([info@ordix.de](mailto:info@ordix.de)).

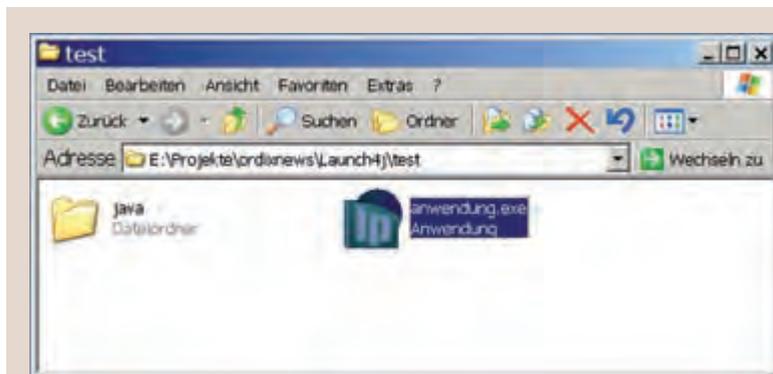


Abb. 3: Darstellung des gewählten Icons im Explorer.

### Links

- ▶ [1] Internet-Seite von Launch4j:  
<http://launch4j.sourceforge.net>
- ▶ [2] Dokumentation zu Java Web Start:  
<http://www.oracle.com/technetwork/java/javase/overview-137531.html>

### Glossar

**Splash-Screen** Start-Bildschirm. Als Splash Screen bezeichnet man einen grafischen Platzhalter, der während des Ladens oder Startens eines Computerprogramms erscheint. Splash Screens füllen meist nicht den gesamten Bildschirm aus, sondern werden häufig in einem rechteckigen Bereich in der Bildschirmmitte angezeigt.

## Seminarempfehlung: Java GUI Entwicklung mit Swing

▶ **Informationen/Online-Anmeldung:** <http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Java-J2EE/nr/79/index.html>

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

### Seminarinhalte

- Grafikbibliothek Swing/Unterschiede zu AWT
- Swing-Container und -Komponenten
- Einfache und komplexe Layout-Manager
- Event-Handling: Interfaces und Adapterklassen, Model-View-Controller (MVC), Umsetzung (Observer/Observable)
- Komplexe Swing-Container und -Komponenten
- Nutzung einer Tabelle in Java/Internationalisierung
- Übungen mit Hilfe einer durchgehenden Praxisaufgabe

### Termine

21.02. - 25.02.2011 in Wiesbaden  
09.05. - 13.05.2011 in Wiesbaden  
12.09. - 16.09.2011 in Wiesbaden  
05.12. - 09.12.2011 in Wiesbaden

**Seminar-ID:** P-JAVA-02

**Dauer:** 5 Tage

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

**Frühbuchepreis:** 1.521,00 € (zzgl. MwSt.)

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



Rückblick DOAG Konferenz und Schulungstag 2010

## Neue Gesichter im ORDIX Team auf der Konferenz



„Sie habe ich hier auf der Konferenz aber noch nie gesehen. Sind Sie neu bei ORDIX?“ So oder so ähnlich ließen sich wohl die Fragezeichen im Gesicht einiger Konferenzbesucher interpretieren. Dieses Jahr hatte ORDIX nämlich nicht nur seine Oracle Spezialisten im Gepäck, sondern wartete zusätzlich neben dem Partner coniatos auch mit einem großen Team von Java-Referenten auf. Neue Themen, frischer Wind - auch hier konnte ORDIX auf ganzer Linie überzeugen.

### Die Konferenz – ein Erfolgsgarant

Seit 23 Jahren ist die Deutsche Oracle-Anwenderkonferenz die jährliche Plattform zum Erfahrungsaustausch der Oracle-Community in Europa.

Die DOAG Konferenz und Ausstellung ist ein fester Termin im Kalender vieler Oracle Administratoren, Entwickler, Projektleiter und Manager. Besonders durch die Übernahme von Sun konnten die Organisatoren das Angebot an Vorträgen in diesem Jahr um zwei komplette Streams erweitern.

Gleichzeitig bildet der jährliche Termin den Treffpunkt für alle Usergroups aus dem Oracle-Umfeld. Vertreten waren in diesem Jahr nicht nur nationale Usergroups aus dem näheren Oracle Umfeld, sondern auch Gruppen aus den Bereichen Sun, Java und Solaris.

Mit insgesamt rund 2.000 Teilnehmern konnte die DOAG 2010 das hohe Niveau des Vorjahres halten.

### Über 400 Vorträge bot die Konferenz

Die DOAG Konferenz bot in diesem Jahr mit über 400 Vorträgen und Keynotes ein vielfäl-

tiges Spektrum an Informationen. Die renommierten Referenten deckten an den drei Konferenztagen alle Produktbereiche von Oracle mit entsprechenden Themen ab.

Das umfangreichste Vortragsprogramm in der Geschichte der DOAG war hauptsächlich auf die Sun-Übernahme zurückzuführen, wodurch komplett neue Streams zu Java und MySQL entstanden sind. Aber auch die traditionellen DOAG-Themen wie Datenbank, Entwicklung und Data Warehouse/BI sind weiter gewachsen.

Durch die Einrichtung neuer Vortragsräume boten sich den Besuchern in diesem Jahr erstmals 20 parallele Sessions. Um auch Anwendern aus anderen Ländern die Teilnahme zu ermöglichen, wurden viele Vorträge in englischer Sprache gehalten sowie eine Simultanübersetzung ausgewählter deutschsprachiger Vorträge angeboten.

Die ORDIX AG und ihre Tochtergesellschaft coniatos präsentierten sich an den drei Konferenztagen mit insgesamt elf Vorträgen. Neben seinem umfangreichen Oracle Know-how konnte ORDIX diesmal seine langjährigen Java- und MySQL-Erfahrungen unter Beweis stellen.

### Die Vorträge der ORDIX Referenten

Bereits am ersten Konferenztag präsentierte sich ORDIX mit drei Vorträgen in den Streams Oracle Datenbank, MySQL und Java. Besonders die Java- und MySQL-Referenten konnten bei ihrem Debüt punkten. In ihren Vorträgen überzeugten die Newcomer mit umfangreichen Praxiserfahrungen und vielen Tipps zu ihren Spezialthemen. Der zweite Konferenztag bildete mit vier eigenen Vorträgen und einem Vortrag der coniatos AG den Höhepunkt der diesjährigen DOAG Konferenz.

Den Abschluss bildeten am Donnerstag zwei weitere ORDIX-Vorträge und ein Vortrag der coniatos AG.

### Schulungstag 2010

Der Schulungstag wurde in diesem Jahr von Klaus Reimers gehalten und hatte das Thema „AWR und ASH Analyse“.

Über 40 Teilnehmer besuchten die Schulung und wurden mit den einzelnen Analyse mög-

lichkeiten vertraut gemacht. „Wie von ORDIX gewohnt, war dies wieder eine sehr gut gehaltene Schulung!“ so ein zufriedener Schulungsteilnehmer.

### ORDIX Messestand mit neuem Konzept

Neben 60 weiteren Firmen aus dem deutschsprachigen und internationalen Raum präsentierte sich die ORDIX AG auf der DOAG Ausstellung mit einem neu konzipierten Standkonzept. Die Präsentation der Dienstleistungen und des Schulungsangebotes standen hierbei im Vordergrund.

### Fazit – wir kommen wieder!

Das Resümee dieser Konferenz könnte positiver nicht ausfallen. Interessante Gespräche, sehr gut besuchte Vorträge und ein gelungener Schulungstag. Diese Faktoren führen uns zu dem Schluss:

Die ORDIX AG wird auch im nächsten Jahr wieder an der Konferenz teilnehmen.



### Links

- ▶ [1] Rückblick DOAG Konferenz 2009  
[http://www.ordix.de/ORDIXNews/3\\_2009/Aktuell/doag\\_konferenz.html](http://www.ordix.de/ORDIXNews/3_2009/Aktuell/doag_konferenz.html)
- ▶ [2] DOAG Webseite  
<http://doag.org>
- ▶ [3] coniatos Webseite  
<http://www.coniatos.de>

### Sie konnten leider nicht dabei sein?

Wir stellen Ihnen gerne die Vorträge auf CD zur Verfügung. Eine E-Mail an [info@ordix.de](mailto:info@ordix.de) genügt

### Die Vorträge der DOAG im Überblick

- Hochverfügbarkeit mit Data Guard
- Partitioning – Technik und Anwendungsbeispiele
- PL/SQL-Tuning
- Datums- und Zeitverwaltung in Oracle
- Endlich dabei – die Criteria-QueryAPI in der JPA 2.0
- Kontext und Dependency Management in Java EE - Einführung in JSR-299
- RichClient- Funktionalitäten per Google Web Toolkit
- Versionierung von Webservice-Schnittstellen
- MySQL Backup- und Recovery-Strategien
- Datenbank- und Betriebsoptimierung dank Agilem Projektmanagement
- Strategic Fit - Strategischer Wertbeitrag von Enterprise Architectures



Java Best Practice (Teil VIII)

# Concurrency - Lästige Konkurrenz (I)

Dieser Artikel richtet sich an Entwickler und Software-Architekten.

Ob bei der Server-Entwicklung mit EJB oder der Implementierung einer Swing-Anwendung: Parallele Ausführungsprozesse sind allgegenwärtig. Programmierrichtlinien sorgen dafür, dass der Entwickler sich um das Thema Multi-Threading keine Gedanken machen muss. Dennoch lohnt ein Blick hinter die Kulissen. Einerseits werden Programmierrichtlinien verständlicher, andererseits lassen sich Risiken einer Multi-Threading-Entwicklung besser einschätzen.

Java  
Best  
Practice



## Wieso parallel?

Unter einem Prozess wird in diesem Artikel ein Ausführungsstrang (Thread, Kontrollfluss) innerhalb eines Java-Programms verstanden.

Beim Multi-Threading gibt es mehrere solcher Ausführungsstränge, die parallel zueinander arbeiten und dabei auf einen gemeinsamen Speicherbereich zugreifen.

Parallele Prozesse werden eingesetzt, um die Wirkung einer parallelen Verarbeitung zu erzielen. Ein offensichtliches Beispiel ist die GUI, die während des Bildaufbaus Benutzer-eingaben zulässt.

Für Web- oder Application-Server werden Threads eingesetzt, um parallel eine Vielzahl von Anfragen zu bearbeiten.

## Wo ist das Problem?

Die Entwicklung von nebenläufigen Prozessen ist dann risikoreich, wenn parallel auf Objekte zugegriffen wird (Concurrency). Dabei werden mehrere Objektmethoden gleichzeitig ausgeführt, die Zustandsänderung ist schwer vorher-sagbar.

Im Umgang mit Concurrency ist daher die wichtigste Regel: Vermeiden Sie parallele Zugriffe auf Objekte. Andernfalls werden Sie erstaunt sein, zu welchem Zeitpunkt Fehler auftauchen und wie viel Aufwand zur Problemlösung benötigt wird.

## Ein Thread ist schnell gestartet

In jedem Fall ist Aufklärung hilfreich. In einem ersten Beispiel werden drei Lichter in rot, gelb

und grün betrachtet. Diese werden kurz hintereinander eingeschaltet und blinken dann für eine Weile. Anschließend erlöschen sie.

In Abbildung 1 ist dieser Sachverhalt in einem Aktivitätsdiagramm dargestellt. Es wird vom Startpunkt aus gelesen. Pfeile verbinden die Aktivitäten eines Kontrollflusses.

Das Einschalten des roten Lichts erfolgt am horizontalen Balken. Dort entsteht aus einem Kontrollfluss ein Zweiter: Das Blinken des Lichts findet in einem unabhängigen Kontrollfluss statt. Das Einschalten der beiden anderen Lichter führt dazu, dass zwei weitere Kontrollflüsse dazu kommen. Insgesamt gibt es vier parallele Kontrollflüsse. Nachdem alle Lichter erloschen sind, vereinen sich die vier Kontrollflüsse wieder.

Die Java Elemente sind in Abbildung 1 als Kommentar angegeben. Ein neuer Kontrollfluss wird durch die Methode `start` der Klasse `Thread` erzeugt. Objekte vom Typ `Runnable` definierten in der Methode `run` die Aktivitäten.

### Verteilung von Rechenzeit nicht vorhersehbar

Abbildung 2 enthält den Quellcode. Die innere Klasse `Licht` ist vom Typ `Runnable`. Sie implementiert das Blinken dadurch, dass zwei Mal pro Sekunde der Farbname des Lichts auf der Konsole ausgegeben wird.

In der Methode `main` wird zu jedem Licht ein `Thread` erzeugt. Gestartet wird der `Thread` – dieser führt dann die Methode `run` aus. Damit die Lichter bis zu ihrem Ende leuchten, muss `Thread.join` aufgerufen werden.

Es ist bemerkenswert, dass die Ausgaben rot, gelb und grün auf der Konsole nicht streng periodisch ausgegeben werden. Ab und an überholt ein Licht das andere. Wann und wie oft ein `Thread` **Rechenzeit** bekommt ist nicht vorhersehbar!

### Das Ende eines Threads

Mit `Thread.start()` beginnt die Ausführung des Threads. Zum Beenden gibt es keine vergleichbare Methode. Das Beenden eines Threads ist nur in Zusammenarbeit mit seinem `Runnable`-Objekt möglich. Die Methode `run` muss sich in Folge eines Signals beenden.

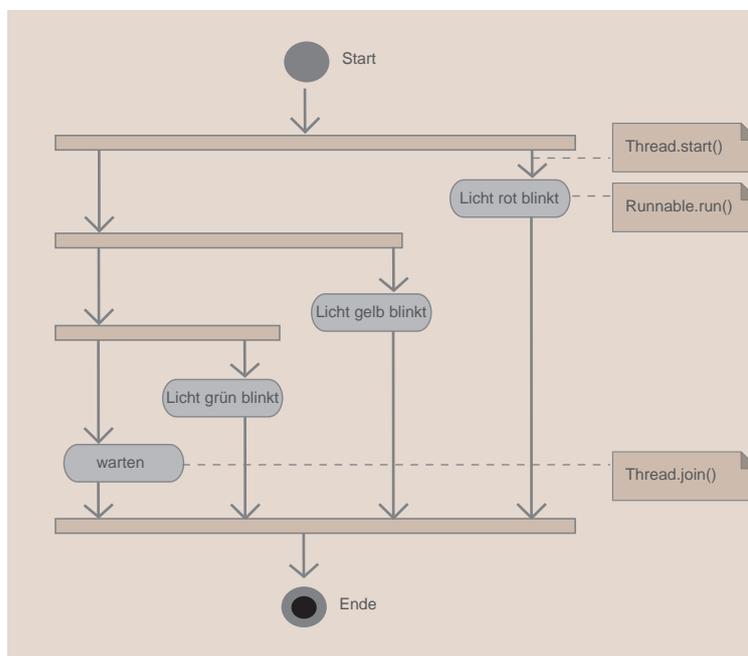


Abb. 1: Aktivitätsdiagramm zum Blinken der Lichter.

```
package de.ordix.thread;
public class Main01 {
    public static void main(String[] args) throws InterruptedException {
        class Licht implements Runnable {
            private String farbe;
            private int counter=0;
            public Licht(String farbe) {this.farbe=farbe;}

            public void run() {
                long startTime = System.currentTimeMillis();
                while (System.currentTimeMillis() - startTime <
5000) {
                    System.out.println(++counter + " " + farbe);
                    try {
                        Thread.sleep(500);
                    } catch (InterruptedException e) {
                        // nothing to do
                    }
                }
            }
        };
        Thread threadRot = new Thread(new Licht("rot"));
        Thread threadGelb = new Thread(new Licht("gelb"));
        Thread threadGruen = new Thread(new Licht("gruen"));
        threadRot.start();
        threadGelb.start();
        threadGruen.start();
        threadRot.join();
        threadGelb.join();
        threadGruen.join();
    }
}
```

Abb.2: Implementierung der blinkenden Lichter.

```

package de.ordix.thread;
public class Main02 {
    public static void main(String[] args) throws InterruptedException {
        class Licht implements Runnable {
            private String farbe;
            private int counter=0;
            public Licht(String farbe) {this.farbe=farbe;}

            public void run() {
                while (true) {
                    System.out.println(++counter + " " + farbe);
                    try {
                        Thread.sleep(500);
                    } catch (InterruptedException e) {
                        System.out.println(">>> interrupted!");
                        return;
                    }
                }
            }
        };
        Thread threadRot = new Thread(new Licht("rot"));
        Thread threadGelb = new Thread(new Licht("gelb"));
        Thread threadGruen = new Thread(new Licht("gruen"));
        threadRot.start(); threadGelb.start(); threadGruen.start();
        Thread.sleep(5000);
        // Threads unterbrechen
        System.out.println("<<< interrupt!!!");
        threadRot.interrupt(); threadGelb.interrupt(); threadGruen.interrupt();
        //.. und warten bis sie fertig sind.
        threadRot.join(); threadGelb.join(); threadGruen.join();
    }
}

```

Abb. 3: Wie werden die blinkenden Lichter ausgeschaltet?

```

public class Verkehrszaehler {
    ...
    private volatile int counter=0;
    public void weiter() {
        counter = counter + 1;
        ergebnisListe.add(System.currentTimeMillis() + " " + Integer.
toString(counter));
    }

    public static void main(String[] args) throws InterruptedException {
        class Hilfskraft implements Runnable {
            ...
            public void run() {
                while (!Thread.interrupted()) {
                    zaehler.weiter();
                }
            }
        };
        List<Thread> threadList = new ArrayList<Thread>();
        int anzahlThreads = 5;
        for(int i=0; i< anzahlThreads; i++) {
            threadList.add(new Thread(new
Hilfskraft(Verkehrszaehler.getInstance())));
        }
        for(Thread thread : threadList) {thread.start();}
        ...
        for(String zaheler : Verkehrszaehler.ergebnisListe) {
            System.out.println(zaheler);
        }
    }
}

```

Abb. 4: Simulation einer Verkehrszählung.

Im Listing der Abbildung 3 können die Lichter unendlich lange blinken. Durch den Aufruf der Methode `interrupt` bekommt das `Runnable`-Objekt das Signal, dass die Ausführung unterbrochen werden soll. In dem Beispielprogramm wird das Signal durch das Fangen der `InterruptedException` ausgewertet. Mit `return` wird die Methode `run` verlassen.

Alternativ kann die `Runnable`-Instanz an bestimmten Stellen die Methode `Thread.interrupted()` aufrufen, um zu erfahren, ob ihm ein `Interrupt` zugesendet wurde (siehe Listing in Abbildung 4).

### Implementierung einer Verkehrszählung

Im Listing der Abbildung 4 ist der Kern eines Verkehrszählungsprogramms dargestellt. Das vollständige Skript kann auf unserer Internetseite heruntergeladen werden [3]. Hilfskräfte drücken für jedes Fahrzeug auf einen Knopf, so dass weiter gezählt wird. Zur weiteren Auswertung befindet sich das Zählwerk auf einem zentralen Server.

Das Beispiel ist etwas vereinfacht: Die Hilfskraft wird durch einen Thread simuliert, der ununterbrochen weiter zählt. Das Zählwerk wird durch ein Singleton implementiert, das den Zeitpunkt und die Nummer des Fahrzeugs in einer Liste notiert.

Zur Berücksichtigung der Konkurrenz beim Zugriff auf die Liste wird eine `Collections.SynchronizedList` verwendet. Damit jeder Thread über den aktuellen Wert des Zählers verfügt, wird dieser als `volatile` gekennzeichnet.

Der Ausschnitt der Ausgabe in Abbildung 5 zeigt, dass es beim Zählen einige Unregelmäßigkeiten gibt. So werden manche Nummern beim Zählen ausgelassen – manche werden doppelt verwendet.

### Woran scheitert die Verkehrszählung?

Die Verkehrszählung scheitert dann, wenn mehrere Threads gleichzeitig die Methode `Verkehrszaehler.weiter` ausführen. Für jede Methode wird zwar ein eigener Speicherbereich für die lokalen Variablen erstellt, die Objekt-Attribute liegen dagegen nur einmalig vor. So werden diese von den verschiedenen Threads in unerwarteter Weise manipuliert.

Zur Diskussion von Konkurrenzsituationen eignen sich besonders Sequenzdiagramme. Der Vorgang `counter = counter + 1` in der Methode wird in Abbildung 6 weiter dargestellt. Immer dann, wenn **Hilfskraft 2** sich den Wert der Variable `counter` holt, bevor **Hilfskraft 1** sein Ergebnis geschrieben hat, erhöht sich der Wert des Counters nur um eins. Dabei wurden zwei Fahrzeuge gezählt.

### Das Schlüsselwort `synchronized`

Demzufolge muss verhindert werden, dass zwei Threads gleichzeitig den Wert der Variable `counter` erhöhen. Java sieht dazu eine Sperre für einen Ausführungsbereich vor. Ist diese gesetzt, so warten andere Threads, dass diese Sperre wieder zurückgenommen wird, bevor sie den Ausführungsbereich betreten.

In Abbildung 7 sind die Zeilen des Codes zu sehen, die für die Sperre sorgen. Um das Inkrementieren des Counters wird ein `Synchronized-Block` erstellt. Dazu muss ein Objekt angegeben werden, bei dem die Sperrinformation gespeichert wird. Dies ermöglicht die gleichzeitige Sperrung mehrerer Blöcke.

In der Abbildung 8 ist dargestellt, wie sich der `Synchronized-Block` auf die Ausführung auswirkt. Das Setzen der Sperre durch die **Hilfskraft 1** verhindert, dass die **Hilfskraft 2** den Block betritt. Diese muss warten. **Hilfskraft 2** wird beim Verlassen des Blocks informiert, dass er nun weiter arbeiten kann.

Wird das Schlüsselwort `synchronized` vor die Methodendefinition gesetzt, also `synchronized public weiter()`, so umfasst die Sperre den gesamten Block der Methode. Die Sperre wird dem zugehörigen Objekt zugeordnet.

### Lock, Unlock, Deadlock

In der Fachliteratur wird das Setzen der Sperre als `lock`, das Entsperren als `unlock` bezeichnet. Und dann gibt es noch den `Deadlock`. Dieser bezeichnet die Blockade, wenn zwei Threads gegenseitig auf die Freigabe eines Locks warten. In diesem Fall wird die Ausführung der beteiligten Threads unterbunden.

Wie ein `Deadlock` zustande kommt ist der Abbildung 9 zu entnehmen. Mitspieler sind dabei zwei Objekte, die synchronisierte Ausführungsblöcke enthalten: `synchable 1` und

```
1287298745640 54840
1287298745640 54841
1287298745640 54842
1287298745640 54843
1287298745640 54844
1287298745640 54846
1287298745640 54846
```

Abb. 5: Ergebnis der Verkehrszählung.

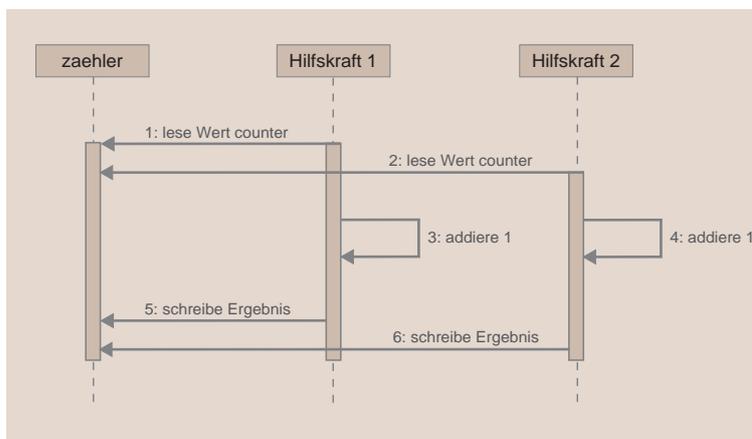


Abb. 6: Zwei Hilfskräfte führen gleichzeitig die Methode weiter aus.

```
synchronized (zaehler) {
    counter = counter + 1;
}
```

Abb. 7: Syntax des `Synchronized-Blocks`.

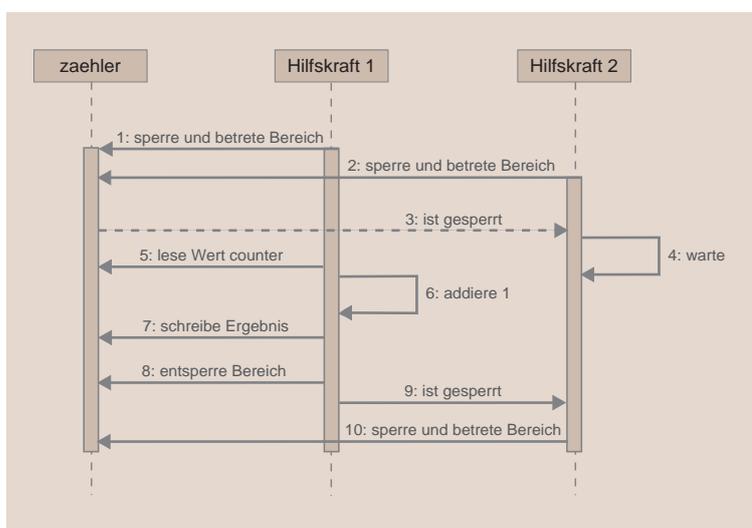


Abb. 8: Synchronisiertes Ausführen der Methode weiter.

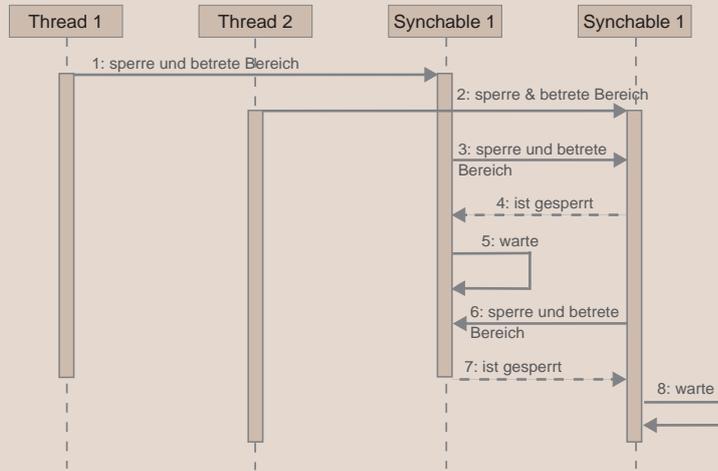


Abb. 9 Konstellation eines Deadlock.

Glossar

- Thread** Elementaraufgabe. Die Befehle eines Threads sind in sich so abgeschlossen, dass sie auf einer CPU zusammenhängend ausgeführt werden können. Um Programme mehrprozessorfähig zu gestalten, müssen die Abläufe in Threads untergliedert sein.
- Multi-Threading** Multi-Threading bedeutet, dass Programme parallel verarbeitet werden können.
- Concurrency** Konkurrierender Zugriff auf Objektattribute.

Links

- ▶ [1] Kapitel Concurrency im Java-Tutorial:  
<http://download.oracle.com/javase/tutorial/essential/concurrency/index.html>
- ▶ [2] Seminarempfehlung „Java Programmierung Grundlagen“:  
<http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Java-JEE/nr/68/index.html>
- ▶ [3] Ausführliches Listing des Verkehrszählungsprogramms:  
[http://www.ordix.de/ORDIXNews/verzeichnis\\_42010.html](http://www.ordix.de/ORDIXNews/verzeichnis_42010.html)

**synchable 2.** Diese Ausführungsblöcke werden von jeweils einem Thread betreten und damit verschlossen. Erfolgt aus dem synchronisierten Ausführungsblock jeweils der Aufruf eines synchronisierten Blocks des anderen Objekts, so ist der Deadlock perfekt. Die beiden Threads verharren auf ewig im Wartezustand.

Der Einwand - so etwas programmiert man doch nicht - ist korrekt. Solche Konstellationen ergeben sich von ganz allein. Derartige Probleme werden erst beim Lasttest der Anwendung festgestellt – wenn nicht gar erst im Betrieb.

Existiert die Möglichkeit eines Deadlocks, so laufen mehr und mehr Threads in diese Falle. Ressourcen werden nicht freigegeben. Die Anwendung wird langsam oder bricht beim Versuch ab, weitere Ressourcen zu bekommen.

Fazit

Das Fazit ist der erhobene Zeigefinger: Achtung, Multi-Threading! Auch wenn grundlegende Syntaxelemente für den Umgang mit parallelen Prozessen dargestellt werden, ergibt sich daraus noch kein Rezept für die problemlose Verwendung.

In den folgenden Artikeln dieser Serie wird daher der Aspekt des Multi-Threadings in unterschiedlichen Situationen näher untersucht und Lösungsansätze werden vorgestellt. Wer bis dahin nicht warten kann, sollte zunächst einen Blick auf das Paket `java.util.concurrent` werfen. Hier finden sich Klassen für viele Standardsituationen.

Übrigens: Die Grundlagen der Thread-Programmierung können Sie auch in unserem Seminar Java Programmierung Grundlagen erlernen. Gern stehen wir Ihnen in Entwicklungsprojekten mit unserer Erfahrung zur Seite.

- ... Teil VII: Build Management und IDE optimal kombinieren
- Teil VIII: Concurrency - Lästige Konkurrenz (1)**
- Teil IX: Concurrency - Lästige Konkurrenz (2)
- ...



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

Informix IDS V11.5

# Das „OpenAdmin Tool“ – IDS-Administration auf grafisch!

Wer anstelle der Befehle `onstat`, `onmode` etc. und/oder SQL-Mitteln (Auswertung via SMI - System Monitoring Interface) lieber eine grafische Oberfläche zur Verwaltung seiner IDS-Systeme einsetzen möchte, der sollte auf das Open Admin Tool (OAT) setzen.

Der Artikel richtet sich an IDS-Administratoren, die ein grafisches Tool bevorzugen bzw. Entwickler, die Statements in Bezug auf Performanceaspekte untersuchen wollen.

## Das OpenAdminTool

Das IBM OpenAdmin Tool (OAT) für Informix ist eine PHP-basierte Webanwendung. Dieses Werkzeug dient zur Verwaltung, Überwachung und Performanceanalyse von beliebig vielen Informix Datenbank-Servern. Das OAT ist frei erhältlich und kann unter anderem mit der Virtual Appliance für IDS installiert werden [1].

Das OpenAdmin Tool können Sie von der Herstellerseite herunterladen [2]. Hier erhalten Sie auch weitere Informationen über die Funktionsweisen, Anleitungen und Plugins.

OAT wurde mit der Version IDS V11.1 eingeführt und wird seitdem ständig weiterentwickelt. Es besteht bis auf einige Teile aus Open Source und ist durch Plug-Ins erweiterbar.

## Performance-Analyse

Wie einfach das OAT zu handhaben ist, soll hier beispielhaft an der Performance-Analyse vorgestellt werden, die unter Leistungsanalyse -- SQL-Explorer (siehe Abbildung 1) aufzurufen ist.

Auf dem Reiter "Trace-Verarbeitungsverwaltung" kann das Tracing von SQL-Anweisungen definiert werden. Dies ist global und somit für alle Benutzer oder für ausgewählte Benutzer, sowie für alle oder ausgewählte Datenbanken möglich. Ein Beispiel für das Tracing ausgewählter Datenbanken finden Sie in Abbildung 2.

Zusätzlich kann man das Tracing aktivieren, anhalten oder ausschalten. Mit den Einstel-

lungen Low, Medium, High wird das Level der Tracing-Informationen gewählt. Hierbei kann ebenfalls die für das Tracing benötigte Ressourcen definiert werden.

## SQL-Anweisungen

Die ausgeführten Kommandos (`SELECT`, `UPDATE`, `INSERT`, `CREATE`, ...) können in der Gesamtheit dargestellt, nach Typ gefiltert und zum Beispiel nach maximaler Antwortzeiten sortiert werden. So findet man leicht die Anweisungen, die die meiste Last auf dem Server verursachen (siehe Abbildung 3). Diese können dann für eine nähere Analyse

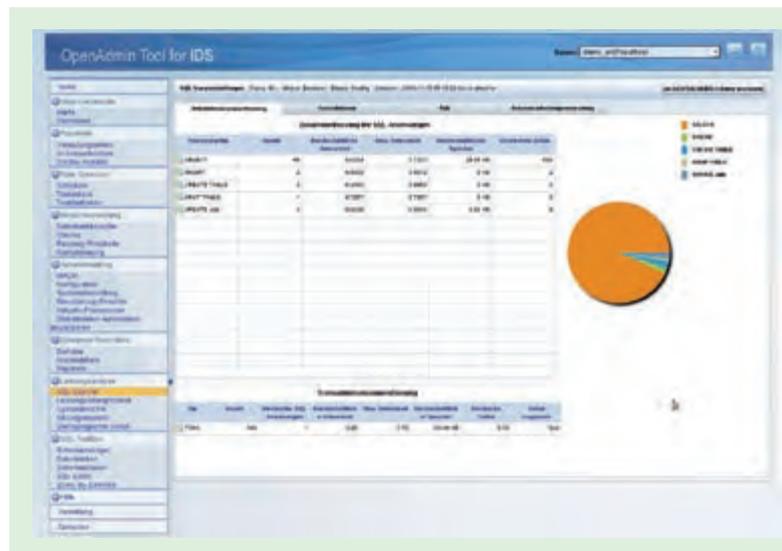


Abb. 1: Aufruf des SQL-Explorers der Performance-Analyse.



Abb. 2: Einschalten des Tracings für ausgewählte Datenbanken.

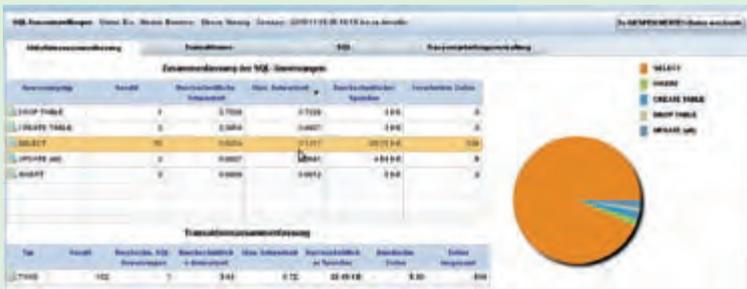


Abb. 3: Sortieren und Selektieren.

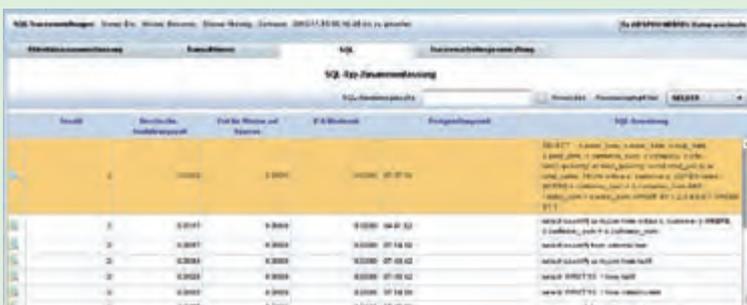


Abb. 4: Drill Down.

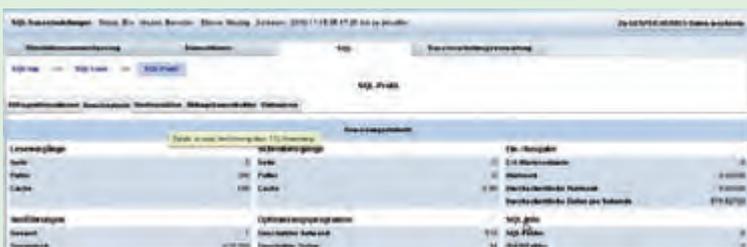


Abb. 5: Detail-Statistik.

herangezogen werden. Einmal die teuren Statements herausgefunden, stellt einen das Tunen der Anwendung nicht mehr vor allzu große Herausforderungen.

### Drill Down

Per "Drill Down" lassen sich schnell alle Anweisungen (Abbildungen 4 und 5) bis ins Detail auflösen. Hierbei wird nicht nur das eigentliche Statement angezeigt, sondern auch Informationen über Transaktionsverarbeitung und Detail-Statistiken zur Abfrage aufbereitet werden.

### Query-Plan

Der SQL-Explorer gewährt auch den Blick auf eine übersichtliche grafische Darstellung des vom Optimizer erzeugten Abfrageplans. Dies ist eine Übersetzung der bekannten "SET EXPLAIN ON" – Ausgabe (Abbildung 6), die wiederum nicht jedem in der reinen Textform leicht fällt zu interpretieren.

### SQL Administration API

Das OpenAdmin Tool bedient sich weitgehend des "SQL Administration API".

Hier einige Beispiele für den Aufruf der API in SQL:

- Zeige Informationen über den Status des Tracings:

```
SELECT task('SET SQL TRACING INFO')
FROM systables WHERE tabid=1;
```

oder

```
EXECUTE FUNCTION task('SET SQL
TRACING INFO');
```

- Füge den Benutzer user1 zur Überwachungsliste hinzu:

```
EXECUTE FUNCTION task('set sql
tracing user add','user1');
```

- Zeige die Liste der überwachten Datenbanken:

```
EXECUTE FUNCTION task('set sql
tracing database list');
```

- Füge die Datenbank sysadmin zur Überwachungsliste hinzu:

```
EXECUTE FUNCTION task('set sql
tracingdatabase add','sysadmin');
```

Wenn man die Datenbank **sysadmin** überwacht, so kann man in der Übersicht der ausgeführten SQL-Anweisungen die vom OpenAdmin Tool verwendeten SQL Administration API's verfolgen (siehe Abbildung 7).

Das kann nützlich sein, selbst wenn man das OAT nicht nutzen möchte, denn natürlich kann man die API Aufrufe auch in eigenen Programmen oder Skripten verwenden. Zusätzlich erfolgt für jede Änderung der Tracing-Vorgänge ein Eintrag in der Befehlshistorie des OAT (Protokolle – Verwaltungsbefehle).

**Zusammenfassung**

Schon dieser kurze Einblick zeigt, dass der Einsatz des OAT im Einzelfall nützlich sein kann, selbst wenn das Open Admin Tool nicht generell für die Administration der Instanzen verwendet wird.

Es bietet auf relativ einfache Weise einen Überblick und erlaubt eine elegante Methode, Performance-Probleme anzugehen und zu bearbeiten.



Werner Wellmann  
(info@ordix.de).



Abb. 6: Der Abfrageplan.

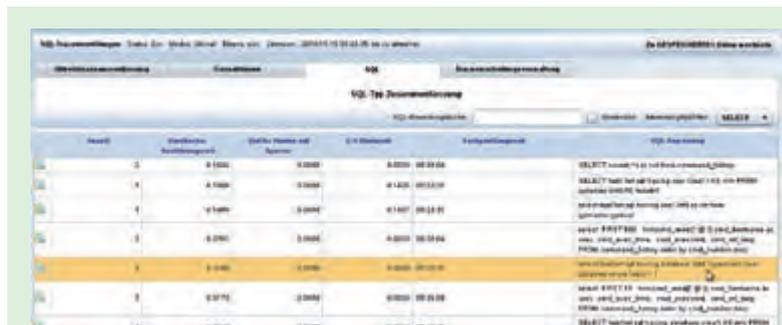


Abb. 7: Die API – Schnittstelle.

**Glossar**

- OAT** OpenAdmin Tool – web-basiertes Systemverwaltungstool für IDS ab Version V11.1.
- PHP** Hypertext Preprocessor - Scriptsprache zur Erstellung dynamischer Webseiten oder Webanwendungen.
- IDS** Informix Dynamic Server
- Optimizer** kostenbasierter Optimierer, der die Antwortzeit für SQL-Anfragen minimiert.
- Tracing** Protokollierung von SQL-Kommandos mit Performance-Informationen

**Links**

- ▶ [1] IBM Virtual Appliances:  
<https://www.ibm.com/developerworks/wikis/display/im/IBM+Virtual+Appliances>
- ▶ [2] Internetseite des OpenAdmin Tools:  
<http://www.openadmintool.com>
- ▶ [3] Informix-Seminare der ORDIX AG:  
[http://training.ordix.de/index\\_informix.htm](http://training.ordix.de/index_informix.htm)



# Wir FÜHREN Interessierte ZUSAMMEN!

Der „Treffpunkt RheinMainIT“ wird in unserem Trainingszentrum Wiesbaden durchgeführt und bietet Ihnen ein abwechslungsreiches Vortragsprogramm zu aktuellen IT-Themen.

Jede Veranstaltung beginnt um 18.00 Uhr, im Anschluss an den Vortrag rundet ein Informationsaustausch und ein Imbiss die Veranstaltung ab.

## Termine und Themen 2011

**29.03.2011**  
**Datenschutz**  
**bei Facebook & Co.**

Der Vortrag zeigt Risiken und Chancen des „Social Webs“ auf. Er soll primär sensibilisieren, verantwortungsvoll mit persönlichen Daten umzugehen.

**13.09.2011**  
**Raus aus der Krise dank**  
**Projekt-Audits**

Der Vortrag zeigt an einem Praxisbeispiel, wie man ein Projekt-Audit durchführt und welcher Nutzen erreicht werden kann.

**10.05.2011**  
**Programmierung mit Oracle**  
**Application Express (APEX)**

Der Vortrag erläutert, wie man mit APEX schnell und einfach datenbankgestützte Webanwendungen erstellen und betreiben kann.

**25.10.2011**  
**Java Software Testing**

Der Schwerpunkt des Vortrags liegt in der Ergänzung manueller Tests durch automatisierte Testverfahren.

**07.06.2011**  
**Effiziente Prozesse dank Capability**  
**Maturity Model Integration (CMMI)**

Der Vortrag erklärt sowohl die Theorie, die sich hinter CMMI verbirgt und setzt sie darüber hinaus in einen praktischen Kontext.

**08.11.2011**  
**Datenbank-Hochverfügbarkeits-**  
**lösungen für Entscheider**

Der Vortrag erläutert die grundlegenden Hochverfügbarkeitstechnologien im Datenbankumfeld.