



## ITIL – Eine Oase in der Servicewüste (Teil I) s. 6

**Oracle HTMLDB – Unendliche Weiten**  
Installation und Konfiguration s. 16

**Linux High Availability Release 2**  
Neue Herzschläge in der frei verfügbaren  
Clustersoftware Heartbeat s. 19

**MySQL 5 – The next Generation**  
Vorstellung der wichtigsten Neuerungen s. 30

**JasperReports**  
Schneller Überblick über die Java Reporting  
Engine s. 12



## Gute Vorsätze: Sparen

Wenn Sie diese Zeilen lesen, ist das neue Jahr schon über zwei Monate alt. Hatten auch Sie gute Vorsätze für 2006? Ein paar Kilos abzunehmen, das Rauchen aufzugeben oder etwas kürzer zu treten, um die Gesundheit zu schonen? Und haben auch Sie schon den einen oder anderen Vorsatz wieder über den Haufen geworfen?

Ich wünsche Ihnen, dass es sich nicht so verhält wie mit den „guten“ Vorsätzen unserer Politiker. Diese redeten zwar auch gerne vom Sparen oder Sparhaushalt, aber letztlich wurde dann doch nur an das Sparen der anderen gedacht: Rente mit 67, streichen von diesem, streichen bei jenem – Hauptsache nicht bei sich selbst.

Wenn ich sparen muss, dann nicht, indem ich meinem Edeka Kaufmann die Rendite streiche und 10 % der Samstagskasse in meine Tasche lege oder indem ich bei meinem Bäcker 5 % Anteil an den verkauften Brötchen einfordere. Sparen bedeutet für mich auch nicht, zu EU-Verhandlungen mit dem dicken Geldkoffer zu gehen, um damit Schweigen oder Zustimmung zu erkaufen.

Sparen ist für mich z. B. das Streichen von Subventionen für die, die die Subventionsgelder in den Sand gesetzt haben und den Steuerzahler und Arbeitnehmer im Nachhinein durch Massenentlassungen ein weiteres Mal die Zehne zahlen lassen.

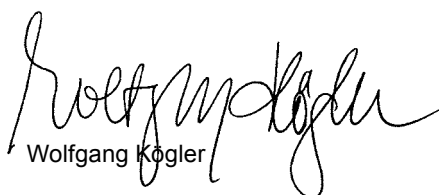
Von Sparen kann man auch nicht sprechen, wenn man die Einkaufstour so mancher Unternehmen betrachtet. Oracle ist da schon fast Weltmeister. Peoplesoft, Siebel ... kein Wunder, dass sich SAP so seine Gedanken macht, wo Oracle immer noch die meistgenutzte Datenbankplattform ist, auf der die SAP Produkte laufen.

HP steht Oracle in Punkto Akquisition kaum nach. Erstaunt es Sie auch, dass für solche Einkaufstouren Geld vorhanden ist?! Sie müssen sich nicht wundern, man hat doch durch die Trennung von einigen tausend Mitarbeitern viel Geld und Steuern gespart.

Was diese News betrifft, sparen wir auch in 2006 nicht mit interessanten Berichten. Dieses Mal habe ich persönlich den Oracle/XML Cocktail am meisten genossen. Bei den Oracle 10g Features lernen Sie neben Produktleistungsmerkmalen auch noch die römische Zählweise kennen. Immerhin sind wir schon bei Teil 7 (VII) angekommen ☺.

Und überhaupt, ich kann mir die Kritik an Oracle in den vorherigen Abschnitten erlauben, wo wir doch so viel kostenlose Werbung für das Produkt machen: Insgesamt 7 der 12 Artikel beschäftigen sich mit Oracle Features. Keine Angst. Beim nächsten Mal ist die Java Fraktion wieder stärker vertreten und mit Linux Cluster und Solaris DTrace sind ja noch Beiträge für diejenigen unter Ihnen da, die sich mehr für die Betriebssystembasis interessieren.

Ich erspare Ihnen und mir jetzt, noch etwas über Massenentlassungen bei Großkonzernen zu schreiben und über die Bemühungen der Mittelständler, dieses wieder aufzufangen. Damit sparen Sie Zeit und können sofort die Artikel lesen. In diesem Sinne warten wir aufs Frühjahr, die Tage werden wieder länger und wir können vielleicht etwas Strom sparen, denn der wird ja auch permanent teurer.

  
Wolfgang Kögler



Standards



- 02 ....Editorial
- 03 ....Inhalt
- 35 ....Impressum

Training



- 22 ....Seminarübersicht bis September 2006
- 29 .... Seminar Oracle 10g Neuheiten

Aktuell



- 26 ....Larry Ratlos  
Partitionierte Tabellen

Java/XML



- 12 ....JasperReports – Java Reporting Engine  
Schneller Überblick über die Java Reporting Engine Jasper-Reports. Titel-thema

Datenbanken



- 04 ....Neues bei Oracle: 10gR2 (Teil II)  
**Data Guard – Oracle goes Cluster**  
Das neue Oracle Feature „Fast Start Failover“ wird vorgestellt und bewertet.
- 10 ....TDE – Top Secret mit Oracle 10gR2  
Die transparente Datenverschlüsselung stellt eine Erweiterung der bisher vorhandenen Verschlüsselungsmethoden im Oracle Umfeld dar. Titel-thema
- 16 ....Oracle HTMLDB – Unendliche Weiten  
Überblick über die Konfiguration und Installation von Oracle HTMLDB.
- 24 ....PL/SQL-Neuerungen mit Oracle 10gR2  
Die Verschlüsselung mit dem WRAP Package, die Conditional Compilation und die Änderungen im Standard Package DBMS\_OUTPUT.
- 27 ....Oracle 10g New Features – Teil VII: Session Tracing  
Das Mitschneiden von Sessions ist unter Tuning-Aspekten eine sehr wichtige Funktion. Entsprechende Möglichkeiten bei Oracle 10g und die notwendigen Voraussetzungen werden vorgestellt.
- 32 ....Jobsteuerung mit dem Microsoft SQL-Server (Teil II):  
**Ereignisgesteuerte Transaktionsprotokoll-Sicherung**  
Grundlagen der Jobsteuerung mit dem Microsoft SQL-Server. Anhand eines Beispiels werden die Möglichkeiten gezeigt, Aufträge automatisch ausführen zu lassen, wenn ein Schwellwert der Datenbank-Überwachung überschritten wird.
- 40 ....Oracle und XML (Teil I): Ein besonderer Cocktail  
Überblick über die wichtigsten Komponenten, die Oracle für die Verwaltung und Generierung von XML-Daten zur Verfügung stellt.

Projektmanagement



- 06 ....ITIL – Titel-thema  
**Eine Oase in der Servicewüste (Teil I)**  
ITIL ist eine Methodensammlung und dient der systematischen, praktischen und kundenorientierten Umsetzung von IT-Service Prozessen und der Optimierung der IT-Infrastruktur. Teil I beschreibt die Service Support Prozesse.

Unix/Linux/Open Source



- 19 ....Linux-HA Release 2: Neue Herzschläge Titel-thema  
Vor einigen Monaten wurde das Release 2 des Linux High Availability Projects veröffentlicht, dessen neue Funktionen hier vorgestellt werden.
- 30 ....MySQL 5 – The Next Generation Titel-thema  
Die wichtigsten Neuerungen der Version 5 von MySQL.
- 36 ....Tiefe Einblicke in Solaris 10 mit DTrace (Teil II):  
**Mit DTrace „Ecken ausleuchten“**  
Solaris 10 bietet mit DTrace ein überzeugendes Werkzeug u. a. für Tracing und Test. Neben einem Überblick über die unterschiedlichen Provider von DTrace erfahren Sie, welche Winkel des Systems Sie mit DTrace beleuchten können.



Neues bei Oracle: 10gR2 (Teil II)

## Data Guard: Oracle goes Cluster

Oracle Data Guard ist eine im Oracle Umfeld etablierte teilautomatisierte Lösung zur Herstellung einer hochverfügbaren Umgebung, die von vielen Kunden bereits eingesetzt wird. Um die Akzeptanz zu erhöhen, bietet Oracle mit dem Release 10gR2 die Möglichkeit der vollständigen Automatisierung an. Diese wollen wir im Folgenden genauer betrachten.



Der Artikel richtet sich an Oracle Datenbankadministratoren und Systembetreuer, die Oracle Data Guard als Hochverfügbarkeitslösung für Oracle Datenbanken einsetzen.

### Historisches

Seit Oracle 9.0 hat uns jedes Release neue, wichtige Features beschert. Einige Features aus dem Bereich Verfügbarkeit möchten wir hier noch einmal aufführen:

Release	Features
9.0	Switchover, Data Guard Manager, File Management
9.2	Logical Standby Database
10.1	Real Time Apply

Alles nützliche Dinge, die Schritt für Schritt in Richtung einer automatischen Übernahme der Produktion durch eine physikalische Standby-Datenbank führen. Genau das hat Oracle mit 10gR2 realisiert. Das neue Feature heißt „Fast Start Failover“.

Wir wollen das Thema in zwei Schritten angehen. Im ersten betrachten wir das Konzept und im zweiten werden wir eine Bewertung vornehmen.

### Fast Start Failover: Idee

Bei einem Ausfall der Produktionsdatenbank oder des Netzwerks führt Data Guard automatisch einen Failover zu einer vorher definierten Standby-Datenbank durch. Dies geschieht ohne manuelle Eingriffe.

### Konzept

Die Umsetzung von Fast Start Failover zeigen die Abbildungen 1 bis 3.

In Abbildung 1 ist die Situation vor dem Fast Start Failover dargestellt. Die Redo Log Daten werden von der Produktionsdatenbank zur Standby Seite übertragen. Der Observer hat Kontakt zu den beteiligten Datenbanken.

Abbildung 2 zeigt die Failover-Situation. Der Observer hat keinen Kontakt mehr zur Produktionsseite. Die Standby-Seite wird produktiv.

Nach der Beseitigung der Störung bekommt der Observer wieder Kontakt zur alten Produktionsdatenbank. Die Datenbank wird automatisch zur Standby-Datenbank gemacht (siehe Abbildung 3).

### Voraussetzungen

Fast Start Failover funktioniert natürlich nicht von selbst. Um es einsetzen zu können, fällt zunächst einiges an Konfigurationsarbeiten an.

### Flashback

Auf beiden Datenbanken muss Flashback Database eingeschaltet sein. Dabei reicht es aus, den Parameter `DB_FLASHBACK_RETENTION_TARGET` auf einen Wert von 10 Minuten zu stellen, um die Fast Start Failover Funktionalität sicherzustellen.

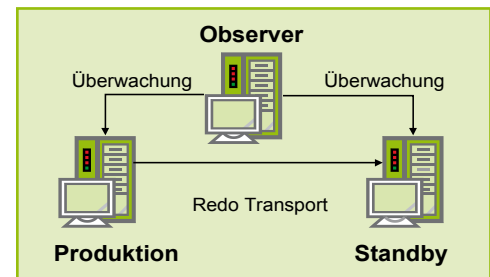


Abb. 1: Situation vor dem Fast Start Failover.

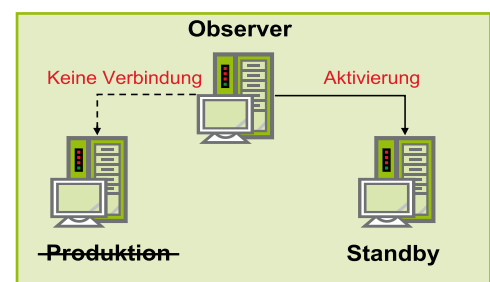


Abb. 2: Failover Situation.

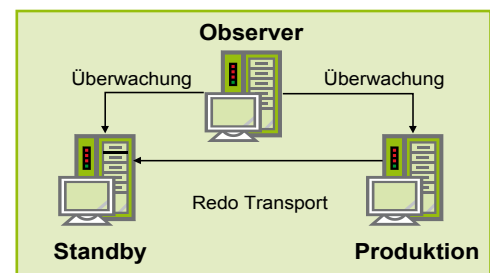


Abb. 3: Situation nach der Fehlerbeseitigung.

## Glossar

<b>Observer</b>	„Der Beobachter“ (wörtlich). Im Zusammenhang mit Oracle Data Guard bedeutet es aber viel mehr. Denn ohne den Observer gibt es keine „Schiedsrichter“- oder „Richter“-Funktion. Und ohne diese keinen Fast Start Failover.
<b>Real Time Apply</b>	Die Daten werden sofort nach dem Transfer auf den Standby Server wieder hergestellt.
<b>Flashback Database</b>	Eine Technik, ein Point in Time Recovery einer Oracle Datenbank ohne das Einspielen eines Backups durchzuführen.
<b>Data Guard Broker</b>	Die Management-Schnittstelle zu Oracle Data Guard.

Lautet die Anforderung an die Konfiguration Fast Start Failover und Point in Time Recovery gleichzeitig, so ist **DB\_FLASHBACK\_RETENTION\_TARGET** natürlich größer zu wählen.

## Real Time Apply

Auch wenn es keine zwingende Voraussetzung ist, so empfiehlt es sich im Sinne einer schnelleren Übernahme, Real Time Apply einzuschalten, um jegliche Verzögerung durch ein eventuelles notwendiges Recovery zu vermeiden.

## Broker-Konfiguration

Natürlich müssen die Datenbanken Bestandteil einer Data Guard Broker Konfiguration sein. Damit der Failover möglichst verlustfrei funktioniert, ist dafür die Variante Maximum Availability zu wählen. Damit ist garantiert, dass die synchrone Übertragung der Redo Log Informationen per Logwriter-Prozess der Standard ist.

## Observer

Auf dem Server, der die Observer-Funktion übernehmen soll, muss selbstverständlich ebenfalls die Oracle Software installiert werden.

Erfolgt die Administration ausschließlich über das Kommandozeilen-Werkzeug DGMGRL, so reicht es, bei der Installation die Variante Administrator Client Kit zu installieren. Bei Nutzung von Grid Control muss das gesamte Oracle Software Paket installiert werden.

Natürlich ist auch Oracle Net auf dem Observer vollständig zu konfigurieren, damit der Observer Kontakt zu den beteiligten Datenbanken aufnehmen kann.

## Konfiguration

Letztlich bleiben vier Dinge zu tun:

1. FastStartFailoverTarget: Das ist diejenige Standby-Datenbank innerhalb der Konfiguration, auf die umgeschaltet werden soll (nur notwendig bei mehr als einer Standby-Datenbank innerhalb der Konfiguration):  

```
DGMGRL> EDIT DATABASE 'ordixprod' SET PROPERTY FastStartFailoverTarget 'ordixstdby1';
DGMGRL> EDIT DATABASE 'ordixstdby1' SET PROPERTY FastStartFailoverTarget 'ordixprod';
```
2. FastStartFailoverThreshold: Nach welcher Zeit (in Sekunden) erfolgloser Kontaktversuche soll umgeschaltet werden  

```
DGMGRL> EDIT CONFIGURATION SET PROPERTY FastStartFailoverThreshold = 45;
```
3. Aktivierung des Fast Start Failovers:  

```
DGMGRL> ENABLE FAST_START FAILOVER;
```
4. Starten des Observers: 

```
DGMGRL> START OBSERVER;
```

Wird der Observer nicht sofort nach dem Aktivieren des Fast Start Failovers gestartet, erhalten wir die folgende Fehlermeldung:

```
Current status for "ordixprod":
Warning: ORA-16819: Fast-Start Failover observer not started
```

## Ereignisse, die einen Fast Start Failover auslösen

Bei folgenden Ereignissen wird ein Fast Start Failover ausgelöst:

- Instanzfehler: betrifft bei RAC nur die letzte Instanz
- Shutdown abort: Shutdown immediate, transactional oder normal lösen keinen Fast Start Failover aus
- Offline setzen von Datenbankdateien aufgrund von I/O-Fehlern
- Sowohl der Observer, als auch die Standby-Datenbank verlieren den Kontakt zur Produktionsdatenbank. Zusätzlich bestätigt die Standby-Datenbank, dass sie synchronisiert ist.

## Fazit

Bisher (10gR1) war ein Failover immer mit anschließender Handarbeit verbunden und daher keine echte Option für eine automatisierte Lösung. Fast Start Failover erweitert jedoch die bisherigen Data Guard Funktionalitäten in Richtung Betriebssystem Cluster-Ersatz. Die alte Produktionsdatenbank wird nach einem Failover automatisch zur neuen Standby-Datenbank. Damit ist es möglich, mit Data Guard echte 7x24-Konfigurationen zu bauen. Ohne Cluster Know-how und ausschließlich mit dem Datenbankadministratoren bekannten Mitteln.

Und das Ganze mit einem echten Mehrwert. Betriebssystem-Cluster oder RAC stellen eine mehr oder weniger hochverfügbare Instanz zur Verfügung. Mit Data Guard Fast Start Failover erhalten wir aber eine 7x24-Lösung für die Datenbank.

Sie wollen mehr über Data Guard und Fast Start Failover erfahren? Es selbst testen? Besuchen Sie unseren viertägigen Data Guard Workshop in Wiesbaden. Detaillierte Seminarinhalte und -termine finden Sie in unserem Trainingsshop unter <http://training.ordix.de>. Schauen Sie mal rein!

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



## IT-Service Management mit ITIL (Teil I): Service Support

# ITIL – Eine Oase in der Servicewüste

Der Begriff „ITIL“ ist zum Trendwort in der IT-Welt geworden. Wir erläutern, was sich hinter diesen vier Buchstaben verbirgt und welchen Nutzen Sie aus ITIL bei der Optimierung von IT-Dienstleistungen und IT-Infrastrukturen ziehen können. Im vorliegenden ersten Teil stellen wir die Prozesse im Operativen Bereich der IT-Dienstleistungen, genannt Service Support, vor.



Der Artikel richtet sich an DV-Leiter, IT-Projektmanager, Organisatoren und Entscheider.

### Was ist ITIL?

Fangen wir damit an, was ITIL NICHT ist: ITIL ist kein Standard, ITIL ist kein Dogma und ITIL ist mit Sicherheit kein Patentrezept. ITIL steht für „IT-Infrastructure Library“ und ist eine Sammlung von sieben Büchern, die in der Praxis als Rahmenwerk für Prozesse in der IT gelten.

In den späten 80er Jahren wurden in Großbritannien die ersten ITIL-Richtlinien von der „Central Computer and Telecommunications Agency“ (CCTA), dem heutigen „Office of Government Commerce“ (OGC) veröffentlicht. Im Laufe der Zeit wurden von Lieferanten, Beratern und IT-Anwendern immer mehr und sogar sehr detaillierte Ratschläge und Vorschläge entwickelt und in über 40 Büchern dokumentiert.

Nach einer größeren Revision in den Jahren 2000/2001, die eine Bündelung und Aktualisierung der Materialien, das Entfernen von Redundanzen und eine bessere Navigation beinhaltete, sind diese sieben Bücher entstanden. Die heute vorliegenden „ITIL Best Practice“-Empfehlungen stellen eine in der Praxis bewährte Methodensammlung zur systematischen, praktischen und kundensorientierten Umsetzung von IT-Service Prozessen und für die Optimierung der IT-Infrastruktur dar. Sie umfassen sowohl die Unternehmenssicht als auch die Technologiesicht. Abbildung 1 zeigt die Rahmenstruktur der ITIL-Publikationen.

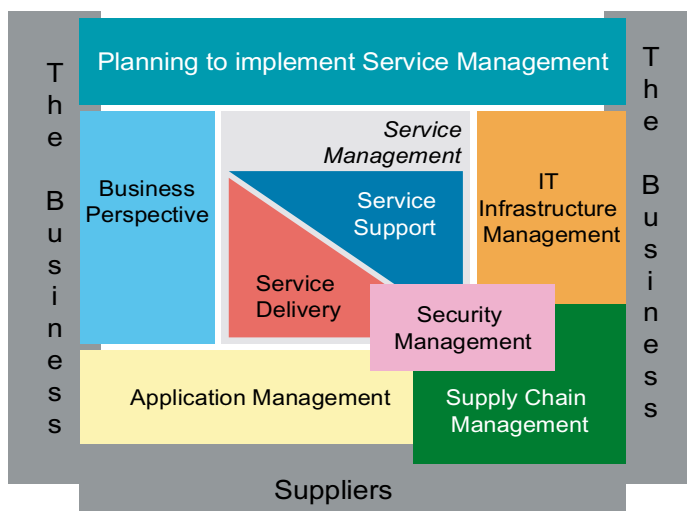


Abb. 1: Die Rahmenstruktur der ITIL-Publikationen. (Quelle: www.itil.org)

### Service in der Praxis

Viele Unternehmen verfügen über eine Hotline, einen User Help Desk und auch über Strukturen im Bereich Problem- und Change-Management. Natürlich verfügen alle diese Unternehmen auch über irgendwelche Serviceprozesse.

Aber wie ist die Qualität? Sind die Anwender zufrieden? Sind die Kunden zufrieden? Ist es vertretbar, dass Sachbearbeiter längere Zeit nicht arbeiten können, weil ein Fehler im Netzwerk nicht behoben oder ein defekter PC nicht repariert wird? Oder dass die Projektteams ihre Arbeit nicht aufnehmen, weil noch keine Accounts eingerichtet wurden? Dass man tagelang auf eine E-Mail-Adresse wartet?

Wenn man sich die Kosten für solche Ausfallzeiten anschaut, ist das sicher nicht akzeptabel! Viele dieser Vorfälle hätten vermieden werden können, wenn man sich bei der Prozessdefinition des IT-Service Managements aus dem ITIL-Methodensammlung bedient hätte.

Die ITIL-Prozesse sind in zwei Gruppen unterteilt:

1. Operativer Bereich der Unterstützung von IT-Dienstleistungen, genannt „Service Support“
2. Taktischer Bereich der Planung und Steuerung von IT-Dienstleistungen, genannt „Service Delivery“.

Abbildung 2 zeigt einen Überblick über die ITIL-Prozesse.

### Service Support

Die Service Support Prozesse betreffen den Umgang mit dem Anwender/Kunden und die Maßnahmen, mit denen dieser Umgang so gut wie irgend möglich gestaltet wird. Sie stellen sicher, dass alle Fragen, Beschwerden, Anforderungen und Service-Nachfragen optimal bearbeitet werden.

Zu diesem Bereich gehören die Funktion „Service Desk“ als Teil des Incident Managements sowie die folgenden Prozesse:

- Incident Management
- Problem Management
- Change Management
- Release Management
- Configuration Management

## Service Desk

Kennen Sie das? Anruf, schöne Musik, Standardfragen, aber keine Hilfe? Die erste Aufgabe eines gut organisierten Service Desks ist laut ITIL, möglichst rasch Hilfe zu leisten und wenn das nicht möglich ist, an das Incident-Management weiterzuleiten.

Das darf aber nicht damit verwechselt werden, dass ein Service Desk ein hilfloses Call-Center sein darf, im Gegenteil.

Der Service Desk ist eine Funktion innerhalb des Incident Managements. Er ist die klar definierte Schnittstelle „Single Point of Contact“ zwischen den Anwendern, Kunden und der IT-Organisation sowie zu weiteren IT-Service Prozessen.

Seine Aufgabe ist es, sämtliche Anfragen, Beschwerden und Störungen anzunehmen, zu dokumentieren und zum Teil selbst zu bearbeiten. Wenn das nicht möglich ist, leitet er jede Anfrage in die richtige Richtung weiter.

Besonders wichtig ist die Behandlung von Störungen. Eine Störung oder Incident im Sinne von ITIL ist jedes Ereignis, das eine Service-Unterbrechung oder eine Verschlechterung des Services zur Folge hat. Hier nimmt der Service Desk eine erste Einschätzung vor und versucht, den Incident selbst zu beheben (1st Level Support).

Zunächst wird in der „Known Error Database“ geprüft, ob der Incident bereits bekannt ist (siehe Abschnitt „Problem Management“) und ob bereits Umgehungsmöglichkeiten (Workarounds) oder Lösungen existieren, die wirksam eingesetzt werden können.

Wenn das alles nicht möglich ist, leitet der Service Desk den Incident weiter zum 2nd Level Support. Außerdem überwacht er die gemeldeten Incidents und löst gegebenenfalls den Eskalationsprozess aus.

Als „Single Point of Contact“ hält er den Kontakt zu Anwendern und Kunden, um diese ständig über ihre Anfragen auf dem Laufenden zu halten. Weiterhin erstellt er wichtige Informationen für das Management, wie z. B. eine Störungsstatistik.

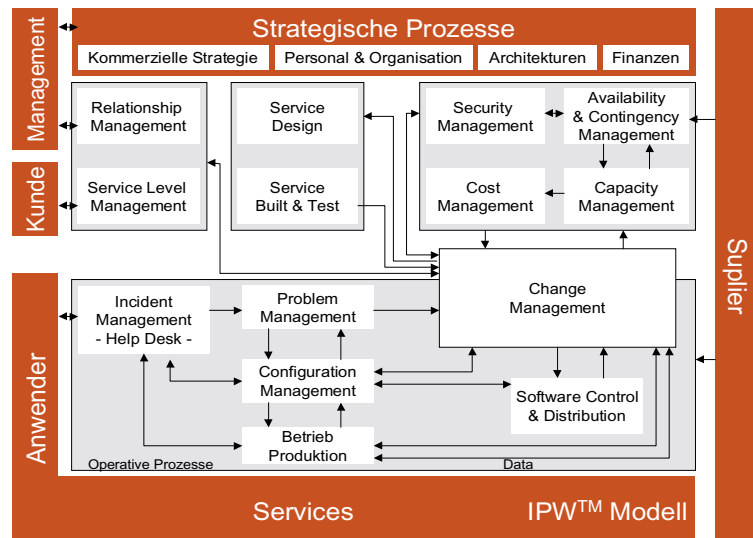


Abb. 2: Überblick über die gesamten ITIL-Prozesse. (Quelle: www.itil.org)

## Incident Management

Wenn eine Störung nicht durch schnelle „erste Hilfe“ durch den Service Desk behoben werden kann, wird sie an das Incident Management weitergeleitet. Ziel des Incident Managements ist die schnellstmögliche Wiederherstellung eines störungsfreien Services innerhalb der mit dem Kunden/Anwender vereinbarten Zeit.

Die Grundlage dafür liefern die sogenannten Service Level Agreements (SLA), die wir im 2. Teil dieses Artikels in einer der nächsten ORDIX News im Abschnitt „Service Level Management“ vorstellen werden. Im 2nd Level Support folgt eine genauere Klassifizierung des Incidents.

Man begutachtet, inwieweit der gemeldete Incident die Arbeit der Anwender/Kunden beeinträchtigt. Daraufhin werden weitere Hilfsmaßnahmen eingeleitet, z. B. neue Workarounds geschaffen.

Wenn auf diese Weise nicht geholfen werden kann, wird gegebenenfalls auch Hilfe vom Hersteller oder Lieferanten in Anspruch genommen (3rd Level Support). Im Erfolgsfall wird der Incident dann dem Service Desk als behoben gemeldet, ansonsten wird dies ein Fall für das Problem Management.

## Problem Management

Leider geht das nicht immer so reibungslos. Der Incident wird im wahrsten Sinne des Wortes zum Problem. Ein Problem im Sinne von ITIL ist die unbekannte Ursache einer oder mehrerer Incidents. Meist wird ein Incident zum Problem erklärt, wenn er

- häufig auftritt,
- die Arbeit der Anwender/Kunden stark beeinträchtigt,
- die vereinbarten SLAs gefährdet,
- und/oder das Incident Management keinen zufriedenstellenden Workaround findet.

Ziel des Problem Managements ist es, die Ursache eines Problems zu finden und zu vermeiden, dass dieses wieder auftritt. Das Problem Management arbeitet eng mit dem Incident Management zusammen.



Die Probleme sowie praktikable Workarounds werden in der „Known Error Database“ dokumentiert. Das Problem Management forscht nach der Fehlerursache und prüft, ob es bereits Workarounds gibt.

Manchmal werden bestehende Workarounds verbessert oder für das spezifische Problem angepasst. Wenn das nicht möglich ist, wird ein neuer Lösungsansatz entwickelt. Das Problem ist nun zum bekannten Fehler „Known Error“ geworden.

Je nachdem wie sehr die Behebung eines Problems die Arbeit der Anwender/Kunden beeinträchtigt, erstellt das Problem Management einen Änderungsantrag „Request for Change“ an das Change Management (siehe Abschnitt „Change Management“).

Sollte dem Antrag stattgegeben werden und die gewünschte Änderung durchgeführt werden, führt das Problem Management anschließend ein Review durch (PIR: Post Implementation Review). Wird dabei festgestellt, dass der Service wieder hergestellt ist, wird auch das Problem abgeschlossen und somit auch der anfängliche Incident. Weiterhin trägt das Problem Management durch proaktive Untersuchungen und Analysen vorhandener Probleme maßgeblich zur Verringerung des Aufkommens von Incidents bei.

### Configuration Management

Ohne detaillierte Informationen über Hardware, Software usw. wären die zuvor genannten Leistungen gar nicht möglich. Aus diesem Grunde ist das Ziel des Configuration Managements die Dokumentation aller servicerelevanten Elemente (CI = Configuration Item) und deren Beziehung zueinander.

Ein CI ist eine Komponente der IT-Infrastruktur, die zur Erbringung der Services erforderlich ist, wie z. B. Server, Drucker und SLAs. Die Erfassung der Beziehungen der CIs, z. B. Kunde ABC nutzt den Drucker XYZ über SLA 4711, ist der wesentliche Unterschied zu einem normalen Asset Management.

Das Configuration Management sorgt dafür, dass alle Informationen zu den CIs sorgfältig aufgenommen und gepflegt werden. Es stellt eine wichtige Informationsquelle für die anderen Service Prozesse dar, insbesondere für Problem, Release und Change Management.

Das Kernstück des Configuration Managements ist die Configuration Management Database (CMDB), in der alle relevanten Informationen, z. B. Typ, Kategorie und Beziehungen gespeichert werden. Beim Aufsetzen der CMDB sollte der Detaillierungsgrad genau festgelegt werden.

Eine zu tiefe Detaillierung führt zu Unübersichtlichkeit, eine zu schwache ist zu oberflächlich und liefert nicht die gewünschten Informationen. So kann in Zusammenarbeit mit dem Change- und Release Management „Wildwuchs“ im Hardwarebereich sowie die Nutzung unautorisierter Software vermieden werden. Weiterhin liefert das Configuration Management wertvolle Informationen und Entscheidungshilfen für das Management.

### Change Management

Ziel des Change Managements ist der Einsatz von standardisierten Methoden und Verfahren für eine effiziente und prompte Handhabung aller Änderungen. ITIL versteht unter einem Change das

Hinzufügen, Ändern oder Beseitigen eines Configuration Items.

Änderungen können aus unterschiedlichen Gründen erforderlich werden, sei es durch Gesetzesänderungen, durch Anforderungen aus dem Problem Management, durch die Geschäftsleitung usw. Ein Request for Change (RfC) im Sinne von ITIL ist ein Antrag auf die Durchführung einer Änderung an einem CI.

Das Change Management schätzt die Auswirkungen ab, die der Change mit sich bringen könnte. Ebenso werden die Kosten, der Nutzen, die Risiken, sowie die Vor- und Nachteile ermittelt. Das Change Management unterstützt auch bei der Formulierung der Begründung für die Änderung. Nach Annahme und Dokumentation des RfC wird die Priorität festgelegt. Die RfCs werden in der Regel kategorisiert nach

- **gering (minor):** Die Auswirkungen des Change sind eher gering und man braucht wenig Ressourcen zur Umsetzung.
- **bedeutend (significant):** Die Auswirkungen sind deutlich größer und man braucht mehr Ressourcen zur Umsetzung.
- **erheblich (major):** Die Auswirkungen des Change sind weitreichend und man braucht entsprechend viele Ressourcen zur Umsetzung.

Nach der Klassifizierung entscheidet der Change Manager bei Minor Changes über die weitere Vorgehensweise und berichtet an das Change Advisory Board (CAB). Inwieweit er das darf, wird bei der Implementierung des Change Managements definiert.

Significant Changes werden an einen Entscheidungsausschuss, das CAB, weiter gegeben. Major Changes werden außerdem noch der Geschäftsleitung zur Genehmigung vorgelegt. Nach dem Genehmigungsprozess verwaltet und koordiniert das Change Management die Durchführung der Änderung und der Implementierung.

Besonders dringende Changes („Hotfixes“) können innerhalb eines gesonderten Prozesses durch das CAB/EC (Emergency Comitee des Change Advisory Boards) schnell genehmigt werden. Weiterhin stellt das Change Management relevante Informationen und Berichte über die Änderung zur Verfügung.

Nach Erfolgsmeldung durch das Problem Management wird der RfC geschlossen. Das Change Management arbeitet sehr eng mit dem Problem und Release Management zusammen.

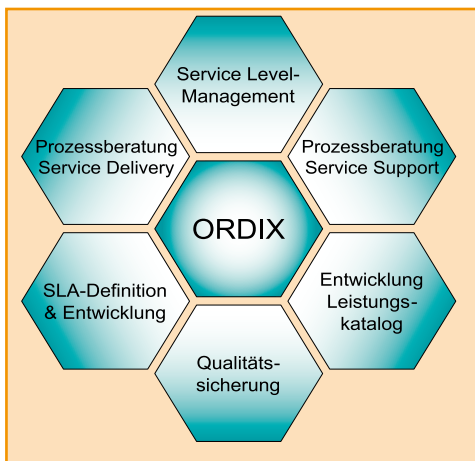


Abb. 3: ORDIX Leistungen im ITIL-Umfeld.

## Release Management

Wenn Hardware, Software oder auch geänderte Hard- oder Software eingeführt werden sollen, kommt das Release Management ins Spiel. In Zusammenarbeit mit dem Change und Configuration Management hat es den Überblick über alle Änderungen an den IT-Services und CIs. So kann sicher gestellt werden, dass alle Aspekte einer Änderung – ob technisch oder nicht-technisch – berücksichtigt werden.

Ein Release im Sinne von ITIL ist die Sammlung von neuen und/oder geänderten CIs, die getestet und zusammen in die Produktivumgebung eingeführt werden. Die wesentlichen Aufgaben sind

- Planung, Überwachung und Dokumentation dieser Einführungen
- Sicherstellen, dass neue Software intensiv und ausreichend getestet wird
- Abstimmung mit dem Change Management über die Inhalte der neuen Releases
- Sicherstellen der Dokumentation aller Änderungen in der CMDB

Eine weitere wichtige Aufgabe ist auch die Verwaltung der Definite Software Library (DSL), in der alle autorisierten Versionen der eingesetzten Software, egal ob Eigenentwicklungen oder zugekaufte Software, gespeichert werden.

Die DSL kann auch als Basis für eine Softwareverteilung im Unternehmen genutzt werden. Durch die DSL entsteht Transparenz über den Softwarebestand. Die Nutzung unautorisierter Software kann auf diese Weise vermieden werden.

Auch die Pflege des Definitive Hardware Store gehört zu den Aufgaben des Release Managements. Dieser beinhaltet die definierte und au-

## Glossar

### IT-Infrastructure Library (ITIL)

Richtlinienrahmen zur Gestaltung von IT-Prozessen: Die ITIL Library ist eine Sammlung von sieben Büchern und stellt eine in der Praxis bewährte Methodensammlung dar. Sie dient der systematischen, praktischen und kundenorientierten Umsetzung von Prozessen und der optimierten Gestaltung der IT-Infrastruktur.

### Central Computer and Telecommunications Agency (CCTA)

Heute OCG (Office of Government Commerce): Eine Organisation, die für die britische Regierung arbeitet. Sie arbeitet an der Verbesserung der Effizienz und Effektivität der Geschäftsprozesse der Regierung.

### Incident

Eine Störung oder Incident im Sinne von ITIL ist jedes Ereignis, das eine Serviceunterbrechung oder eine Verschlechterung des Services zur Folge hat.

### Post Implementation Review (PIR)

Nachträgliche Kontrolle der Qualität eines Changes in der Produktivumgebung.

### Service Level Agreement (SLA)

Vereinbarung über eine bestimmte Leistung, Art und Umfang eines zur Verfügung gestellten Dienstes.

### Configuration Items (CIs)

Komponente der IT-Infrastruktur, die zur Erbringung der Services erforderlich ist, wie z. B. Server, Drucker und SLAs.

### Configuration Management Database (CMDB)

Kernstück des Configuration Managements, in dem alle relevanten Informationen zu den Configuration Items, z. B. Typ, Kategorie und Beziehungen gespeichert werden.

### Request for Change (RFC)

Im Sinne von ITIL ist das ein Antrag auf die Durchführung einer Änderung an einem Configuration Item.

### Change Advisory Board (CAB)

Gremium, welches bei der Bewertung der Auswirkungen, sowie bei der Planung der Umsetzung eines Changes unterstützt.

### Emergency Committee des CAB (EC Service Delivery)

Das „Notfall-Komitee“ des Change Advisory Boards.

Taktische Prozesse zur Planung und Steuerung der IT-Dienstleistungen.

### Incident Management

Schnelle und wirksame Lösung von Problemen, sowie das Vorbeugen und Reduzieren von Störungen.

torisierte Ersatz-Hardware, die ebenso wie die Produktiv-Hardware regelmäßig gewartet werden soll.

## ORDIX kann helfen!

Natürlich ist es mit der Zurverfügungstellung von ITIL-Prozessen und -Methoden alleine nicht getan. Die Einführung neuer Strukturen und Prozesse bringt Veränderungen mit sich. Und Veränderungen erzeugen immer auch Unsicherheit oder gar Angst bis hin zu bewusstem oder unbewusstem Widerstand. Daher muss im Vorfeld Akzeptanz und Klarheit für die neuen Strukturen und Prozesse geschaffen werden. Wir haben die Erfahrung bereits in vielen Projekten gemacht, dass es sich lohnt, für die Akzeptanz Zeit und Aufwand zu investieren.

In Teil II des Artikels stellen wir die taktischen Prozesse der Service Delivery vor.

Christiane Westerhoff und Rainer Restat ([info@ordix.de](mailto:info@ordix.de)).



## Datenverschlüsselung mit Oracle 10g Release 2:

# TDE – Top Secret mit Oracle 10gR2

Um sensible Daten vor unberechtigten Zugriffen zu schützen, werden verschiedene Techniken eingesetzt. Neben Authentifizierung und Zugriffskontrolle kann zusätzlich Datenverschlüsselung eingesetzt werden. Dieser Artikel behandelt das neue Feature „Transparente Datenverschlüsselung“ im Release 2 von Oracle 10g.



Dieser Artikel richtet sich an Oracle Administratoren und Entwickler, die sich in der Version Oracle Database 10g Release 2 mit der Datenverschlüsselung beschäftigen.

### Datenverschlüsselung Version 8 und 10g Release 1

Mit Hilfe der Datenverschlüsselung wird ein zusätzlicher Schutz vor unberechtigten Zugriffen von Dritten auf sensible Daten und deren Manipulation und Missbrauch erreicht, da die Daten mit einem Verschlüsselungsalgorithmus und einem Schlüssel codiert werden. Seit der Version 8i stellt Oracle das Package *DBMS\_OBFUSCATION\_TOOLKIT* zur Verfügung. In der Version 10g Release 1 kam das Package *DBMS\_CRYPTO* hinzu.

### Datenverschlüsselung Version 10g Release 2

In dem nun beschriebenen Release 2 von Oracle 10g gibt es die so genannte „Transparente Datenverschlüsselung“ (Transparent Data Encryption, kurz TDE) als zusätzliche Möglichkeit, die Daten innerhalb der Datenbank (also auch auf der Platte) und auch im Netzwerk zu verschlüsseln (siehe Abbildung 1).

Mit TDE ist es möglich, Tabellenspalten ohne zusätzlichen Programmieraufwand zu verschlüsseln. Selbst beim Einsatz von Views entsteht kein administrativer Mehraufwand.

Ist TDE für eine oder mehrere Spalten einer Tabelle eingerichtet, findet bereits beim **INSERT** in eine Tabelle automatisch eine Verschlüsselung der Daten statt. Bei der Abfrage der Daten aus einer Tabelle werden die Daten automatisch entschlüsselt. TDE unterstützt 3DES (Data Encryption Standard) und AES (Advanced Encryption Standard) Verschlüsselungsalgorithmen.

### Anwendung der transparenten Datenverschlüsselung

Die Verschlüsselung kann sehr einfach eingerichtet werden. Soll eine Tabellenspalte mit TDE verschlüsselt werden, so kann diese Information bereits beim Anlegen einer Tabelle mit dem Schlüsselwort **ENCRYPT** angegeben werden. In Abbildung 2 wird eine Tabelle **mitarbeiter** mit einer verschlüsselten Spalte **gehaltsguppe** unter Nutzung des Verschlüsselungsalgorithmus „AES128“ angelegt.

Um eine Tabellenspalte im Nachhinein als verschlüsselt zu kennzeichnen, kann das Kommando **ALTER TABLE** verwendet werden. Die einzelnen, verschlüsselten Tabellenspalten lassen sich über die Datenbanksicht **DBA\_ENCRYPTED\_COLUMNS** ermitteln. Die Verwendung dieser und folgender Kommandos wird in Abbildung 2 gezeigt.

Mit dem Schlüsselwort **DECRYPT** kann eine Verschlüsselung auf einer Tabellenspalte wieder deaktiviert werden.

### Master-Schlüssel

Wird eine Spalte mit dem Kommando **ENCRYPT** als verschlüsselt gekennzeichnet, so wird implizit ein Schlüssel für die Tabelle zum Verschlüsseln erzeugt und im Data Dictionary abgelegt. Aus Sicherheitsgründen wird dieser Schlüssel mit einem Master-Schlüssel verschlüsselt.

### Wallet

Der Master-Schlüssel wird wiederum in einem Wallet abgelegt, das sich standardmäßig auf dem Datenbankserver unter dem in Abbildung 2 genannten Verzeichnis befindet. Die Ablage der Wallets kann in der `sqlnet.ora` nach Bedarf geändert werden. Das Wallet dient zur Ablage von Master-Schlüsseln. Es zeichnet sich dadurch aus, dass es durch ein Passwort geschützt ist und über den Oracle Wallet Manager (OWM) verwaltet werden kann.

Ein Wallet kann mit dem Kommando **ALTER SYSTEM** erzeugt werden. In Abbildung 2 wird ein Wallet angelegt und sein Kennwort auf „geheim“ gesetzt. Gleichzeitig wird es für die transparente Datenverschlüsselung geöffnet.

### Wallet und Tabellen

Beim Einfügen von Datensätzen in eine Tabelle mit einer verschlüsselten Spalte wird zuerst der Master-Schlüssel aus dem Wallet ermittelt.

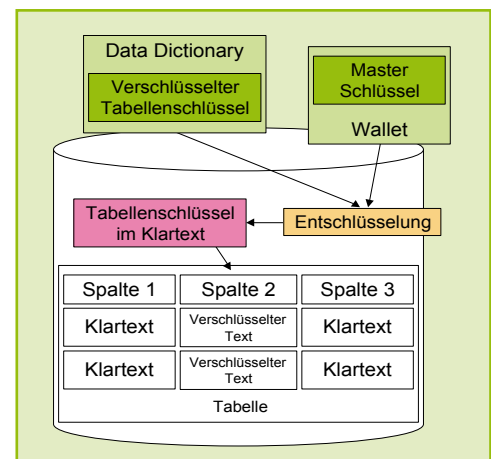


Abb. 1: Die transparente Datenverschlüsselung in ihrer schematischen Übersicht.

Danach wird mit Hilfe des Master-Schlüssels der Tabellenschlüssel aus dem Data Dictionary entschlüsselt. Anschließend werden mit dem entschlüsselten Tabellenschlüssel die in die Tabelle eingefügten Daten verschlüsselt und in der Datenbank abgelegt.

Außerdem besteht die Möglichkeit, bereits verschlüsselte Tabellenspalten mit einem neuen Kodieralgorithmus neu zu verschlüsseln. Mit der **REKEY** Anweisung wird für die Tabelle **mitarbeiter** ein neuer Tabellenschlüssel erzeugt. Gleichzeitig werden die bereits verschlüsselten Spalten mit dem Kodieralgorithmus „AES256“ neu verschlüsselt.

### Data Pump und TDE

Beim Extrahieren von Daten in eine Datei mit dem Werkzeug Data Pump (ab Oracle Version 10) werden die Daten standardmäßig im Klartext (also unverschlüsselt) abgelegt. Lediglich eine Warnung weist darauf hin, dass eventuell verschlüsselte Daten aus einer Tabelle unverschlüsselt in eine Datei gespeichert werden.

Um die mit Data Pump erzeugten Dateien zu schützen, kann mit der Option **ENCRYPTION\_PASSWORD** ein Kennwort für die erzeugten Dateien festgelegt werden.

Sollen die mit Data Pump extrahierten Dateien wieder in eine Datenbank importiert werden, so wird das beim Exportieren festgelegte Kennwort benötigt.

**Anmerkung:** Das Export Utility (EXP) unterstützt keine mit TDE verschlüsselten Tabellen und kann solche somit auch nicht exportieren.

### Resümee zur Datenverschlüsselung und Performance

Das Ver- und Entschlüsseln beim Zugriff auf Daten stellt einen im Einzelfall minimalen Aufwand dar, der sich bei der Verarbeitung von Massendaten jedoch akkumulieren kann. Insbesondere beim Zugriff auf verschlüsselte, indizierte Attribute sollte man dies bedenken.

Die Performance beim TDE ist im Vergleich zu den anderen Verfahren im Hinblick auf verschlüsselte, indizierte Attribute besser, da auch die Indizes verschlüsselt abgelegt werden.

Die Verschlüsselung sollte demzufolge nur dann eingesetzt werden, wenn sensible Daten gespeichert werden sollen und eine hohe Sicherheit der Daten erforderlich ist.

Klaus Günther (info@ordix.de).

```
CREATE TABLE mitarbeiter (
  ma_nr          NUMBER,
  name           VARCHAR2 (30) ,
  gehaltsgruppe VARCHAR2 (9) ENCRYPT USING 'AES128' );

ALTER TABLE mitarbeiter
  MODIFY (gehaltsgruppe ENCRYPT USING 'AES128');

ALTER TABLE mitarbeiter MODIFY (gehaltsgruppe DECRYPT);

SELECT * FROM DBA_ENCRYPTED_COLUMNS;

Standard Wallet-Ablageverzeichnis für Master-Schlüssel:
$ORACLE_BASE/admin/$ORACLE_SID/wallet

Ändern der Wallet-Ablage auf dem Datenbankserver:
$ORACLE_HOME/network/admin/sqlnet.ora

ALTER SYSTEM SET ENCRYPTION KEY AUTHENTICATED BY
"geheim";

ALTER TABLE mitarbeiter REKEY USING 'AES256';

expdp kg/kg ENCRYPTION_PASSWORD=geheim
TABLES=mitarbeiter

impdp kg/kg ENCRYPTION_PASSWORD=geheim
TABLES=mitarbeiter
```

Abb. 2: Sammlung beispielhafter Listings und Ablageverzeichnisse.

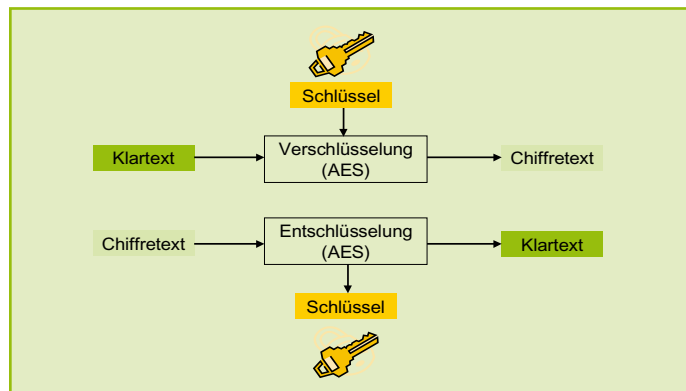


Abb. 3: Vorgehensweise bei der symmetrischen Verschlüsselung.

Glossar	
<b>TDE</b>	Transparent Data Encryption (deutsch: Transparente Datenverschlüsselung)
<b>Package</b>	Stellt eine Sammlung logisch zusammengehöriger Funktionalitäten (Prozeduren, Funktionen, Typdeklarationen, ... ) dar.
<b>Export Utility (EXP)</b>	Werkzeug zum Extrahieren von Strukturen und Daten aus einer Oracle Datenbank.
<b>Data Pump</b>	Werkzeug zum Extrahieren und Importieren von Strukturen und Daten einer Oracle Datenbank ab der Version Oracle Database 10g (siehe ORDIX News 2/2004).
<b>Wallet</b>	Aufbewahrungsort der Zertifikate.
<b>OWM</b>	Oracle Wallet Manager, Tool zur Erzeugung und Bearbeitung von Wallets.



# JasperReports – Java Reporting Engine

Eine sehr häufige Anforderung in Entwicklungsprojekten ist die Erstellung von Reports in verschiedene Dateiformate. Eine Java Reporting Engine, die genau für diesen Sachverhalt entwickelt wurde, ist das Open Source Produkt JasperReports [1]. Wir stellen diese Engine und einige ihrer Funktionalitäten vor.



Dieser Artikel richtet sich an Java-Entwickler, die auf der Suche nach einer flexiblen Reporting Engine sind.

## Designvorlage

In der Designvorlage (\*.jrxml) wird das Layout des Reports definiert und dient als Basis für den späteren Report. Diese wird als einfache XML-Datei realisiert und ist dementsprechend einfach zu handhaben. Die Vorlage ist allgemein in sieben Bereiche strukturiert und setzt sich aus den folgenden Elementen zusammen:

- <title>
- <pageHeader>
- <columnHeader>
- <detail>
- <columnFooter>
- <pageFooter>
- <summary>

Eine fertige Vorlage wird, bevor sie verwendet werden kann, in ein so genanntes JasperDesign (\*.jasper) kompiliert. Dabei wird die XML-Struktur in Java-Code gewandelt und gleichzeitig serialisiert. Durch die Methode *compileReportToFile(String)* der Klasse *JasperCompileManager* wird dieser Prozess durchgeführt.

Um die Performance zu steigern, ist es sinnvoll, den Report vor dem Einsatz zu kompilieren und nur die JasperDesign-Datei zu verwenden, da das Kompilieren doch einige Zeit beansprucht.

## Füllen und Exportieren des Reports

Die fertige Jasper-Datei wird durch eine Methode *fillReport(...)* der Klasse *JasperFillManager* mit den gewünschten Daten ergänzt.

Das daraus resultierende *JasperPrint*-Objekt wird nun durch die Klasse *JasperExportManager* in die Dateiformate XLS, RTF, CSV, XML, HTML oder PDF konvertiert oder direkt mit Hilfe der Klasse *JasperPrintManager* auf dem Drucker oder Bildschirm ausgegeben.

Die Funktionsweise von JasperReports wird in der Abbildung 1 schematisch aufgezeigt.

## Datenquellen

Für die Erstellung der Reports besteht die Möglichkeit, diverse Datenquellen heranzuziehen. Dabei werden folgende Datenquellen standardmäßig unterstützt:

- Empty Data Source
- Database Data Source
- XML Data Source
- Custom Data Source

Die *Empty Data Source* wird in der Regel nur für kurze Tests der Reports genutzt. Dabei wird

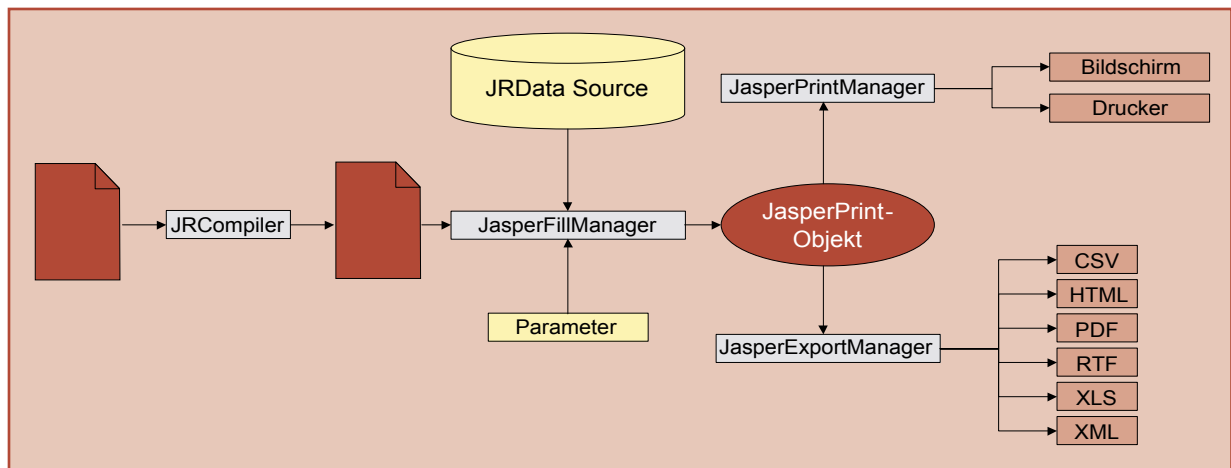


Abb. 1: Schematische Funktionsweise von JasperReports.

der Report mit mehreren Zeilen gefüllt, die jeweils nur `null`-Werte besitzen.

Neben der *Database Data Source*, die normale JDBC-Statements verarbeitet, stellt JasperReports auch eine *XML Data Source* zur Verfügung, welche ausschließlich XML-Dateien als Datenquelle akzeptiert.

Um weitere Quellen anzubinden, die JasperReports nicht direkt unterstützt, wird die *Custom Data Source* angeboten. Hierfür ist es nötig, die passende Java-Schnittstelle selbst zu implementieren und dem Report zu übergeben. Dabei sind gerade die Interfaces `JRBeanArrayDataSource` und `JRBeanCollectionDataSource` von großer Bedeutung.

Hierbei ruft JasperReports immer die zugehörige *get*-Methode eines deklarierten Feldes auf. Die Anwendung ruft beispielsweise bei dem Feld mit der Bezeichnung „projectno“ die Methode *getProjectno()* der zugehörigen Java-Bean auf.

## Felder

Um die gewünschten Daten in den Report zu transferieren, werden Felder benötigt, welche die jeweiligen Daten aufnehmen. In der Regel definiert der Entwickler pro Datenfeld ein Feld in der XML-Vorlage. Dieses Feld kann anschließend die Daten in dem gewünschten Format aufnehmen und in den Report schreiben.

JasperReports unterstützt hierfür einige Java-Datentypen. Neben den Wrapper-Klassen für primitive Datentypen (`java.lang.Boolean`, `java.lang.Integer`, etc.) können hier noch Objekte der folgenden Typen benutzt werden:

- `java.util.Date`
- `java.sql.Timestamp`
- `java.io.InputStream`
- `java.awt.Image`
- `java.math.BigDecimal`

Falls als Quelle ein anderer Datentyp vorliegt, wird dieser vom Typ `java.lang.Object` über-

geben und anschließend in eines der oben genannten Formate gecastet. Neben diesen dynamischen Feldern gibt es auch die Möglichkeit, statische Felder zu definieren. Diese enthalten einen einfachen Text und dienen somit z. B. als Spaltenüberschrift.

## Parameter

Weiterhin besteht die Möglichkeit, Parameter – genau wie Felder – innerhalb der XML-Vorlage zu definieren. Diese werden aber nicht während des Füllprozesses von der Datenquelle geholt, sondern schon beim Aufruf der Methode *fillReport(...)* der Klasse *JasperFillManager* mit übergeben. Diese wurden zuvor in der Java-Anwendung als Collection vom Typ `HashMap` angelegt.

## Variablen

JasperReports verarbeiten zudem auch Java-Variablen. Standardmäßig stehen folgende Variablen zur Verfügung:

<code>PAGE_NUMBER</code>	Die aktuelle Seitenzahl
<code>COLUMN_NUMBER</code>	Die aktuelle Spaltennummer
<code>REPORT_COUNT</code>	Anzahl der verarbeiteten Einträge
<code>PAGE_COUNT</code>	Anzahl der Seiten
<code>COLUMN_COUNT</code>	Anzahl der Spalten
<code>GroupName_COUNT</code>	Anzahl der Datensätze innerhalb einer spezifischen Gruppe

Auch eigene Variablen, die auch während des Füllprozesses verwendet werden sollen, können definiert werden.

## Expressions

Um dynamisch in den Füllprozess eingreifen zu können, bietet JasperReports so genannte Expressions an. Dafür stehen dem Entwickler unter anderem folgende Elemente zur Verfügung:

- `<textFieldExpression>`
- `<printWhenExpression>`
- `<variableExpression>`
- `<initialValueExpression>`

Expressions erlauben den Zugriff auf Parameter, Felder und Variablen. Eine einfache Ausgabe dieser Typen innerhalb einer Expression hat folgende Syntax:

```
$P{ParameterName}
$F{FeldName}
$V{VariablenName}
```

```
<pageFooter>
  <band height="15">
    <textField>
      <reportElement x="0" y="20" width="170" height="15"/>
      <textElement textAlignment="Left"/>
      <textFieldExpression class="java.lang.String">
        new java.text.SimpleDateFormat("dd.MM.yyy HH:mm").format(new java.util.Date())
      </textFieldExpression>
    </textField>
  </band>
</pageFooter>
```

Abb. 2: Beispiel-Ausdruck innerhalb einer XML-Vorlage.

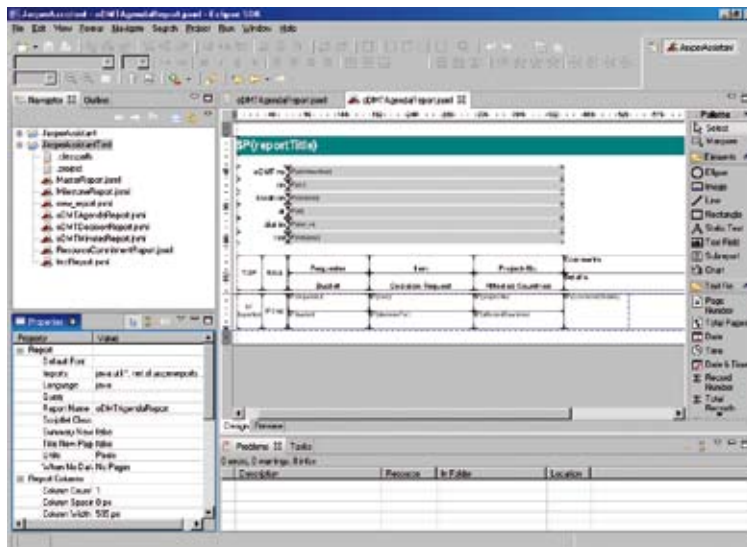


Abb. 3: Screenshot von Eclipse mit JasperAssistant.

- `afterColumnInit()`
- `afterDetailEval()`
- `afterGroupInit()`
- `afterPageInit()`
- `afterReportInit()`
- `beforeColumnInit()`

In der Vorlage muss darüber hinaus der Name und das Package des Scriptlets mit Hilfe des Attributs `scriptletClass` im Element `<jasperReport>` angegeben werden.

### Diagramme und Grafiken

Die Möglichkeit, eigene Diagramme oder andere Grafiken in den Report einzubinden, ist eine weitere, nützliche Funktion von JasperReports. Der Anwender hat dabei die Auswahl zwischen den gängigsten Diagrammtypen, wie Säulen-, Balken-, Kreisdiagramme etc.

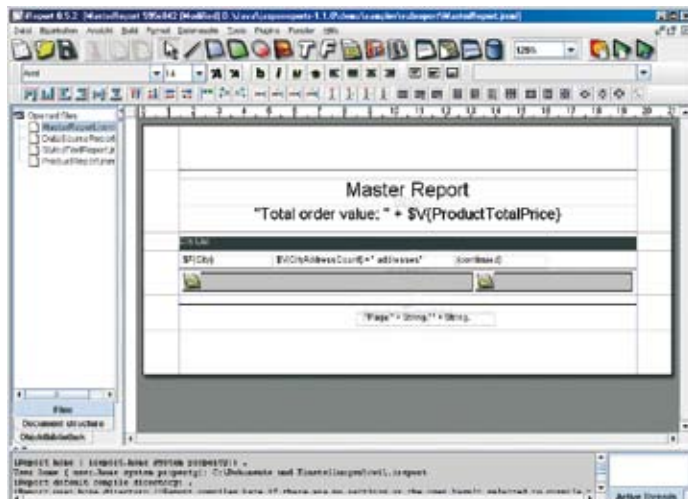


Abb. 4: Screenshot von iReport.

Darüber hinaus gibt es die Möglichkeit, innerhalb der Expressions einfachen Java Code zu implementieren um z. B. die Ausgabe in einem bestimmten Format zu realisieren oder auch um auf `null`-Werte zu überprüfen. So lässt sich z. B. mit dem Ausdruck in der Abbildung 2 ganz einfach das aktuelle Datum mit Uhrzeit in der Fußzeile eines Reports erzeugen.

### Scriptlets

Zwar ermöglichen Variablen und Expressions es, direkt in den Prozess der Reporterstellung einzugreifen, aber komplexe Logik kann man mit ihnen nicht abbilden.

Genau an diesem Punkt setzen Scriptlets an. Die jeweiligen Methoden der Scriptlet-Klasse werden zur gegebenen Zeit, während des Füllprozesses des Reports, aufgerufen.

Die Scriptlet-Klasse erbt dabei alle Methoden und Attribute von der Klasse `JRDefaultScriptlet`. Innerhalb der neuen Scriptlet-Klasse stehen anschließend u. a. folgende Methoden zur Verfügung:

Auch Grafiken und andere Bilder lassen sich einfach in einen Report einarbeiten. Dabei ist es wichtig, darauf zu achten, dass die Skalierung der Bilder oder Grafiken nicht innerhalb des Reports geschieht. Denn bei dieser findet keine neue Berechnung statt, sondern nur die reine Größenanpassung. Daher ist es von Vorteil, wenn die Grafiken bereits in der nativen Auflösung vorliegen, um den Füllprozess zu beschleunigen und die Dateigröße des späteren Reports nicht unnötig in die Höhe zu treiben.

### Gruppen

Die Gruppierung von Datensätzen ist eine weitere Möglichkeit, den Report zu organisieren. Dabei wird die XML-Vorlage um ein Element `<group>` erweitert. Zu einer Gruppe gehören dann `<groupExpression>`, `<groupHeader>` und `<groupFooter>`. Der Gruppenwechsel wird über die `<groupExpression>` gesteuert. Ändert sich der Ausdruckswert (z. B. der Wert einer Variablen oder eines Feldes) während der Iteration der Datensätze, werden die Elemente `<groupFooter>` und `<groupHeader>` erneut aufgerufen und durchlaufen, bis eine weitere Änderung des Wertes erfolgt.

### Subreports

Ein weiterer Vorteil von JasperReports ist die Unterstützung von Subreports. Ein Subreport ist in erster Linie ein „normaler“ Report, für den auch eine eigene Designvorlage angelegt wird. Innerhalb der XML-Vorlage des Hauptreports referenziert man auf den Subreport durch die Angabe der Datenfelder, welche in Verbindung zueinander stehen. Auch ein rekursiver Subre-

port, der auf sich selbst verweist, kann durchgeführt werden. Somit lassen sich auch komplexe Datenstrukturen innerhalb eines Reports abbilden.

### Editoren für die Vorlagenerstellung

Um die Erstellung des Reports zu vereinfachen, bieten einige Dritthersteller freie Editoren an, mit denen die XML-Vorlagen komfortabel erstellt werden können.

#### JasperAssistant

Bei JasperAssistant [2] handelt es sich um ein Eclipse-Plugin. Um dieses in Eclipse zum Laufen zu bringen, wird zusätzlich das „Graphical Editing Framework“ benötigt, welches auf der Homepage des Eclipse-Projektes [3] zu finden ist. Der Vorteil hierbei ist natürlich die direkte Implementierung in der Entwicklungsumgebung, wodurch keine weiteren Tools notwendig werden (siehe Abbildung 3).

JasperAssistant stellt an sich selbst den Anspruch, dass die Anwendung alle Funktionalitäten der letzten JasperReports Version unterstützt. Das ist aber in der Realität nur bedingt der Fall, da die aktuellste Version häufig mit einer ein bis zwei Generationen älteren Version von JasperReports arbeitet.

#### iReport

Das Problem der Abhängigkeit von der Entwicklungsumgebung umgeht der in Java geschriebene Editor iReport [4]. Hierbei handelt es sich um eine Standalone-Anwendung, die somit unabhängig einsetzbar ist (siehe Abbildung 4).

Der iReport-Editor zeichnet sich im Gegensatz zum JasperAssistant durch eine deutlich benutzerfreundlichere Anwendungsumgebung aus. Hier wird einem nicht so versierten Benutzer die Arbeit erheblich erleichtert. Nur werden leider noch nicht alle Funktionalitäten der neuesten Version wie z. B. Crosstabs unterstützt.

### Schwachstellen

Die Exportfunktionalitäten von JasperReports leisten einwandfreie Arbeit. Nur der XLS-Exporter lässt noch einige Wünsche offen. Hierbei ist eine Schwäche, dass die Seitenränder eines Reports als eigene Spalten bzw. Zeilen wiedergegeben werden und nicht die eigentlichen Excel-Seitenränder angepasst werden.

Des Weiteren muss gerade bei der Erstellung der XML-Vorlage penibel darauf geachtet werden, dass sich die einzelnen Felder nicht überschneiden. Tritt der Fall dennoch ein, wird das

### Links

- ▶ [1] <http://jasperreports.sourceforge.net>
- ▶ [2] <http://www.jasperassistant.com>
- ▶ [3] <http://www.eclipse.org/gef/>
- ▶ [4] <http://ireport.sourceforge.net>
- ▶ [5] <http://www.jaspersoft.com>

### Glossar

<b>Java Data Base Connectivity (JDBC)</b>	Standard für den Zugriff auf relationale Datenbanken mit Java.
<b>Java-Bean</b>	Java-Klasse mit genau festgelegten Konventionen für die Schnittstellen.
<b>Application Programming Interface (API)</b>	Sammlung von Klassen, welche eine Schnittstelle zu einer Hardware oder Anwendung definieren.
<b>Framework</b>	Sammlung von Tools oder Kombination von Klassenbibliotheken.
<b>Collection</b>	Datenstrukturen in Java. Objekte, die mehrere Elemente zu einer Einheit gruppieren. Collections sind in Listen (List), Mengen (Set) und Abbildungen (Map) kategorisiert.

überlappende Feld konsequent ignoriert und nicht in den Report eingebaut.

Das ist vor allem bei doppelzeiligen Reports der Fall. Hierbei kann es passieren, dass die obere Zeile mehr Daten enthält als vorher angenommen und sich dadurch vergrößert. Dabei überschneiden sich sehr häufig die beiden Zeilen, wodurch JasperReports einige Felder im fertigen Report nicht mehr mit Daten füllt.

Daher sollte man doppelzeilige Reports möglichst vermeiden oder die Vergrößerung der Felder durch das Attribut `isStretchWithOverflow` innerhalb des Elements `<textField>` unterbinden, was aber den Nachteil nach sich zieht, dass JasperReports überlange Daten einfach abschneidet.

Ein weiterer Nachteil des XLS- und des RTF-Exporters ist die fehlende Unterstützung von Hyperlinks. Selbst durch Angabe der benötigten `HyperlinkExpression` verweigern die Hyperlinks ihren Dienst und werden nur als „normaler“ Text ausgegeben. Erst durch den Einsatz eines neuen XLS-Exporters, der aber **nur** als Sample ab der Version 1.1.0 mitgeliefert wird, lässt sich diese Funktionalität realisieren.

### Fazit

JasperReports ist eine dynamische und sehr flexible Reporting Engine. Die sehr gute Exportfunktion in diverse Dateiformate arbeitet einwandfrei und mit wenigen Handgriffen. Nicht zuletzt durch den Einsatz von Scriptlets ist JasperReports eine mächtige Reporting Engine, die kaum Wünsche offen lässt. Der sehr erfolgreiche Einsatz in einigen unserer langjährigen Java-Entwicklungsprojekten zeigt, dass JasperReports auch in der Praxis eine ausgezeichnete Wahl für Report- und Druckfunktionalität in Java ist.

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



# Oracle HTMLDB – Unendliche Weiten

HTMLDB ist neu seit Oracle 10g und dient als Oberfläche, um Web-Applikationen zu erstellen. Dieser Artikel zeigt die Installation und Konfiguration von HTMLDB, so dass Sie selbst mit Oracles jüngstem Produkt auf Forschungsreise gehen können.

## Scotty an Brücke ...



Dieser Artikel richtet sich an Oracle Datenbankadministratoren, die Webapplikationen mit HTMLDB erstellen wollen.

Abgesehen davon, dass die Zukunft hoffentlich nicht so aussehen wird, hat sicherlich schon jeder von Ihnen einen ähnlichen Dialog führen müssen. Das Problem bei Dokumenten, die meist für eine ganze Abteilung gedacht sind, wie z. B. Urlaubslisten, Stundenerfassung, Projektberichte, ist, dass sie oft nur zentral unter einer Windows-Freigabe zu erreichen sind.

Die Nachteile sind schnell genannt: Die Pflege der Daten ist software- und zumeist auch plattformabhängig. Oft existieren viele lokale aber jeweils unterschiedliche Kopien, Versionsänderungen gehen verloren. Solche Lösungen sind nicht teamfähig, unsicher und erfordern ein hohes Maß an Disziplin.

Genau für diesen Anwendungsfall hat Oracle mit Version 10 des RDBMS ein neues Produkt herausgebracht: ORACLE HTMLDB. Ginge es nach Oracle, ist es das Ende der vielen kleinen EXCEL Spreadsheets und ACCESS Anwendungen, die innerhalb eines Unternehmens kursieren. Nur, was genau steckt eigentlich hinter HTMLDB?

## Das unentdeckte Land – Installation und Konfiguration

HTMLDB wird erst ab der Oracle Datenbank Version 10g ausgeliefert. Die Installation kann aber auch in eine bestehende 9.2 Datenbank erfolgen (siehe Kasten). HTMLDB selbst liegt mittlerweile in der Version 2.0 vor und kann direkt von der Oracle Webseite [1] heruntergeladen werden.

Die Software erfordert einen installierten Apache Server, der mit dem mod\_plsql Modul ausgestattet und konfiguriert ist. Weiterhin wird eine Oracle Instanz benötigt, die ca. 100 MB freien Speicherplatz innerhalb eines Tablespaces aufweisen muss.

Hier wird das Repository von HTMLDB hineingeladen, das die eigentliche Web-Applikation beinhaltet. Die Installation und Konfigu-

ration wird bequem mittels des Oracle Universal Installers, der sich auf der Oracle 10g Companion CD befindet, durchgeführt.

Dabei wird der Apache Webserver in ein neues „ORACLE\_HOME“ Verzeichnis installiert. Das Repository der HTMLDB Anwendung kann in eine bereits vorhandene oder neu zu erstellende Oracle Datenbank geladen werden. Gleichzeitig wird die Konfiguration des Database Access Descriptors (DAD) durchgeführt. Dazu später mehr.

Diese Schritte können notfalls auch manuell vorgenommen werden. Das Repository kann mit Hilfe des mitgelieferten Scripts ins.sql geladen werden. Es befindet sich im Basisverzeichnis der HTMLDB-Installation. Dies kann in einem beliebigen Schema passieren.

### Treffen der Generationen – HTMLDB und Oracle 9i

Wer zur Zeit noch Oracle 9i einsetzt, kann sich trotzdem ein Bild von HTMLDB machen, denn wie schon zu Beginn erwähnt, ist es auch möglich, eine 9i Instanz als Repository zu nutzen. Installieren Sie hierfür den Webserver der 9er Version und laden HTMLDB von der Oracle Internet Seite [1] herunter.

Die Konfiguration des Oracle 9i Webservers für HTMLDB ist in einem Mini Howto [3] nachzulesen. Anschließend führen Sie die Installation wie im Abschnitt „Das unentdeckte Land“ beschrieben durch.

Während der Ausführung werden bestimmte Daten vom Administrator erfragt. Dies sind im Einzelnen:

```
# =====
# mod_plsql configuration file
# =====
# 1. Please refer to plsql.README for a description of this file
# 2. Parameters in this file have to be configured manually
# =====
# Configure Oracle HTTP Server to load mod_plsql
LoadModule plsql_module /oracle/ora10gcomp/Apache/modplsql/bin/modplsql.so
# Load in mod_plsql directives only if mod_plsql is loaded
<IfModule mod_plsql.c>
[ ... ]
```

Abb. 1: Die Datei plsql.conf mit eingebundenem mod\_plsql Modul.

```
<Location /pls/htmlldb>
SetHandler pls_handler
Order deny,allow
Allow from all AllowOverride None
PlsqlDatabaseUsername HTMLDB_PUBLIC_USER
PlsqlDatabasePassword geheim123
PlsqlDatabaseConnectString htmlldb_server:1521:ora01 ServiceNameFormat
PlsqlDefaultPage htmlldb
PlsqlDocumentTablename wwv_flow_file_objects$
PlsqlDocumentPath docs
PlsqlDocumentProcedure wwv_flow_file_manager.process_download
PlsqlAuthenticationMode Basic
PlsqlNLSLanguage GERMAN_GERMANY.WE8MSWIN1252
</Location>
```

Abb. 2: Eine Beispielkonfiguration der dads.conf.

- Verzeichnis und Name des Logfiles, welches geschrieben wird.
- Der TNS-Alias für die Datenbank, in der das Repository geladen wird.
- Das Passwort des Benutzers SYS.
- Das Initial Passwort für den HTMLDB-Administrator (admin).
- Das Tablespace, in dem später die Metadaten der eigentlichen HTMLDB-Applikationen gespeichert werden.
- Das Default Upload/Download Verzeichnis, das es ermöglicht, Daten aus einer erstellten HTMLDB-Anwendung zu ex- und importieren.
- Der Name des temporären Tablespaces.

Damit sich die Oberfläche von HTMLDB in der eigenen Muttersprache präsentiert, kann anschließend zusätzlich das deutsche Sprachpaket mittels load\_de.sql geladen werden.

In der Konfigurationsdatei des Apache Webservers (apache.conf) muss sichergestellt sein, dass das Modul mod\_plsql angezogen wird (siehe Abbildung 1).

Dies ist für die Generierung der dynamischen Webinhalte aus dem Repository verantwortlich und wird indirekt über die Dateien oracle\_apache.conf und plsql.conf inkludiert.

### dads.conf – Database Access Descriptor

Für die Konfiguration des Database Access Descriptors wird die Datei \$ORACLE\_HTTP\_SERVER\_HOME/Apache/modplsql/conf/dads.conf (ehemals marvel.conf) herangezogen. Mit Hilfe eines DADS wird im Prinzip ein Mapping zwischen HTTP Request und Prozeduraufruf vorgenommen.

In unserem Beispiel in Abbildung 2 werden so alle Aufrufe über die URL http://htmlldb\_server:7777/pls/htmlldb an die Datenbank mit dem Service Namen ora01 weitergeleitet. Dort wird die Prozedur htmlldb aufgerufen, die im Parameter PlsqlDefaultPage hinterlegt ist. Diese beantwortet den Request mit der Generierung einer Startseite. Voraussetzung hierfür ist die Anmeldung an die Instanz, die über den Parameter PlsqlDatabaseConnectString identifiziert wird.

Das Passwort für den Datenbank-Connect sollte mit Hilfe des Perl Scripts dadTool.pl nachträglich verschlüsselt werden.

### opmnctl

Im Anschluss daran kann der Oracle HTTP Server gestartet werden. Das Tool opmnctl zum Starten und Stoppen des Servers befindet sich unter \$ORACLE\_HTTP\_SERVER\_HOME/opmn/bin.

Der Aufruf ist denkbar einfach: Mit opnctl startall wird der Server hochgefahren, mit opnctl stopall wiederum gestoppt. Gleichzeitig wird ein Logfile (\$ORACLE\_HTTP\_SERVER\_HOME/opmn/logs/ApacheServer.log) geschrieben, in das sämtliche Meldungen des Webservers protokolliert werden. Bei Problemen, den Server zu starten, ist dies der Einstiegspunkt für genauere Fehleranalysen.



## Der erste Kontakt

Ist der Webserver erfolgreich gestartet worden, so kann die HTML-DB Oberfläche über folgende URLs erreicht werden:

- [http://<hostname>:7777/pls/htmldb/htmldb\\_admin](http://<hostname>:7777/pls/htmldb/htmldb_admin)
- <http://<hostname>:7777/pls/htmldb/htmldb>

Kommt es hier zu einer Fehlermeldung im Browser, liegt es meistens an einem Konfigurationsproblem im Database Access Descriptor. Hier sollten folgende Schritte nacheinander überprüft werden:

- Kontrollieren Sie die Log Files des Webserver (\$ORACLE\_HTTP\_SERVER\_HOME/Apache/Apache/logs). Insbesondere die Dateien access\_log.<PID> und error\_log.<PID>
- Kann der Webserver generell angesprochen werden: <http://hostname:7777>
- Überprüfen Sie manuell den Verbindungsaufbau zur Datenbank, mit den Angaben aus der dads.conf: `sqlplus htmldb_public_user/geheim123@ora01`

### Links

- ▶ [1] Download Link 2.0: [http://download.oracle.com/otn/nt/oracle10g/htmldb/htmldb\\_2.0.zip](http://download.oracle.com/otn/nt/oracle10g/htmldb/htmldb_2.0.zip)
- ▶ [2] htmldb Webpage Link: <http://www.oracle.com/global/de/community/index.html>
- ▶ [3] Konfiguration Mini Howto: [http://www.oracle.com/technology/products/database/htmldb/howtos/howto\\_use9ir2ohs.html](http://www.oracle.com/technology/products/database/htmldb/howtos/howto_use9ir2ohs.html)

### Glossar

<b>mod_plsql</b>	Ein Apache Modul von Oracle, das zur Generierung von dynamischen HTML Seiten eingesetzt wird.
<b>Database Access Descriptor (dads.conf)</b>	Konfigurationsdatei, die die Schnittstelle zwischen Webserver und Datenbank bildet.
<b>opmnctl</b>	Kommandozeilen-Tool zum Hochfahren des Webserver.
<b>Workspace</b>	Definiert eine Anwendung unter HTMLDB. Jede Anwendung hat ein eigenes Workspace.
<b>Region</b>	Teile einer Anwendung werden als Region bezeichnet, so z. B. ein HTML Formular oder eine Tabelle.
<b>Quota</b>	Begrenzung von Speicherplatz in der Datenbank für einzelne HTMLDB Anwendungen.

HTMLDB teilt sich in zwei Bereiche auf. Der erste Bereich ([http://<hostname>:7777/pls/htmldb/htmldb\\_admin](http://<hostname>:7777/pls/htmldb/htmldb_admin)) dient der reinen Administration von Anwendungen, Benutzern und Workspaces und ist ausschließlich autorisierten Benutzern zugänglich. Der zweite Bereich (<http://<hostname>:7777/pls/htmldb/htmldb>) dient als Einstieg in die angefertigten HTMLDB Applikationen und kann von allen Anwendern erreicht werden.

Selbstverständlich werden Authentifizierungsmechanismen unterstützt. So kann gezielt gesteuert werden, welche Anwendungen für welche Benutzer zugänglich sind. Der Administrator von HTMLDB ist der Benutzer „admin“ mit dem bei der Installation festgelegten Passwort.

## The Beginning

Nun, wie geht es weiter? Zum Erforschen von HTMLDB bedarf es zunächst der Erstellung eines Workspaces mit Hilfe der Administrationskonsole. Jede HTMLDB Anwendung läuft in einem eigenen Workspace. Dieser bekommt einen eindeutigen Namen und kann optional mit einer Quota belegt werden.

Sämtliche Objekte der Anwendung (Tabellen, Prozeduren, Reports) werden in einem zentralen Schema hinterlegt. Hierfür empfiehlt es sich, einen eigenen Benutzer anzulegen. Alternativ kann auch ein bereits vorhandenes Schema benutzt werden.

Ist das Workspace angelegt worden, kann mit der Anwendungsentwicklung begonnen werden. Hierfür meldet man sich zunächst am neu erstellten Workspace (siehe Abbildung 3) an (<http://<hostname>:7777/pls/htmldb/htmldb>). Anschließend stehen sämtliche Werkzeuge von HTMLDB bereit, um die einzelnen Komponenten einer Anwendung, auch Regionen genannt, zu erstellen. Unterstützt wird man dabei von vielen, nützlichen Assistenten, um so schnell zu einem Ergebnis zu kommen.



Abb. 3: Nach erfolgreicher Anmeldung am Workspace kann „geforscht“ werden.

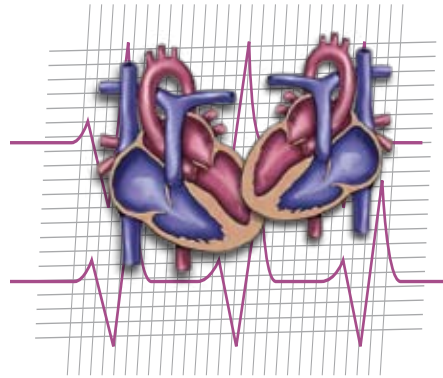
Als Einstiegspunkt sei hier die offizielle HTMLDB-Internetseite [2] empfohlen, von der eine große Menge an Tipps und kleinen Beispielanwendungen heruntergeladen werden kann. Auch ein sehr aktives Forum für alle Themenbereiche von HTMLDB ist hier anzutreffen, in dem fleißig diskutiert werden darf.

Und stoßen Sie bei der Arbeit mit HTMLDB auf ein Problem, das schwierig zu lösen scheint, klappen Sie einfach Ihren Communicator auf, um uns zu kontaktieren.

*Michael Lindermann (info@ordix.de).*

Neue Leistungsmerkmale bei Heartbeat Version 2

# Linux-HA Release 2: Neue Herzschläge



In vorherigen Ausgaben der ORDIX News berichteten wir des öfteren schon von der Open Source Software des Linux High Availability Projekts (Linux-HA), mit der sich ausgesprochen kostengünstige Hochverfügbarkeitslösungen implementieren lassen.

Nach mehr als einem Jahr Entwicklungszeit wurde Ende Juli 2005 die erste, als stabil bezeichnete Version des Releases 2 der Cluster Software veröffentlicht. Wir stellen Ihnen die neuen Möglichkeiten dieses Releases vor.

## Mehr als zwei Knoten

Bisher war ein Linux-HA Cluster auf zwei Knoten beschränkt. In der neuen Version gibt es zumindest theoretisch keine Beschränkung mehr hinsichtlich der Anzahl der beteiligten Systeme. Getestet wurden bisher Cluster mit bis zu 16 Knoten, was im Normalfall jedoch mehr als ausreichend sein sollte.

## Neue Architektur

Die Unterstützung für mehr als zwei Knoten macht weit reichende Veränderungen an der Architektur der Software notwendig. Bisher lief jeweils nur ein zentraler Prozess auf beiden Knoten, welcher für den Versand und Empfang der Heartbeat-Meldungen sowie für den Start und Stopp von Ressourcen zuständig war. Nun besteht die Software aus einer Reihe unterschiedlicher Module, die miteinander kommunizieren.

Der Artikel richtet sich an Systemadministratoren, die Cluster unter anderen Betriebssystemen betreiben, bereits einen Cluster mit Linux-HA im Einsatz haben oder die sich über die neuen Funktionen des Linux-HA Release 2 informieren möchten.



Die wesentlichen Komponenten werden in den nächsten drei Abschnitten vorgestellt.

## Cluster Information Base

Die Cluster Information Base (CIB) stellt die zentrale Informationsstelle des Clusters dar. Sie besteht aus einem XML-Datensatz, der alle Informationen über den aktuellen Status des Clusters und der einzelnen Knoten sowie der vom Cluster verwalteten Ressourcen beinhaltet.

Manuelle Änderungen an der Cluster Information Base werden auf einem dedizierten System durchgeführt, dem „Designated Coordinator“ (DC), und anschließend automatisch auf die übrigen Knoten repliziert.

```
<cib>
  <configuration>
    <crm_config>
      <!-- Basiskonfiguration des Clusters -->
    </crm_config>
    <nodes>
      <!-- Die beteiligten Knoten -->
    </nodes>
    <resources>
      <!-- Vom Cluster verwaltete Ressourcen -->
    </resources>
    <constraints>
      <!-- Abhaengigkeiten von Ressourcen -->
    </constraints>
  </configuration>
  <status>
    <!-- Statusinformationen ueber Knoten und Ressourcen -->
    <!-- Wird NICHT manuell gepflegt, sondern vom CRM verwaltet -->
  </status>
</cib>
```

Abb. 1: Grundgerüst der Cluster Information Base (CIB).



```
<resources>

  <primitive id="Apache_IP" class="ocf" type="IPaddr" provider="heartbeat">
    <instance_attributes>
      <attributes>
        <nvpair name="ip" value="192.168.1.100"/>
      </attributes>
    </instance_attributes>
  </primitive>

  <primitive id="Apache_Process" class="ocf" type="Apache" provider="heartbeat"/>

</resources>
```

Abb. 2: Definition von Ressourcen in der Cluster Information Base.

```
<constraints>

  <!-- Ressource "Apache_IP" soll praeferiert auf dem Knoten "node1" laufen -->
  <rsc_location id="Apache" rsc="Apache_IP">
    <rule id="" score="100">
      <expression attribute="#uname" operation="eq" value="node1"/>
    </rule>
  </rsc_location>

  <!-- Ressource "Apache_Process" muss immer zusammen mit "Apache_IP" laufen -->
  <rsc_colocation id="Apache_IP_Process" from="Apache_IP" to="Apache_Process" score="INFINITY"/>

  <!-- Ressource "Apache_IP" darf niemals auf Knoten "node3" laufen -->
  <rsc_location id="Apache_IP_not_on_node3" rsc="Apache_IP">
    <rule id="" score="-INFINITY">
      <expression attribute="#uname" operation="eq" value="node3"/>
    </rule>
  </rsc_location>
</constraints>
```

Abb. 3: Definition von Constraints in der Cluster Information Base.

Das Grundgerüst der CIB zeigt Abbildung 1.

Die vom Cluster verwalteten Dienste und die dazugehörigen Ressourcen werden ebenfalls in der CIB gepflegt.

Abbildung 2 zeigt ein Beispiel für die Definition zweier Ressourcen – einer IP Adresse sowie einen Apache Webserver Dienst.

Bei der Ressource „Apache\_IP“ wird die IP-Adresse als zusätzliches Attribut innerhalb des Resource-Objekts definiert.

### Cluster Resource Manager

Der Cluster Resource Manager (CRM) ist die Komponente des Clusters, welche den Status der beteiligten Ressourcen verwaltet. Auf jedem Knoten läuft zu diesem Zweck ein Cluster Resource Manager Dämonprozess (crmd), der die Informationen aus der Cluster Information Base verarbeitet bzw. bereitstellt.

Soll beispielsweise ein Dienst von einem Knoten auf einen anderen verschoben werden, wird diese Information über den Cluster Resource Manager an die Knoten verteilt, welche wiederum dafür zuständig sind, die notwendigen Aktionen einzuleiten und den Erfolg oder Misserfolg zu melden.

### Local Resource Manager

Der Local Resource Manager (LRM) sorgt für den eigentlichen Start und Stopp der Dienste auf einem Knoten. Er empfängt Befehle zur Änderung oder Überwachung des Status einer Ressource vom CRM und gibt entsprechende Meldungen zurück.

### Resource Monitoring

Nachdem die Überwachung von laufenden Ressourcen bisher nur durch die Einbindung externer Tools möglich war, unterstützt das Release 2 nun ein integriertes Resource Monitoring.

Zu diesem Zweck werden Skripte zum Start und Stopp einer Ressource nach den Vorgaben des Open Cluster Frameworks (OCF) erstellt, welche eine Monitor Operation vorsehen. Der Local Resource Manager nutzt diese, um in zu definierenden Abständen den Status der Ressource zu erfragen.

**Glossar**

<b>Cluster Information Base (CIB)</b>	Zentrale Informationsstelle des Clusters
<b>Cluster Resource Manager (CRM)</b>	Cluster-Komponente, die den Status der beteiligten Ressourcen verwaltet.
<b>Local Resource Manager (LRM)</b>	Sorgt für den eigentlichen Start und Stopp der Dienste auf einem Knoten.
<b>Linux-HA</b>	Linux High Availability Project

Im Fehlerfall wird je nach Konfiguration zunächst versucht, die Ressource lokal neu zu starten oder direkt ein Failover eingeleitet.

**Abhängigkeiten zwischen Ressourcen**

Mit dem neuen Release gibt es nun sehr gute Möglichkeiten, um Abhängigkeiten für Ressourcen zu definieren. So ist es z. B. möglich, zu bestimmen, auf welchen Knoten ein Dienst bevorzugt, nur im Fehlerfall oder auch auf gar keinen Fall laufen sollte.

Zu diesem Zweck werden innerhalb der CIB Einschränkungen („Constraints“) definiert, die wiederum Punkte („Scores“) für bestimmte Bedingungen vergeben. Die Anzahl der Punkte entscheidet darüber, auf welchem Knoten eine Ressource läuft.

Die Schlüsselwörter „INFINITY“ bzw. „-INFINITY“ bestimmen dabei, dass eine Bedingung „auf jeden Fall“ beziehungsweise „auf keinen Fall“ erfüllt sein muss.

Abbildung 3 zeigt einen Auszug aus der cib.xml, welcher ein typisches Constraint-Objekt für einen Webserver-Dienst beinhaltet.

Mit solchen Constraints lassen sich nahezu beliebige Bedingungen konstruieren. So ist es z. B. möglich, dafür zu sorgen, dass ein Dienst immer auf einem Knoten läuft, der beispielsweise

- eine bestimmte Hauptspeichergröße zur Verfügung hat,
- Anschluss an ein externes Storage-Gerät hat,
- eine bestimmte Kernel-Version hat, u.s.w.

**Administration**

Die Änderungen an der Architektur haben auch Auswirkungen auf die Administration des Clusters.

**Die wichtigsten Neuerungen des Linux-HA Release 2**

- Integriertes Resource Monitoring
- Unterstützung für mehr als 2 Knoten
- XML-basierte Konfiguration
- Komplett überarbeitete Architektur
- Möglichkeit zur Definition komplexerer Abhängigkeiten für Ressourcen

Standardparameter, wie z. B. die Konfiguration der Heartbeat-Leitungen oder das Logging, werden nach wie vor in der bekannten Datei „ha.cf“ gepflegt.

Die vom Cluster verwalteten Ressourcen werden nun allerdings in der CIB definiert, welche damit die „hresources“ Datei abgelöst hat. In diesem Zusammenhang sind eine Reihe von Kommandozeilenwerkzeugen entwickelt worden, die an dieser Stelle nicht im Einzelnen angesprochen werden können.

**Unser Eindruck**

In unseren ersten Tests machte das neue Release einen positiven Eindruck. Die Software läuft stabil und die neuen Features machen den Cluster um ein Vielfaches leistungsfähiger.

Die neue Architektur sorgt allerdings auch für eine deutlich höhere Komplexität, was sich zwangsweise auch auf die Administration auswirkt.

Konzeption und Betrieb eines Clusters dieser „neuen Generation“ erfordert zumindest anfangs mehr Aufwand als dies bei der „alten“ Version der Fall war.

**Ausblick**

Die Roadmap des Projekts verspricht auch weiterhin interessante Neuerungen. Neben einer Reihe von technischen Verbesserungen lässt vor allem die Ankündigung einer grafischen Administrationsoberfläche hoffen, dass sich die Software zukünftig noch größerer Beliebtheit erfreuen wird.

**Fazit**

Mit dem neuen Release ist die Entwicklergemeinschaft des Linux High Availability Projects vielen Wünschen und Anregungen der Anwender nachgekommen und hat vor allem im Vergleich zu etablierten, kommerziellen Clusterlösungen einiges an Boden gut gemacht.

Die Beschränkung auf zwei Knoten sowie das fehlende, integrierte Resource Monitoring waren bisher die größten Nachteile gegenüber kommerziellen Alternativen. Die potenziellen Einsatzgebiete sind damit zahlreicher geworden.

Für den produktiven Einsatz ist es auf Grund fehlender Erfahrungswerte vielleicht noch etwas zu früh, aber in einigen Monaten sollte dem nichts mehr im Wege stehen.

*Christof Amelunxen (info@ordix.de).*



# Seminartermine - herausnehmbare Übersicht -

	Preis in EURO**)	März				April				Mai				Juni			
		KW 10	KW 11	KW 12	KW 13	KW 14	KW 15	KW 16	KW 17	KW 18	KW 19	KW 20	KW 21	KW 22	KW 23	KW 24	
<b>Datenbanken</b>																	
Oracle SQL	1790,00																
Oracle SQL für Experten	1190,00																
Oracle Datenbankprogrammierung mit PL/SQL	1790,00																
Oracle Datenbankadministration Grundlagen	1890,00																
Oracle Datenbankadministration Aufbau	1890,00																
Oracle Backup und Recovery	1890,00																
Oracle Tuning und Monitoring	1890,00																
Oracle Troubleshooting Workshop	1890,00																
Oracle Real Application Cluster RAC	1490,00																
Oracle 10g Neuheiten	1890,00																
Oracle Net	790,00																
Oracle Security Workshop	1190,00																
Oracle Data Guard Workshop	1190,00																
Oracle RMAN Workshop	1490,00																
Informix SQL	1590,00																
Informix Dynamic Server Administration	1790,00																
Informix Tuning und Monitoring	1890,00																
Informix Backup und Recovery mit ON-Bar	1090,00																
IBM DB2 UDB für Unix/Windows SQL Grundlagen	1790,00																
IBM DB2 UDB für Unix/Windows Administration Grundlagen	1890,00																
MySQL Administration	1090,00																
<b>Programmierung</b>																	
Einführung in die objektorientierte Programmierung	1090,00																
Java Programmierung Grundlagen	1590,00																
Java Programmierung Aufbau	1590,00																
Java 5.0 Neuheiten	1090,00																
Perl Programmierung Grundlagen	1590,00																
Perl Programmierung Aufbau	1590,00																
Einführung in XML	1090,00																
XML Programmierung unter Java mit DOM und SAX	790,00																
Oracle und XML	790,00																
PHP Programmierung Grundlagen	1590,00																
E-Commerce Lösungen mit PHP (Workshop)	1090,00																
PHP Programmierung Aufbau	1090,00																
Einführung in Oracle Forms Developer	1790,00																
Einführung in Oracle Reports Developer	1790,00																
Oracle Designer 6i/9i	1790,00																
Shell, Awk und Sed	1590,00																
<b>J2EE</b>																	
J2EE für Entscheider	450,00																
Einführung in J2EE	1090,00																
JSP und Servlet Programmierung	1590,00																
EJB Programmierung	1590,00																
<b>Systemmanagement</b>																	
BMC PATROL/Performance Manager Basics	1890,00																
BMC PATROL/Performance Manager Customizing and Development	1490,00																
BMC PATROL/Performance Manager Advanced	1890,00																
PATROL for Oracle	790,00																
<b>Web- und Applikations-Server</b>																	
Apache Web-Server Installation und Administration	1090,00																
Tomcat Konfiguration und Administration	1090,00																
WebSphere Application Server Installation und Administration	1290,00																
Administration und Konfiguration für JBoss	1090,00																
<b>Betriebssysteme</b>																	
Unix/Linux Grundlagen für Einsteiger	1590,00																
Linux Systemadministration	1590,00																
Linux Hochverfügbarkeits-Cluster	1190,00																
Solaris Systemadministration Grundlagen	1890,00																
Solaris Systemadministration Aufbau	1890,00																
Solaris 10 Neuheiten	1890,00																
Solaris für Unix Umsteiger	1890,00																
Samba Administration und Konfiguration	1090,00																
<b>Netzwerke</b>																	
Linux Netzwerkadministration	1590,00																
<b>Security</b>																	
Unix/Linux Security im Internet	1590,00																
<b>Projektmanagement</b>																	
IT-Projektmanagement	1890,00																
Grundlagen des IT-Controlling	1190,00																

- Wiesbaden
- Saarbrücken
- Lippstadt

\*) Preise pro Seminar pro Teilnehmer in Euro. Alle Preise gelten zzgl. ges. MwSt.  
 \*\*) Inhousepreise auf Anfrage.

Einige der hier aufgeführten Bezeichnungen sind eingetragene Warenzeichen ihrer jeweiligen Inhaber. Irrtümer vorbehalten.

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

Juni		Juli				August					September				Preis in EURO(**)		
KW 25	KW 26	KW 27	KW 28	KW 29	KW 30	KW 31	KW 32	KW 33	KW 34	KW 35	KW 36	KW 37	KW 38	KW 39			
																1790,00	<b>Datenbanken</b>
																1190,00	Oracle SQL
																1790,00	Oracle SQL für Experten
																1790,00	Oracle Datenbankprogrammierung mit PL/SQL
																1890,00	Oracle Datenbankadministration Grundlagen
																1890,00	Oracle Datenbankadministration Aufbau
																1890,00	Oracle Backup und Recovery
																1890,00	Oracle Tuning und Monitoring
																1890,00	Oracle Troubleshooting Workshop
																1490,00	Oracle Real Application Cluster RAC
																1890,00	Oracle 10g Neuheiten
																790,00	Oracle Net
																1190,00	Oracle Security Workshop
																1190,00	Oracle Data Guard Workshop
																1490,00	Oracle RMAN Workshop
																1590,00	Informix SQL
																1790,00	Informix Dynamic Server Administration
																1890,00	Informix Tuning und Monitoring
																1090,00	Informix Backup und Recovery mit ON-Bar
																1790,00	IBM DB2 UDB für Unix/Windows SQL Grundlagen
																1890,00	IBM DB2 UDB für Unix/Windows Administration Grundlagen
																1090,00	MySQL Administration
																1090,00	<b>Programmierung</b>
																1090,00	Einführung in die objektorientierte Programmierung
																1590,00	Java Programmierung Grundlagen
																1590,00	Java Programmierung Aufbau
																1090,00	Java 5.0 Neuheiten
																1590,00	Perl Programmierung Grundlagen
																1590,00	Perl Programmierung Aufbau
																1090,00	Einführung in XML
																790,00	XML Programmierung unter Java mit DOM und SAX
																790,00	Oracle und XML
																1590,00	PHP Programmierung Grundlagen
																1090,00	E-Commerce Lösungen mit PHP (Workshop)
																1090,00	PHP Programmierung Aufbau
																1790,00	Einführung in Oracle Forms Developer
																1790,00	Einführung in Oracle Reports Developer
																1790,00	Oracle Designer 6i/9i
																1590,00	Shell, Awk und Sed
																450,00	<b>J2EE</b>
																1090,00	J2EE für Entscheider
																1090,00	Einführung in J2EE
																1590,00	JSP und Servlet Programmierung
																1590,00	EJB Programmierung
																1890,00	<b>Systemmanagement</b>
																1490,00	BMC PATROL/Performance Manager Basics
																1490,00	BMC PATROL/Performance Manager Customizing and Development
																1890,00	BMC PATROL/Performance Manager Advanced
																790,00	PATROL for Oracle
																1090,00	<b>Web- und Applikations-Server</b>
																1090,00	Apache Web-Server Installation und Administration
																1090,00	Tomcat Konfiguration und Administration
																1290,00	WebSphere Application Server Installation und Administration
																1090,00	Administration und Konfiguration für JBoss
																1590,00	<b>Betriebssysteme</b>
																1590,00	Unix/Linux Grundlagen für Einsteiger
																1590,00	Linux Systemadministration
																1190,00	Linux Hochverfügbarkeits-Cluster
																1890,00	Solaris Systemadministration Grundlagen
																1890,00	Solaris Systemadministration Aufbau
																1890,00	Solaris 10 Neuheiten
																1890,00	Solaris für Unix Umsteiger
																1090,00	Samba Administration und Konfiguration
																1590,00	<b>Netzwerke</b>
																1590,00	Linux Netzwerkadministration
																1590,00	<b>Security</b>
																1590,00	Unix/Linux Security im Internet
																1890,00	<b>Projektmanagement</b>
																1890,00	IT-Projektmanagement
																1190,00	Grundlagen des IT-Controlling

**Informationen und Anmeldung:**

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

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

**zentrales Fax:** 0180 1 ORDIX 0  
**bzw.** 0180 1 67349 0  
**E-Mail:** [training@ordix.de](mailto:training@ordix.de)  
**Online-Anmeldung:** <http://training.ordix.de>



# PL/SQL-Neuerungen mit Oracle 10g Release 2

Mit jedem Release gibt es wieder einige neue Features rund um PL/SQL. Wir stellen Ihnen in diesem Artikel die Neuerungen im Release 2 vor:



Dieser Artikel richtet sich an Oracle Datenbankadministratoren und -entwickler, die in der Version Oracle Database 10g Release 2 PL/SQL-Skripte einsetzen.

- Ausgabe von Daten mittels *DBMS\_OUTPUT*
- Verschlüsselung mittels *WRAP* Package
- Conditional Compilation

## Unlimited im *DBMS\_OUTPUT* Package

In den vorhergehenden Releases war das Maximum der Ausgabe durch *DBMS\_OUTPUT* auf eine Million Bytes beschränkt. Dies war insbesondere bei der Entwicklung oft störend. Der Vorteil von *DBMS\_OUTPUT* im Vergleich zu *UTL\_FILE* ist, dass die Ausgabe auf dem Rechner möglich ist, auf dem der Aufruf stattfindet.

```
ERROR at line 1:
ORA-20000: ORU-10027: buffer overflow, limit of 1000000 bytes
ORA-06512: at "SYS.DBMS_OUTPUT", line 32
ORA-06512: at "SYS.DBMS_OUTPUT", line 97
ORA-06512: at "SYS.DBMS_OUTPUT", line 112
ORA-06512: at line 2
```

Ab Oracle Database 10g Release 2 ist diese Einschränkung aufgehoben und kann nun unbeschränkt („unlimited“) sein.

```
SERVEROUTPUT ON SIZE UNLIMITED FORMAT WORD_WRAPPED
```

```
CREATE OR REPLACE
PROCEDURE test AS
BEGIN
    null;
END;

BEGIN
DBMS_DDL.CREATE_WRAPPED
('CREATE OR REPLACE PROCEDURE test as BEGIN null; END;');
END;
/

SELECT text FROM user_source WHERE name = 'TEST';

TEXT
-----
PROCEDURE test wrapped
a000000
369
abcd
....
```

Abb. 1: Beispiel für eine Prozedur *TEST* im *WRAP*-Format.

Eine weitere Einschränkung war die maximale Größe (255 Bytes) einer angezeigten Zeile durch *DBMS\_OUTPUT*. In Oracle 10gR2 dürfen nun auch die Zeilen unbeschränkt lang sein.

## Das *WRAP* Package

Das Release 2 von Oracle 10g liefert ein Package, das benutzt wird, um den Code in einem Wrapped-Format zu erstellen. Dieses Package ergänzt das Dienstprogramm „WRAP Utility“, aber ersetzt es nicht. Dies ist sinnvoll, wenn viele Quelldateien im Wrapped-Format erstellt werden sollen.

Allgemein werden PL/SQL-Programme „ge-wrapped“, um darin enthaltene, sensitive und vertrauliche Informationen von Firmen über Geschäftsprozesse und andere Interna zu schützen. Dies soll eine nicht autorisierte Betrachtung des Quellcodes verhindern. In Abbildung 1 ist ein Beispiel für eine Prozedur *TEST* im *WRAP* Format dargestellt.

Die erste Zeile der Prozedur *TEST* ist die Bestätigung, dass das *WRAP*-Verfahren zur Erstellung genutzt wurde. Wenn das DDL-Statement mit der *DBMS\_METADATA.GET\_DDL()* Funktion ermittelt wird, ist auch nur das Wrapped-Format erkennbar.

Wenn der PL/SQL-Code 32 K übersteigt, muss eine andere Methode verwendet werden. Es wird eine Kollektionsvariable als Eingabeparameter benutzt. Hier kann der mitgelieferte Datentyp *VARCHAR2* aus dem Standard Package *DBMS\_SQL* benutzt werden. Der Kollektionsdatentyp (*TABLE OF VARCHAR2*) jedes einzelnen Elementes der Tabelle kann bis zu 32 K des Textes aufnehmen. Die Variable wird um die benötigte Anzahl an Elementen erweitert. Ein Beispiel finden Sie in Abbildung 2.

Die Variable *v\_input* dient dazu, den Originaltext aufzunehmen. In den Zeilen 4 bis 10 sind die Codezeilen zugewiesen, die im *WRAP*-Format verwendet werden sollen. Damit der entsprechende Code aufgenommen wird, dürfen der Variable bis zu 32-K-große Zeilen und

die benötigte Anzahl an Elementen zugewiesen werden.

Die Zeilen 11 bis 15 zeigen den Aufruf für die Erstellung der Prozedur im *WRAP*-Format. Die Kollektion wird als DDL-Parameter in Zeile 12 übergeben. Im ersten und letzten Element ist jeweils ein Kommentar zur Dokumentation hinterlegt. Dies ist eine nicht PL/SQL konforme Syntax. Damit der gültige PL/SQL Code erkannt wird, ist der Parameter *lb* und der Parameter *ub* anzugeben. Damit ist festgelegt, dass der zu benutzende Quellcode innerhalb der Zeilen 2 bis 6 liegt. Mit diesem Verfahren kann nun jede Prozedur innerhalb eines PL/SQL Codes im *Wrapped*-Format angelegt werden.

### „Conditional Compilation“ in PL/SQL

Ein nützliches und elegantes Feature in der Programmierung ist der Einsatz von Compiler Direktiven. Bereits in der Programmiersprache „C“ wurde dieses Konzept benutzt. Preprozessor Anweisungen können nun während der Kompilierung ausgewertet werden – allerdings nicht zur Laufzeit (siehe Abbildung 3). Der Einsatz ist sinnvoll für

- den Test von Programmeinheiten bei der Entwicklung durch Einschalten eines Tracing-Mechanismus in der Entwicklungs- und Testumgebung und einfaches Ausschalten in der Produktionsumgebung.
- alternative Programmteile, die abhängig vom Betriebssystem und der Datenbankversion aktiviert werden können. Dies ist insbesondere interessant, wenn Features neuer DB-Versionen bereits eingesetzt werden können.

Ab Zeile 5, beginnt die Verwendung der Preprozessor Direktiven, um die *VP\_VAL* Variable auszuwerten. Weil die Preprozessorvariable *VP\_VAL* nicht normales PL/SQL ist, wird \$\$ verwendet. Ebenso wird der Compiler informiert, dass er die Zeilen nur während der Kompilierung verarbeiten soll.

Dazu wurde das \$-Zeichen z. B. für *\$\$if* benutzt. Gibt es keine session-weit gültige Einstellung, kann optional die Klausel **REUSE SETTINGS** benutzt werden, die sicherstellt, dass die gleiche Compiler-Anweisung verwendet wird, wenn die Funktionen später rekompiliert werden (siehe Abbildung 4).

Der Wert von *VP\_VAL* wurde während der Kompilierung automatisch auf **FALSE** gesetzt. Nun wird er aktiv auf **FALSE** gesetzt und dann die Funktion noch einmal ausgeführt (siehe Abbildung 5).

```
CREATE OR REPLACE PROCEDURE proc
AS
v_key VARCHAR2(200);
BEGIN
v_key := 'TEST_PROC';
END;

1 DECLARE
2   v_input      DBMS_SQL.VARCHAR2S;
3 BEGIN
4   v_input(1) := 'Array to hold the procedure proc';
5   v_input(2) := 'create or replace procedure proc as';
6   v_input(3) := '  v_key VARCHAR2(200)';
7   v_input(4) := 'begin';
8   v_input(5) := '  v_key := ''TEST_PROC''';
9   v_input(6) := 'end';
10  v_input(7) := 'the end';
11  sys.dbms_ddl.create_wrapped (
12    ddl => v_input,
13    lb  => 2,
14    ub  => 6
15  );
16 END;
```

Abb. 2: Beispiel für eine Prozedur *PROC* mit Variable im *WRAP*-Format.

```
1 CREATE OR REPLACE FUNCTION func
2   RETURN varchar2
3 AS
4 BEGIN
5   $$IF $$ppval $$THEN
6     return 'VP_VAL was TRUE';
7   $$ELSE
8     return 'VP_VAL was FALSE';
9   $$END
10 END;
```

Abb. 3: Beispiel einer Funktion für die Rückgabe einer Zeichenkette.

```
ALTER SESSION SET plsql_ccflags = 'VP_VAL:TRUE';
ALTER FUNCTION func COMPILE plsql_ccflags = 'VP_VAL:TRUE'
REUSE SETTINGS;

SELECT func FROM dual;

FUNC
-----
VP_VAL was TRUE
```

Abb. 4: Beispiel für die **REUSE SETTING** Klausel.

```
ALTER SESSION SET plsql_ccflags = 'VP_VAL:FALSE';
SELECT func FROM dual;

FUNC
-----
VP_VAL was TRUE
```

Abb. 5: Beispielausgabe für das Ignorieren durch **ALTER SESSION**.

```
ALTER FUNCTION func COMPILE;
SELECT func FROM dual;

FUNC
-----
VP_VAL was FALSE
```

Abb. 6: Beispielausgabe für das manuelle Rekompilieren der Funktion.



## Glossar

<b>DDL</b>	Data Definition Language. Dient zur Erstellung der Datenbankobjekte innerhalb eines Schemas.
<b>Schema</b>	Sammlung der Datenbankobjekte eines Benutzers innerhalb der Datenbank.
<b>PL/SQL</b>	Prozedurale Erweiterung der Abfragesprache SQL (Structured Query Language).
<b>Package</b>	Stellt eine Sammlung logisch zusammengehöriger Funktionalitäten (Prozeduren, Funktionen, Typdeklarationen, ...) dar.

Obwohl der Wert innerhalb der Session auf **FALSE** steht, benutzt ihn die Funktion nicht. Er wird während der Kompilierung gesetzt. Nun wird einmal rekompiliert und dann die Funktion noch einmal

ausgeführt. Während der Kompilierung ist der Wert **FALSE** und wird nun zurückgegeben (siehe Abbildung 6).

Dies kann nur als Einstieg in das Thema verstanden werden. Weitergehende Informationen sind natürlich der Oracle-Dokumentation zu entnehmen oder sprechen Sie uns an.

## Resümee

Die hier beschriebenen Verbesserungen im PL/SQL des neuen Database 10g Release 2 sind nicht sehr umfangreich, aber nützlich und helfen dem Entwickler an vielen Stellen.

*Klaus Günther (info@ordix.de).*

## Aktuell



# Larry Ratlos: Partitionierte Tabellen

Larry Ratlos wird von dem Kollegen Siegfried Quelle aus der Datenbank-Administration „um Hilfe“ gebeten.

Die zentrale Data Warehouse Datenbank platzt aus allen Nähten. Viele Tabellen werden immer größer, die Zugriffe inperformanter, die Wartbarkeit schlechter.

Siegfried ist auf einem Lehrgang gewesen und hat von dort mitgenommen, dass partitionierte Tabellen hier helfen.

So hat er mittlerweile beispielsweise die Tabelle KUNDEN partitioniert. Die Partitionierung erfolgt je Quartal und unterliegt folgender Namenskonvention:

**KUNDEN \_ JAHR \_ QUARTAL**

also z. B. **KUNDEN \_ 2005 \_ 3**  
oder z. B. **KUNDEN \_ 2004 \_ 4**

Larry hat sich in den letzten Monaten viel mit SQL auseinandergesetzt. Den Wissensstand will Siegfried nun testen, indem er ihn um Hilfe bittet.

So sagt Siegfried: „Ich muss alle 3 Monate ein sehr kompliziertes Kommando absetzen, um die nächste Partition zu erstellen. Kannst Du mir das Kommando mit einem einzigen SQL-Statement generieren?“

Ich möchte, dass Du folgendes Statement erstellst:  
**ALTER TABLE KUNDEN ADD PARTITION KUNDEN\_JJJJ\_Q  
VALUES LESS THAN (TO\_DATE('01.MM.JJJJ,'DD.MM.YYYY'));**

Die variablen Teile sind in diesem Beispiel rot markiert.“

Larry macht sich an die Arbeit und holt sich dabei die notwendige Information aus der View USER\_TAB\_PARTITIONS heraus.

Aber er schafft es nicht, das Statement zu generieren.

## Können Sie Larry helfen?

Senden Sie Ihren Lösungsvorschlag **bis zum 21. April 2006 an kniffel@ordix.de**. In der nächsten Ausgabe werden wir die „piffigsten“ 3 Lösungsvorschläge veröffentlichen.

## Lösung Aufgabe aus 4/2005:

In der letzten ORDIX News brauchte A. Zubi Hilfe beim ermitteln der Linux-Distribution eines Rechners. Es haben sich viele Helfer gefunden, die eine Möglichkeit gefunden haben, die Distributionen auszulesen. Die fünf schnellsten waren: Gabriele Herold, Thomas Jansen, Thomas Beerheide, Rainer Grafe und Matthias Brunschen.

Die Helfer haben A. Zubi alle (zum Teil in leicht abgewandelter Form) darauf aufmerksam gemacht, dass die Informationen in /etc/issue oder in den Dateien /etc/\*release zu finden sind.

# Session Tracing

Wer kennt es nicht! Der Buchhalter Müller beschwert sich, dass die „Datenbank“ bei ihm so langsam läuft. Bei seinem Nachbarn Meier würde alles sehr schnell gehen. Sie machen daraufhin Performance-Messungen, stellen aber keine globalen Probleme fest. Also muss mitgeschnitten werden, was Müller genau macht, um detaillierte Aussagen treffen zu können.

## Client/Server-Technologien

Die gute, alte Client/Server-Welt gibt es heute kaum noch. Die Anwender greifen häufig über einen Terminal Server oder einen Application Server auf die Datenbank zu (siehe Abbildung 1).

Der Administrator kann deshalb nicht mehr erkennen, welcher Prozess oder welche Statements auf der Datenbank von welchem Anwender initiiert wurden.

Der Client ist aus Sicht der Datenbank der Application Server oder der Terminal Server und nicht mehr der Anwender. Eine ähnliche Problematik stellt sich bei Verwendung der Multi-Threading Architektur (MTS).

## Was muss der Entwickler im Vorfeld leisten?

Das Package **dbms\_application\_info** steht schon seit Oracle 8 zur Verfügung, wird aber leider von den Entwicklern selten genutzt.

Wir möchten hier darstellen, warum die Verwendung dieser Funktionalität sehr wichtig sein kann und wie man dieses Paket sinnvoll verwendet.

Das Package **dbms\_application\_info** besteht aus 6 Prozeduren, die in Abbildung 2 kurz aufgeführt werden.

Wichtig für das Session Tracing sind die Prozeduren **set\_module** und **set\_action** (siehe Abbildung 3).

Der Entwickler sollte bei jedem Aufruf eines Moduls zunächst **dbms\_application\_info.set\_module** aufrufen. Innerhalb eines Moduls können Teilsteps über **dbms\_application\_info.set\_action** definiert werden (siehe Abbildung 4).

## Was haben wir nun davon?

Grundsätzlich kann der Administrator so relativ schnell erkennen, in welchem Modul sich der Anwender gerade befindet und braucht

Dieser Artikel richtet sich vor allem an erfahrene Oracle-Administratoren, die in der Lage sind, Tracefiles zu erzeugen und zu interpretieren.

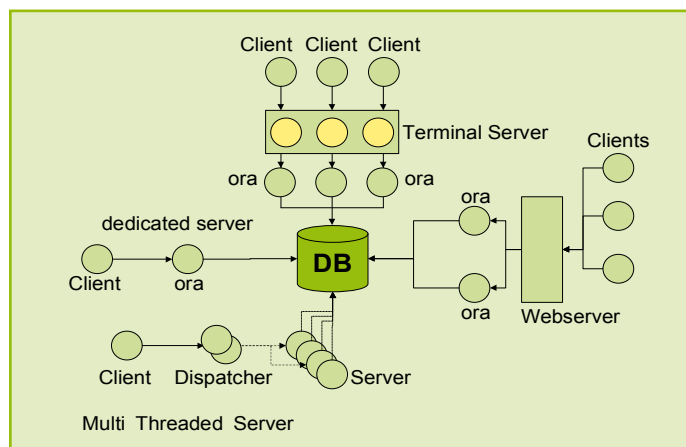


Abb. 1: Schematische Darstellung der Möglichkeiten eines Connects an die Datenbank.

read_client_info	Auslesen der Client-Informationen
read_module	Auslesen des Modulnamens und der Aktion
set_action	Definition einer Aktion im aktuellen Modul
set_client_info	Setzen einer zusätzlichen Client-Information
set_module	Definition eines Modulnamens
set_session_longops	Information in die v\$session_longops View laden

Abb. 2: Die sechs Prozeduren des Packages **dbms\_application\_info**.

```

DBMS_APPLICATION_INFO.SET_MODULE (
    module_name IN VARCHAR2, action_name IN VARCHAR2);
DBMS_APPLICATION_INFO.SET_ACTION (
    action_name IN VARCHAR2);
    
```

Abb. 3: Beispielaufrufe für **dbms\_application\_info**.

```

begin
dbms_application_info.set_module ('REIMERS','STEP1');
end;

select sid, serial#, module, action
from v$session where module is not null;

SID SERIAL# MODULE ACTION
-----
158 232 REIMERS STEP1
    
```

Abb. 4: Info aus v\$session nach Initialisierung über **dbms\_application\_info**.



<code>sql_trace=true</code> in der <code>init.ora</code>	Alle Sessions mittracen
<code>alter session set sql_trace=true;</code>	Die eigene Session mittracen
<code>alter session set events '10046 trace name context forever, level 8';</code>	Die eigene Session mitschneiden, dabei zusätzlich Wait-Events und Bind-Variable mit ausgeben
<code>Exec dbms_system.set_sql_trace_in_session (17,255,true);</code>	Vor Oracle 10g: Eine fremde Session, hier z. B. die Session mit SID=17 und Serial#=255 mittracen
<code>exec dbms_session_trace_enable(17, 255, true, true);</code>	Neu in Oracle 10g: Eine fremde Session, hier z. B. die Session mit SID=17 und Serial#=255 mittracen
<code>exec dbms_monitor.serv_mod_act_trace_disable ('PPS', 'REIMERS', 'STEP1');</code>	Das spezifische Tracing wieder ausschalten

Abb. 5: Möglichkeiten, Session Tracing ein- und auszuschalten.

SID	System Identifier, Information aus <code>v\$session</code> (im Beispiel „17“, Abb. 5)
Serial#	Information aus <code>v\$session</code> (im Beispiel „255“, Abb. 5)
Wait Events	true/false – Tracing mit oder ohne Wait Events (im Beispiel „true“, Abb. 5)
Bind Variable	true/false – Tracing mit oder ohne Bind Variable (im Beispiel „true“, Abb. 5)

Abb. 6: Parameter der Prozedur `session_trace_enable`.

<code>exec dbms_monitor.serv_mod_act_trace_enable ('PPS', 'REIMERS', 'STEP1', true, true);</code>	
Service_Name	Der über die <code>init.ora</code> (Parameter <code>service_names</code> ) definierte und über die <code>tnsnames.ora</code> angesprochene <code>Service_Name</code> (hier im Beispiel PPS)
Modul_Name	Der über <code>dbms_application_info.set_module</code> definierte Modulname (hier im Beispiel REIMERS)
Action_Name	Der über <code>dbms_application_info.set_module</code> oder <code>dbms_application_info.set_action</code> definierte <code>Action_name</code> (hier im Beispiel STEP1)
Wait Events	true/false (hier im Beispiel true)
Bind Variable	true/false (hier im Beispiel true)

Abb. 7: Anwendung und Parameter der Prozedur `serv_mod_act_trace_enable`.

nicht aufwendig in vielen Sourcen nach dem Statement zu suchen.

## Wie habe ich bisher Session Tracing angestoßen?

Das Mitschneiden einer Session konnte schon immer auf verschiedene Weise vollzogen werden (siehe Abbildung 5).

## Das Package `dbms_monitor`

In Oracle 10g gibt es nun das neue Package `dbms_monitor`, über das das Session Tracing neu strukturiert wird. Dieses Package enthält neben vielen anderen Prozeduren folgende für uns hier relevante Funktionalitäten:

- `SESSION_TRACE_ENABLE`
- `SESSION_TRACE_DISABLE`
- `SERV_MOD_ACT_TRACE_ENABLE`
- `SERV_MOD_ACT_TRACE_DISABLE`

## Einfaches Session Tracing mit `dbms_monitor`

Um eine fremde Session mitschneiden, gibt es in Oracle 10g nun `dbms_session_trace_enable`.

Diese Vorgehensweise entspricht dem alten, schon lange nicht mehr dokumentierten Package `dbms_system` (siehe Abbildung 5).

Die Parameter der Prozedur `session_trace_enable` haben die in Abbildung 6 genannte Bedeutung.

Ein Tracing einer fremden Session ist nun syntaktisch einfacher geworden, der Event 10046 muss nicht eingebaut werden.

## Session Tracing mit `dbms_monitor` im Application Server Umfeld

Wie eingangs bereits dargestellt, war das Mitschneiden von Benutzeraktionen im Application Server Umfeld bisher nahezu unmöglich. Hier bringt `dbms_monitor` nun die entscheidende Abhilfe.

Die Parameter der Prozedur `serv_mod_act_trace_enable` haben die in Abbildung 7 genannte Bedeutung.

Somit ist der Administrator in der Lage, definierte Teile einer Gesamtapplikation zu tracen. Es entstehen hier allerdings im Verzeichnis `user_dump_dest` viele einzelne Tracefiles.

## ... und nun die Auswertung mit `trcsess` und `tkprof`

Wie schon erwähnt, entstehen viele einzelne Tracefiles im `udump`-Verzeichnis. Zur Verdichtung dieser Daten liefert Oracle nun ein neues Utility `trcsess` aus. Dabei müssen zwingend ein `Service_Name` und eine Ausgabedatei eingegeben werden (siehe Abbildung 8).

```

trcsess
  [output=output_file_name]
  [session=session_id]
  [clientid=client_id]
  [service=service_name]
  [action=action_name]
  [module=module_name]
  [trace_files]

```

**Beispiel:**

```

trcsess
  output=reimers.txt
  service=PPT *.*

```

Abb. 8: Vorstellung des Utility `trcsess` sowie ein Beispiel für die mindestens zu übergabenden Parameter.

Diese kumulierte Datei dient nun als Eingabe für den allseits bekannten `tkprof`:

```
tkprof reimers.txt reimers.out
```

Zurück zu unserem Einstiegsbeispiel: Herr Müller sollte mitgetracet werden. Wir können jetzt ein Modul oder eine Aktion mitschneiden, nicht aber den Benutzer direkt.

**Fazit**

Der Entwickler muss die Voraussetzung schaffen, dass Session Tracing auch in modernen

**Glossar**

<b>Multi Threading Service (MTS)</b>	Mehrere Clients teilen sich einen Datenbankprozess.
<b>Wait Event</b>	Wartezustand innerhalb der Datenbank.
<b>Bind-Variable</b>	Variabel gehaltener Teil eines SQL-Befehls.
<b>Service_Name</b>	Logischer Name, unter dem der Client die Instanz erreicht.
<b>Tnsnames.ora</b>	Datei, in der der Client den Zugriff über Oracle*Net definiert.
<b>Init.ora</b>	Parameterdatei der Instanz und Datenbank.

Umgebungen noch möglich ist. Die Verwendung von `dbms_application_info` ist in unseren Entwicklungsprojekten zur Pflicht erhoben worden. Das sollten auch Sie zum absoluten Standard erheben!

Da das Tracing nun modulweise angestoßen wird, muss mit dem Start des Tracings nicht mehr bis zum Connect eines Benutzers gewartet werden.

Klaus Reimers ([info@ordix.de](mailto:info@ordix.de)).

**Seminar: Oracle 10g Neuheiten**

Der Teilnehmer lernt in diesem Seminar die Neuheiten von Oracle 10g kennen und diese anzuwenden.

**Voraussetzungen**

Gute Grundlagen in Oracle 9i, das Wissen aus dem Seminar „Oracle Datenbankadministration Grundlagen“ ist zwingend erforderlich.

**Zielgruppe**

Datenbankadministratoren, Softwareentwickler, Systembetreuer.

**Kursgebühr/Teilnehmer:**

1.890,00 Euro zzgl. MwSt.

**Dauer:** 5 Tage**Termine**

27.03.2006 - 31.03.2006 in Wiesbaden  
11.09.2006 - 15.09.2006 in Wiesbaden

**Seminarinhalte**

- Überblick GRID-Computing
- SQL-Erweiterungen (Datatypes, reguläre Ausdrücke, statistische Funktionen)
- SQL\*Plus Erweiterungen
- PL/SQL-Neuerungen (dbms\_wm, utl\_mail, utl\_compress, dbms\_scheduler)
- Backup & Recovery Erweiterungen (Flashback, RMAN, Data Guard, LogMiner)
- Automatic Workload Repository (AWR)
- Advisor Framework (SQL Tuning, SQL Access, PGA, Buffer Cache, Library Cache, Segment, Undo, Redo Logfile Size)
- Tablespace Erweiterungen (SYSAUX, Groups, BigFile)
- DataPump
- Übungen



## MySQL 5 New Features

# MySQL 5 – The Next Generation

Seit Oktober 2005 ist die neue MySQL Version 5 für den produktiven Einsatz freigegeben. Der schwedische Datenbankhersteller MySQL AB feiert den neuen Sprössling als „The most important release ever ...“. Als Ziel wurde klar der Einsatz in geschäftskritischen Bereichen definiert – einem Bereich, in dem sich vor allem die bekanntesten, kommerziellen Anbieter etabliert haben. Ob mit dieser Version wirklich der ganz große Wurf gelungen ist, soll im Folgenden näher untersucht werden.



Der Artikel richtet sich an Datenbankadministratoren und Entwickler, die bereits mit älteren Versionen von MySQL vertraut sind und einen Umstieg planen oder auf wichtige Features warten.

### Ein ernst zu nehmender Gegner?

Bislang fristeten MySQL-Datenbanken in größeren Unternehmen ein Schattendasein. Das Fehlen einiger wichtiger Funktionen (Stored Procedures, Funktionen, Views usw.) machte es schwer, Geschäftslogik in der Datenbank zu hinterlegen.

Mit der neuen Version will MySQL diese Schwächen endlich hinter sich lassen und bemüht sich, den Wechsel auf das eigene Produkt so einfach wie möglich zu gestalten.

### Stored Procedures, Trigger, Views und User Functions

Eine der wichtigsten Neuerungen dürfte wohl die Einführung einiger lange vermisster Datenbankobjekte sein. Mit der neuen Version haben endlich Stored Procedures, Trigger, Views und User Functions Einzug in die Open Source Datenbank gehalten.

Besonders lobenswert ist die Tatsache, dass sich MySQL hierbei an den Standard SQL:2003 hält, der z. B. auch von IBMs DB2 unterstützt wird (siehe Abbildungen 1 und 2). Ebenfalls neu in der Version 5.0 sind updateable Views. Bislang waren nur lesende Zugriffe über Views möglich.

### Darf es ein bisschen mehr sein?

MySQL ist von jeher für seine unterschiedlichen Storage Engines bekannt, die je nach den gestellten Anforderungen hinsichtlich Performance oder Sicherheit ihre Stärken ausspielen können. MyISAM, InnoDB, MERGE und HEAP sind für die meisten MySQL-Nutzer längst alte Bekannte. Mit der neuen Version haben sich zwei neue „Datenverwalter“ dazu gesellt: Die Archive Storage Engine und die Federated Storage Engine.

### Klein aber fein

Die Archive Storage Engine soll große Datenmengen platzsparend verwalten. Gedacht ist sie z. B. für alte Bestandsdaten, die im Zu-

```
create function zinsen
  (betrag decimal(10,2), zinsen int(3), jahre int(2))
returns decimal(15,2)
return (betrag * power(1+(zinsen/100), jahre))
```

Abb. 1: Das Beispiel zeigt eine simple Funktion zur Berechnung von Zinseszinsen.

griff der Anwender bleiben sollen, jedoch nicht mehr verändert werden müssen. Die Daten werden dabei über einen zlib-Algorithmus direkt beim Anlegen (INSERT) komprimiert und platzsparend gespeichert.

Die Storage Engine unterstützt ausschließlich INSERT- und SELECT-Kommandos. Trotz der hohen Kompressionsrate erfolgen die Zugriffe außerordentlich schnell. Im Vergleich zu MyISAM-Tabellen können bis zu 80 Prozent Speicherplatz eingespart werden.

### So nah und doch so fern

Die Federated Storage Engine verwaltet lokal keine Daten. Sie bietet dafür die Möglichkeit, Daten aus entfernten Tabellen beziehungsweise Datenbanken einzubinden. Beim Anlegen einer Tabelle wird definiert, auf welcher Instanz die dazugehörigen Daten zu finden sind (siehe Abbildung 3).

Die Storage Engine ist optimal geeignet für Daten, die in mehreren Applikationen parallel benötigt werden. So können etwa Informationen zu Kunden zentral in einer Instanz gespeichert und dort auch gesichert werden. Alle anderen Datenbanken greifen dann über diesen Tabellentypen zentral auf die Daten zu.

### Ideal Standard

Bis zur Version 5.0 wurden die Metadaten einer MySQL-Datenbank nicht – wie bei anderen Datenbanksystemen üblich – durch Systemtabellen bereitgestellt. Vielmehr gab es eine Reihe von SHOW-Befehlen, über welche die benötigten Informationen abgefragt werden konnten (z. B. „SHOW databases“, um die verfügbaren Schemata/Datenbanken zu ermitteln).

Leider sind diese SHOW-Befehle MySQL-spezifisch und entsprechen keinem Standard. Aus diesem Grund wurde in der Version 5 die Meta-Datenbank „information\_schema“ implementiert.

Diese enthält in Tabellenform alle benötigten Metadaten zu Tabellen, User, Prozeduren etc. und entspricht damit ebenfalls dem vereinbar-

```
DELIMITER /;
CREATE PROCEDURE build_table()
BEGIN
DECLARE i INTEGER;
DECLARE v INTEGER;
SET i = 1;
SET v = 100;

WHILE i <= 125 DO
INSERT into mytable VALUES (i, v);
SET i = i + 1;
SET v = v + 2;
END WHILE;
END
/
DELIMITER ';' /
```

Abb. 2: Das Beispiel zeigt eine Prozedur, welche die Tabelle „mytable“ mit 125 Datensätzen füllt.

ten Standard SQL:2003. Im Vergleich zu anderen Datenbanksystemen, kommt die MySQL-Metadatenbank mit der geringen Anzahl von 20 Tabellen aus.

### Fernbedienung für Administratoren

Ebenfalls neu ist der so genannte Instance Manager, der es Administratoren erlaubt, mehrere Datenbanken remote zu starten und/oder zu stoppen. Darüber hinaus überwacht der Instance Manager die Datenbanken und deren Dienste und startet diese zum Beispiel nach einem Crash unverzüglich neu. So können teure Ausfallzeiten auf ein Minimum reduziert werden.

Zusätzlich können die Datenbanken über den Instance Manager remote konfiguriert und deren Logfiles eingesehen und analysiert werden. Leider steht dieses Produkt derzeit nicht für Windows-Systeme zur Verfügung.

### Bäumchen wechsel dich

Zusätzlich hat MySQL der aktuellen Datenbank ein neues Tool beiseite gestellt, um den Wechsel von Fremdprodukten noch einfacher zu gestalten. Das grafisch orientierte „MySQL Migration Toolkit“ ermöglicht derzeit eine problemlose Migration von den folgenden Datenbanksystemen zu MySQL:

- MS Access
- MS SQL Server
- Oracle
- und alle JDBC tauglichen Datenbanken

In acht einfachen Schritten werden wahlweise ganze Schemata oder nur einzelne Tabellen in die MySQL-Datenbank migriert (siehe Abbildung 4). Die internen Test-Migrationen von Oracle 10g/9i und einem MS SQL Server auf MySQL verliefen absolut problemlos.

```
CREATE TABLE remote_plz_orte (
    plz          int(5) NOT NULL,
    ort          varchar(50) NOT NULL,
    PRIMARY KEY (plz, ort),
)
ENGINE=FEDERATED
CONNECTION=
'mysql://user:password@production:3306
/instance1/plz_orte';
```

Abb. 3: Anlegen einer Tabelle, die ihre Daten vom Server „produktion“ aus der Tabelle „plz\_orte“ in der Datenbank „instance1“ bezieht. Natürlich muss für die entfernte Datenbank ein gültiger User Account bereitgestellt werden (hier repräsentiert durch „user“ und „password“).

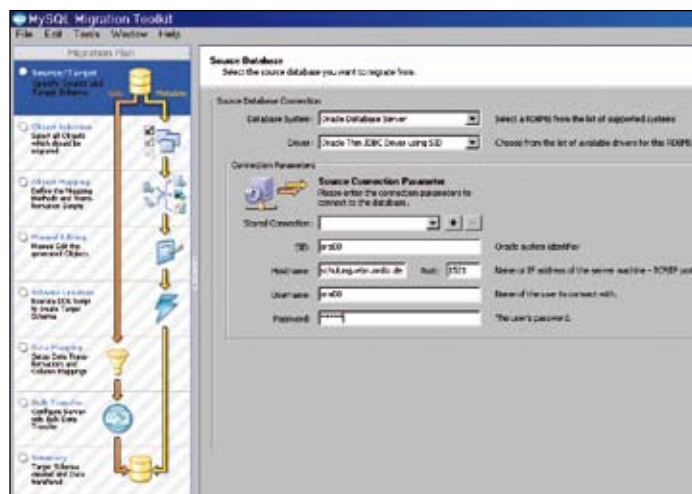


Abb. 4: Das neue „MySQL Migration Tool“ erleichtert jedem DBA die Migration. In acht einfachen Schritten werden die notwendigen Parameter zur Migration abgefragt.

Neben dem grafischen Produkt steht natürlich auch eine Kommandozeile zur Verfügung, über welche sich Migrationsvorgänge sehr leicht automatisieren lassen.

### Fazit

Der Abstand zu den etablierten Datenbanksystemen ist mit der neuen Version kleiner geworden. Viele lange vermisste Features können nun genutzt werden. Ob dies zu einem bahnbrechenden Erfolg der neuen Version führt, ist allerdings fraglich. Neben der konsequenten Weiterentwicklung der Datenbank muss auch am Image des Produktes weiter gearbeitet werden. All zu oft hört man weiterhin das Vorurteil, dass MySQL lediglich als Datenverwalter für Online-Shops oder Gästebücher im Web-Bereich zu gebrauchen sei.

Insofern kommt die neue Version – in Zeiten knapper IT-Budgets – vielleicht gerade im richtigen Moment, um doch noch als „wichtigstes Release“ in der Geschichte von MySQL gefeiert zu werden. Das Potential dafür ist vorhanden.

Haben Sie Interesse am Einsatz von MySQL-Datenbanken oder benötigen Produktschulungen in diesem Umfeld? Dann wenden Sie sich einfach an uns.

Matthias Jung (info@ordix.de).



Jobsteuerung mit dem Microsoft SQL-Server (Teil II):

## Ereignisgesteuerte Transaktionsprotokoll-Sicherung

Im ersten Teil unserer kleinen Artikelserie (siehe ORDIX News 4/2005, Seite 16) über Jobsteuerung behandeln wir deren grundlegende Aspekte. Dieser Beitrag betrachtet zunächst die beiden Objekttypen *Operatoren* und *Warnungen*. Anschließend zeigen wir an einem Beispiel die vielfältigen Einsatzmöglichkeiten der Jobsteuerung auf. Dabei wird der SQL-Server Agent dazu genutzt, einen Datenbankausfall aufgrund eines vollgelaufenen Transaktionsprotokolls zu verhindern, indem dieses automatisch gesichert wird.



Dieser Artikel richtet sich an Administratoren von Microsoft SQL-Servern, die sich einen Überblick über die Möglichkeiten der Jobsteuerung mit dem SQL-Server Agenten verschaffen möchten.

### Operatoren

Beim Objekttyp *Operatoren* handelt es sich um tatsächliche Personen, die vom SQL-Server Agenten über die Ausführung von Aufträgen oder das Auftreten von Warnungen informiert werden. Es gibt mehrere Möglichkeiten, Operatoren zu informieren:

#### Operator per E-Mail informieren

Um E-Mail nutzen zu können, muss auf dem System ein MAPI-1-kompatibler E-Mail-Client installiert sein (beispielsweise Outlook oder der Exchange Client).

#### Operator durch Paging informieren

Für Paging wird zusätzlich zum MAPI-1-kompatiblen E-Mail-Client noch eine Drittanbieter-Software benötigt, die für die Umsetzung von E-Mails in Pager-Nachrichten sorgt.

#### Operator über den „NET SEND“-Befehl informieren

Anhand eines kleinen Beispiels zeigen wir im Quelltext in Abbildung 1, wie man den Operator per „NET SEND“ informiert. Dabei richten wir einen Operator „Administrator vom Dienst“ ein und senden die Nachrichten mit Hilfe des „NET SEND“-Befehls an ihn. Die Nachricht erscheint dann direkt auf dem Bildschirm des Rechners „Adminrechner“.

### Warnungen

Vom Objekttyp *Warnungen* gibt es prinzipiell zwei Arten: Ereignis- und Leistungsstatuswarnungen.

#### Ereigniswarnung

Schwerwiegende Fehler des SQL-Servers (ab Schweregrad 19) oder „RAISERROR WITH LOG“-Anweisungen werden im Windows

Anwendungsprotokoll eingetragen. Zusätzlich können Einträge mit der Prozedur „xp\_logevent“ erzeugt werden.

Die Einträge im Windows-Anwendungsprotokoll werden vom SQL-Server Agenten ausgelesen und mit einer Liste von Meldungen verglichen (SQL-Server Ereigniswarnungen), die der Anwender selbst definiert. Wird eine Übereinstimmung gefunden, so wird zunächst die betreffende Warnung ausgelöst.

```
-- Operator erstellen
EXEC msdb..sp_add_operator
    @name = 'Administrator vom Dienst',
    @netsend_address = 'Adminrechner'
```

Abb. 1: Operator für „NET SEND“ einrichten.

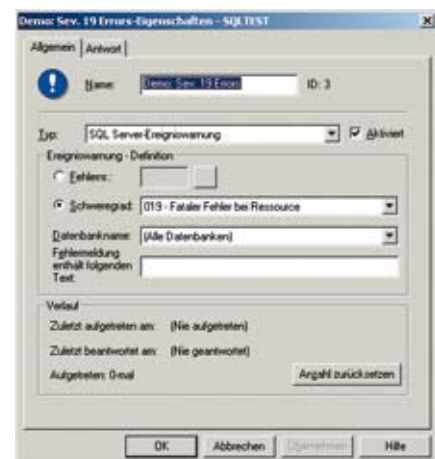


Abb. 2: Vorinstallierte SQL-Server Ereigniswarnung.

```
-- Sicherungsmedium erstellen
EXEC master..sp_addumpdevice
    @devtype = 'disk',
    @logicalname = 'Transaktionslogs',
    @physicalname = '\\backupserver\sqlbackup\Transaktionslogs.bak'
```

Abb. 3: Erstellen eines Sicherungsmediums auf einem entfernten System.

In der Standardinstallation sind zu Demonstrationszwecken bereits einige Warnungen vordefiniert, so auch die in Abbildung 2 dargestellte Warnung.

### Leistungsstatuswarnung

Im Rahmen der Installation des MS SQL-Servers werden im Windows-Systemmonitor zusätzliche Leistungsindikatoren hinzugefügt, die eine Vielzahl von Parametern des SQL-Servers überwachen. Unter anderem werden folgende Werte ermittelt:

- Anzahl der gleichzeitig angemeldeten Benutzer
- Reservierung und Verwendung des Arbeitsspeichers
- Prozentsatz der Seiten, die ohne Festplattenzugriff aus dem Arbeitsspeicher entnommen werden (Caching)
- Prozentsatz (Füllgrad) der verwendeten (Transaktions-)Protokolldatei
- Größe der (Transaktions-)Protokolldatei(en) in KB
- Größe der Datendatei(en) in KB

Für alle Parameter können Schwellwerte definiert werden, so dass beim Erreichen, Über- oder Unterschreiten dieser Schwellwerte eine entsprechende Warnung (SQL-Server-Leistungsstatuswarnung) ausgelöst wird.

### Reaktion auf eine Warnung

Immer wenn eine Warnung ausgelöst wird – unabhängig davon, ob es sich um eine Ereigniswarnung oder eine Leistungsstatuswarnung handelt – wird zunächst ein zugehöriger Zähler hochgezählt. Dieser Zähler kann jederzeit im Enterprise Manager abgefragt werden (siehe Abbildung 2, Bereich „Verlauf“). Er sagt aus, wie oft die jeweilige Warnung bereits ausgelöst wurde.

Bis zu diesem Zeitpunkt wird alles im Hintergrund abgewickelt, der Operator nicht explizit informiert und somit hat er das Problem gegebenenfalls noch nicht realisiert.

Möchte man aber, dass noch eine andere Reaktion neben dem Hochzählen des Zählers stattfindet, kann man zusätzlich zu jeder Warnung zwei Reaktionstypen definieren:

Zum einen kann ein Auftrag ausgeführt werden. Zum anderen kann die Warnung – zusammen mit einer zusätzlichen Nachricht – an einen Operator weitergeleitet werden.

Soweit zu den Grundlagen, die wir im Folgenden anhand eines Beispiels anwenden werden.

### Das Problem des „vollen“ Transaktionsprotokolls

Alle produktiven Datenbanken (von reinen Analysesystemen einmal abgesehen) sollten als Datenbank-Wiederherstellungs-Modell „Vollständig“ eingestellt haben. Dadurch werden alle durchgeführten Transaktionen im Transaktionsprotokoll gespeichert. Die Protokolldatei wird somit stetig größer.

Problematisch wird das, wenn die konfigurierte maximale Größe der Protokolldatei erreicht oder die genutzte Festplatte voll ist. Dann können keine Änderungen mehr in der Datenbank vorgenommen werden.

Die einzige Möglichkeit, den Speicherplatz innerhalb der Protokolldatei wieder freizugeben, ist, eine Sicherung des Transaktionsprotokolls durchzuführen.

### Lösung: Wir sichern bei Bedarf

In unserem Fall kann es natürlich nicht ausreichen, einen Zähler hochzuzählen, den unter Umständen niemand bemerkt. Wir werden also zusätzliche Reaktionen definieren.

Zum einen werden wir folgenden Auftrag einrichten: Der Inhalt des Transaktionsprotokolls soll immer dann auf eine andere Festplatte (oder einen anderen Rechner) gesichert werden, wenn die Protokolldatei zu einem bestimmten Prozentsatz (hier 70 %) gefüllt ist.

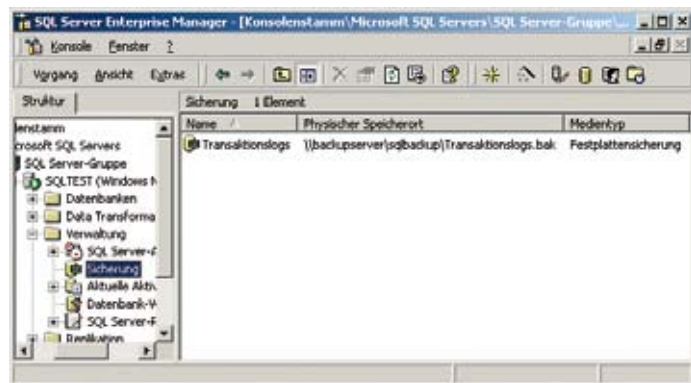


Abb. 4: Verwaltung der Sicherungsmedien im Enterprise Manager.

```
-- Auftrag erstellen
EXEC msdb..sp_add_job
    @job_name = 'Log-Archivierung (Produktion)',
    @owner_login_name = 'sa'

-- Zielsever des Auftrags festlegen
EXEC msdb..sp_add_jobserver
    @job_name = 'Log-Archivierung (Produktion)',
    @server_name = '(local)'

-- Auftragsschritt zu Auftrag hinzufügen
EXEC msdb..sp_add_jobstep
    @job_name = 'Log-Archivierung (Produktion)',
    @step_name = 'BACKUP LOG',
    @subsystem = 'TSQL',
    @command = 'BACKUP LOG Produktion TO Transaktionslogs'
```

Abb. 5: Auftragserstellung zur Sicherung des Transaktionsprotokolls.



```
-- Warnung erstellen
EXEC msdb..sp_add_alert
    @name = 'Log-Archivierung notwendig (Produktion)',
    @performance_condition = 'SQLServer:Databases|Percent Log Used|Produktion|>|70',
    @job_name = 'Log-Archivierung (Produktion)',
    @delay_between_responses = 60
```

Abb. 6: Festlegen, wann eine Warnung ausgelöst werden soll, die die Sicherung des Transaktionsprotokolls anstößt.

So wird zum einen verhindert, dass die Protokolldatei oder die Festplatte vollläuft, zum anderen können die Transaktionen so früher von einem externen Datensicherungsprogramm auf ein Band geschrieben werden und gehen bei einem Systemausfall nicht verloren.

Das Ergebnis ist dann vergleichbar mit dem Archiver unter Oracle, der die Protokolldateien (Online Redologs) zyklisch in so genannte Archivelogs sichert.

Zum anderen werden wir bei einem Fehlschlag des Backups eine Warnung per „NET SEND“ an den bereits eingerichteten Operator senden.

### Schritt 1: Ein Sicherungsmedium erstellen

Zunächst werden wir ein Sicherungsmedium erstellen. Das ist eine Datei, die wie ein Bandlaufwerk genutzt werden kann und in die wir sichern werden. Neue Sicherungen werden dabei immer an die Datei angefügt. Natürlich kann als Sicherungsmedium auch ein angeschlossenes Bandlaufwerk genutzt werden.

Neben der Sicherung in eine Datei auf dem lokalen Server kann auch, wie in unserem Beispiel in Abbildung 3, eine Datei auf einem entfernten System verwendet werden, wenn diese über eine Netzwerkfreigabe erreichbar ist. Dabei ist darauf zu achten, dass das Dienste-Konto des SQL-Servers ausreichende Rechte hat, in die Datei zu schreiben.

Die Sicherungsmedien können auch über den Enterprise Manager erstellt und verwaltet werden (siehe Abbildung 4).

### Schritt 2: Das Backup-Skript als Auftrag

Im zweiten Schritt erstellen wir nun den Auftrag (siehe Abbildung 5), der als Kern die Anweisung zur Sicherung des Transaktionsprotokolls enthält.

Wir kümmern uns in diesem Beispiel nur um die Datensicherung der Transaktionen, aber natürlich könnte als zweiter Schritt innerhalb des Auftrages auch die Sicherung der erstellten Datei mit Hilfe eines zusätzlichen Datensicherungsprogramms auf ein Band angestoßen werden.

Wenn Sie – unabhängig vom Füllgrad der Protokolldatei – in regelmäßigen Abständen eine Sicherung des Transaktionsprotokolls durchführen möchten, so können Sie jetzt einen Zeitplan für diesen Job einrichten. Das hätte den Vorteil, dass auch in Zeiten mit nur wenigen Transaktionen alle Transaktionen nach einem festgelegten Zeitraum gesichert werden.



Abb. 7: Die im Beispiel definierte SQL-Server Leistungsstatuswarnung.

### Schritt 3: Eine Warnung zur Überwachung

Warnungen können entweder mit dem Enterprise Manager oder der Prozedur `sp_add_alert` erstellt werden. Bei der Verwendung der Prozedur bestimmen die verwendeten Parameter, ob eine Ereigniswarnung oder eine Leistungsstatuswarnung erstellt wird (siehe Abbildung 6).

Wir verwenden – neben dem notwendigen Parameter `name` für den Namen der Warnung – den Parameter `performance_condition` zur Definition des zu überwachenden Leistungsobjektes, des Schwellwertes sowie des Vergleichsoperators.

Zusätzlich geben wir mit dem Parameter `job_name` den Namen des als Reaktion auszuführenden Auftrages an.

Der Parameter `delay_between_responses` bewirkt, dass innerhalb des in Sekunden angegebenen Zeitraums der Auftrag auch dann nicht erneut ausgeführt wird, wenn die Warnung wiederholt ausgelöst wird.

In unserem Fall hat der im Auftrag ausgeführte Befehl `BACKUP LOG . . .` also eine Minute Zeit, das Transaktions-Protokoll zu sichern und den Speicherplatz wieder freizu-

```
-- Operator benachrichtigen
EXEC msdb..sp_update_job
  @job_name = 'Log-Archivierung (Produktion)',
  @notify_level_netsend = 2 -- bei Fehler
  @notify_netsend_operator_name = 'Administrator vom Dienst'
```

Abb. 8: Benachrichtigung des Operators im Fehlerfall festlegen.

geben, bevor ein erneuter Sicherungsbefehl abgesetzt wird.

Wie diese Warnung im Enterprise Manager definiert wird, zeigt Abbildung 7.

Damit ist unser Ziel erreicht und eine zu mehr als 70 Prozent gefüllte Protokolldatei löst die definierte Leistungsstatuswarnung aus, die wiederum den Auftrag anstößt, der das Transaktionsprotokoll sichert.

#### Schritt 4: Der Operator wird per „NET SEND“ benachrichtigt

Bei einem Fehler während der Bearbeitung des Auftrags, also in unserem Beispiel bei einem fehlgeschlagenen Backup-Versuch, wird zwar ein Eintrag im Windows-Ereignisprotokoll erzeugt, eine automatische Benachrichtigung erfolgt jedoch nicht. Hier kommt nun der zu Anfang angelegte Operator ins Spiel.

Dieser Operator wird bei einem Fehlschlag unseres Auftrages benachrichtigt. Dazu ist der Auftrag mit dem Befehl in Abbildung 8 zu ändern.

#### Fazit

Der SQL-Server Agent bietet neben dem Funktionsumfang einer klassischen Jobsteuerung – Ausführung von Datenbankbefehlen zu vorgegebenen Zeitpunkten – zusätzliche Möglichkeiten, unter anderem:

- Ausführung eines Auftrages als Reaktion auf das Auftreten eines definierten Ereignisses.
- Benachrichtigung von Operatoren auf verschiedenen Kommunikationswegen.

Natürlich sollten Sie beim Stichwort Sicherung nicht nur an die Sicherung des Transaktionsprotokolls denken. Auch die Datenbank-Vollsicherung ist immens wichtig.

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

## Impressum



#### Herausgeber:

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

#### Redaktion:

Helma Jenniches, Sascia Brinkmann

**V.i.S.d.P.:** Wolfgang Kögler

#### Anschrift der Redaktion:

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

#### Gestaltung/Layout:

Sascia Brinkmann

**Auflage:** 8.900

#### Druck:

Druckerei Reike GmbH,  
Paderborn

#### Autoren dieser Ausgabe:

Christof Amelunxen, Klaus Günther, Kathrin Hammerschmidt, Andreas Jordan, Matthias Jung, Wolfgang Kögler, Andreas Kother, Michael Lindermann, Klaus Reimers, Rainer Restat, Markus Schreier, Christiane Westerhoff, Christian Wiesing

#### Copyright:

ORDIX AG. Alle Rechte vorbehalten. Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

Die Zeitschrift ORDIX News wird von der ORDIX AG an ausgesuchte Kunden verteilt und kann für 2,20 Euro bestellt werden. Sie finden sowohl die neueste Ausgabe als auch ältere Ausgaben im Archiv der ORDIX News im Internet unter: <http://www.ordix.de>. Schauen Sie mal rein!

Der Kontakt zu unseren Lesern ist uns sehr wichtig. Für Anregungen, Kritik und Anmerkungen zu den Themen, aber auch für interessante Ideen sind wir immer offen und dankbar. Sie erreichen die Redaktion auch per E-Mail unter [redaktion@ordix.de](mailto:redaktion@ordix.de).

Wir freuen uns auf Ihr Feedback.





Tiefe Einblicke in Solaris 10 mit DTrace (Teil II):

# Mit DTrace „Ecken ausleuchten“

In der ORDIX News 4/2005 sind wir in das Thema DTrace eingestiegen. Mit diesem Artikel wird ein Schritt mehr Vertrautheit im Umgang mit DTrace erreicht, indem wir verdeutlichen, welche Systemeigenschaften sich mit Providern analysieren lassen.



Dieser Artikel richtet sich an Administratoren und Entwickler, die weitgreifende Analysemöglichkeiten unter Solaris 10 nutzen möchten.

Der Hauptteil dieses Artikels wird sich mit Messpunkten befassen, beziehungsweise mit den Providern, die diese Messpunkte bereitstellen. Einleitend möchten wir deshalb eine ausgereifte Skriptensammlung darstellen: Das DTraceToolkit. In vielen Fällen reichen diese „fertigen“ Skripte schon für eine Analyse aus.

## DTraceToolkit

Mit über 80 Skripten ist das DTraceToolkit [1] sehr umfassend. Für eine flexiblere Handhabung sind die Skripte teilweise in Perl oder Shell geschrieben. Das eigentliche D-Skript wird darin definiert und kommt dann zur Ausführung.

Durch dieses Vorgehen ist es möglich, Parameter auszuwerten, die den genauen Aufbau des D-Skriptes und somit die Ausgabe beeinflussen. Außerdem finden sich unter [1] zu allen Skripten Man Pages, Beispiele und teilweise noch zusätzliche Erklärungen.

## Provider

Für eine Analyse ist es notwendig, relevante Messpunkte zu aktivieren. Sobald ein Thread einen aktivierten Messpunkt durchläuft, „zündet“ dieser (im Englischen: „the probe fires“) und der zum Messpunkt gehörige Anweisungsblock wird ausgeführt.

Die Informationen des Messpunktes werden gesammelt und angezeigt. Von circa 20 unterschiedlichen Providern werden über 30.000 Messpunkte bereitgestellt. Im Folgenden stellen wir diese Provider vor und betrachten einige Beispiele.

```

1  #!/usr/bin/dtrace -qs
2  dtrace::BEGIN
3  {
4  printf("%20s %7s %7s\n", "Time", "sread/s", "swrit/s");
5  }
6
7  sysinfo::sysread { sread++; }
8  sysinfo::syswrite { swrit++; }
9  profile::tick-1sec
10 {
11  printf("%20Y %7d %7d\n", walltimestamp, sread, swrit);
12  sread = 0; swrit = 0;
13  }

```

Abb. 1: Kleines sar mit Hilfe von dtrace- und profile-Provider.

## Anfang und Ende (dtrace-Provider)

Der dtrace-Provider stellt die Messpunkte dtrace::BEGIN dtrace::END und dtrace::ERROR bereit. BEGIN wird vor allen anderen Messpunkten ausgewertet. Er kann genutzt werden, um notwendige Initialisierungen durchzuführen oder eine Überschrift zu erzeugen. END wird nach allen anderen Messpunkten durchlaufen. BEGIN und END entsprechen den namensgleichen Blöcken in awk.

In unserem Beispiel in Abbildung 1 werden die Read- und Write-Aktivitäten gezählt und jede Sekunde ausgegeben. Im BEGIN-Messpunkt dieses Beispiels wird die Überschrift erzeugt. Der END-Messpunkt wird dazu genutzt, um gesammelte (summierte) Informationen auszugeben.

Der ERROR-Messpunkt wird bei Laufzeitfehlern durchlaufen und dient der Fehlerbehandlung.

## Metronom (profile-Provider)

Der profile-Provider stellt Messpunkte bereit, die in regelmäßigen Intervallen zünden. Hochfrequente Intervallzeiten bis hin zu 200 Mikrosekunden sind möglich.

Die Intervalldauer wird an den Namen des Messpunktes mit der entsprechenden Einheit angehängt (z. B. profile::profile-tick-10ns). Die möglichen Suffixe mit den entsprechenden Zeiteinheiten sind in Abbildung 2 dargestellt.

Suffix	Zeiteinheit
nsec oder ns	Nanosekunden
usec oder us	Mikrosekunden
Msec oder ms	Millisekunden
sec oder s	Sekunden
min oder m	Minuten
Hour oder h	Stunden
Day oder d	Tage
Hz oder kein Suffix	Herz (Anzahl pro Sekunde)

Abb. 2: Suffixe des profile-Providers.

Die profile-<n> Messpunkte zünden regelmäßig auf allen CPUs. Es können somit z. B. Statistiken über die auf den CPUs laufenden Prozesse ermittelt werden. Die profile-tick-<n> Messpunkte zünden nur auf einer CPU. Sie dienen z. B. der Ausgabe von Informationen, die durch andere Messpunkte gesammelt wurden.

Es lässt sich somit z. B. das Verhalten von `sar` nachbauen, welches in regelmäßigen Intervallen die aktuelle Auslastung bzw. die Häufigkeit bestimmter Ereignisse darstellt.

In Abbildung 1 werden in Zeile 7 und 8 Lese- und Schreibereignisse gezählt. Jede Sekunde (Zeile 9) werden die Zähler zusammen mit dem aktuellen Zeitstempel (Zeile 11) ausgegeben. Nach einer Ausgabe werden die Zähler wieder auf 0 zurückgesetzt (Zeile 12).

### Truss+ (syscall-Provider)

Schon im ersten Teil der Artikelreihe, in der ORDIX News 4/2005, S. 12, nutzten wir den syscall-Provider. Er stellt für jeden Systemcall einen Entry und einen Return-Messpunkt bereit. Diese Messpunkte erlauben es, z. B. `truss` nachzubauen.

Mit DTrace ist aber mehr denkbar: So können recht einfach Statistiken über die verwendeten Systemcalls erstellt werden, indem z. B. dargestellt wird, wie lange ein Systemcall gedauert hat. Dies wird erreicht, indem die Differenz der Zeitstempel beim Entry- und Return-Messpunkt gemessen wird.

Das Beispiel aus der letzten ORDIX News wurde in Abbildung 3 um die Zeilen 7, 13 und 14 ergänzt. In Zeile 7 wird für jeden Thread separat der aktuelle Zeitstempel (in Nanosekunden) festgehalten.

In Zeile 13 wird die verstrichene Zeit berechnet und das Ergebnis direkt ausgegeben. In Zeile 14 wird die nicht mehr benötigte Variable auf 0 gesetzt, da dies unter DTrace gleichzeitig den verwendeten Speicherplatz wieder freigibt.

### Userlandtracing (pid-provider)

Der pid-Provider bietet für jeden Funktionsaufruf eines jeden Prozesses eigene Messpunkte. Bei der Angabe des Messpunktes wird die Prozess-ID des zu überwachten Prozesses schon im Namen des Providers angegeben. Die Verwendung von Variablen ist möglich. Der Messpunkt in Zeile 2 der Abbildung 4 zündet bei allen libc-Aufrufen eines angegebenen Prozesses.

Pauschal alle Funktionsaufrufe eines Prozesses durch DTrace zu analysieren, ist wenig sinnvoll. Das Ergebnis ist zu umfangreich und damit eher unbrauchbar. Der Aufruf von `ls` in einem kleinen Verzeichnis erzeugt in unserer Testumgebung insgesamt 5870 Funktionsaufrufe von 271 verschiedenen Funktionen.

### Wann komme ich dran? (sched-Provider)

Der sched-Provider bietet unter anderem Messpunkte, die zünden

- wenn ein Prozess seine Priorität verändert bekommt,
- wenn dieser einem Prozessor zugeteilt wird oder
- wenn er von einem wieder heruntergenommen wird.

Das Skript in Abbildung 5 stellt dar, wie lange welcher Prozess eine CPU zugeteilt bekam. Die beiden Messpunkte (Zeile 1 bzw. Zeile 5) zünden genau dann, wenn ein Thread die CPU zugeteilt bekommt beziehungsweise wenn er sie wieder verliert.

Aggregatfunktionen ermöglichen es, die Ausgabe (siehe Abbildung 6) zu optimieren, doch möchten wir dieses Skript lieber einfach halten.

```

1  #!/usr/sbin/dtrace -qs
2
3  syscall:::entry
4  /pid == $target/
5  {
6      printf("%s(%d, 0x%x, %4d)", probefunc, arg0, arg1, arg2) ;
7      self->start = timestamp ;
8  }
9  syscall:::return
10 /pid == $target/
11 {
12     printf("\t\t = %d\t", arg1) ;
13     printf("Time: %dmicrosecs\n", (timestamp - self->start) /1000) ;
14     self->start = 0 ;
15 }
```

Abb. 3: Dauer von Systemcalls messen.

```

1  #!/usr/sbin/dtrace -s
2  pid$target:libc:::entry
3  {
4      @[probefunc]=count()
5  }
```

Abb. 4: Zählen aller Aufrufe von libc-Funktionen.

```

1  sched:::on-cpu
2  {
3      self->ts = timestamp;
4  }
5  sched:::off-cpu
6  /self->ts/
7  {
8      printf ("%10s %d\n", execname, (timestamp - self->ts) );
9      self->ts = 0;
10 }
```

Abb. 5: Wieviel Zeit verbringen Prozesse auf einer CPU?



## Wer sperrt wann was? (lockstat- und plockstat-Provider)

Bestimmte Zugriffe dürfen von unterschiedlichen Threads nicht gleichzeitig ausgeführt werden. Um solche gleichzeitigen Zugriffe zu verhindern, wird die entsprechende Ressource gesperrt (locking).

Das Locking-Verhalten im Kernel kann durch den lockstat-Provider erforscht werden. Auch das bisher unter Solaris vorhandene Unix-Kommando `lockstat` nutzt nun den lockstat-Provider, um entsprechende Daten anzuzeigen. Der plockstat-Provider bietet für Locking-Mechanismen im User-Land die entsprechenden Messpunkte.

Beide Provider bieten prinzipiell zwei Arten von Messpunkten:

### „Hold-event“-Messpunkte

Diese werden bei jedem Zugriff auf einen Sperrmechanismus durchlaufen. Allein die Kernel-Sperrmechanismen werden auf einem gut ausgelasteten Solaris-System in einer Größenordnung von mehreren Millionen mal pro Sekunde und CPU angefordert oder freigegeben. Die Aktivierung eines solchen Messpunktes kann somit durchaus die Performance des Gesamtsystems beeinflussen.

### „contention-event“-Messpunkte (contention = Konkurrenzsituation)

Sie werden nur gezündet, wenn ein Thread auf einen anderen warten muss. Diese Fälle sollten möglichst selten eintreten. Die Aktivierung eines contention-event-Messpunktes stellt somit keine Beeinflussung für das Gesamtsystem dar.

### Von „Gabeln“ und der „Exekutive“ – fork and exec (proc-Provider)

Das Entstehen eines neuen Prozesses durch `fork` und das meist anschließende Überlagern mittels `exec()` kann durch Messpunkte des proc-Providers analysiert werden. Separate Messpunkte werden im Fehlerfall gezündet.

Auch das Beenden von Prozessen oder der Empfang von Signalen wird durch Messpunkte des proc-Providers greifbar. Das Beispiel in Abbildung 7 gibt aus, welcher Prozess von welchem anderen durch `exec` überlagert wurde.

Aufgerufen wird `exec` von dem Prozess, der zunächst geklont wurde und nun überlagert werden soll (quasi beendet sich der geklonte Vater dadurch selbst). In den Zeilen 2 bis 5 wird der Prozessname in der Variablen `self->parent` (Zeile 4, `execname`) für den aktuellen Thread festgehalten. Ist `exec` erfolgreich abgeschlossen, so ist der Prozess überlagert und wir finden als Prozessname den neuen Prozess (Zeile 9, `execname`).

Wie oft das Wertepaar `self->parent, execname` auftritt, wird in Zeile 9 gezählt. Im END-Messpunkt werden eine Überschrift (Zeile 19) und alle bisher festgehaltenen Wertepaare mit dem aktuellen Zählerstand (Zeile 20) ausgegeben. Durch Zeile 10 und Zeile 15 werden nicht mehr benötigte Variablen auf Null gesetzt und somit freigegeben.

```

sched 44475
sshd 1391862
sh 3235674
xauth 13884316
sched 33131
xauth 252894
sh 464464
sched 9627974
xauth 799826
sched 61670
sched 1380158
sched 1641965
ksh 12031592
fsflush 707995
ksh 4370410

```

Abb. 6: Ausgabe vom Skript aus Abbildung 5.

## Selbstgebacken (sdt-Provider)

Anwendungsentwickler können in ihren Anwendungen auch eigene Messpunkte definieren. Diese Messpunkte werden durch den sdt-Provider bereitgestellt. Auch im Kernel finden sich einige Messpunkte, die über den Statically Defined Tracing (sdt)-Provider bereitgestellt werden. In zukünftigen Solarisversionen können diese Messpunkte jedoch hinsichtlich des Namens und des Verhaltens verändert worden sein.

## Statistiken (sysinfo- und vminfo-Provider)

Der Kernel hält eigene Statistiken in der Sys- und in der vm-Struktur bereit. In der erstgenannten wird z. B. die User-, Sys-, Wait-IO- oder Idle-Time der CPU nachgehalten. Bisher konnten diese Informationen nur mit den Unix-Kommandos `mpstat`, `vmstat` oder `sar` ausgewertet werden.

Für diese Kernel-Statistiken gibt es eigene Provider. Vor der Veränderung eines Wertes zündet der entsprechende Messpunkt des sysinfo-Providers bzw. des vminfo-Providers.

### Statistiken II (io-Provider)

Engpässe bei der Ein- und Ausgabe sind häufig der Grund für Performance-Probleme. Der io-Provider stellt Messpunkte bezüglich Platten-IO bereit. Das Unix-Kommando `iostat` zeigt die Daten wie bisher an. Bei Veränderung der damit sichtbaren Daten zünden Messpunkte des io-Providers.

### Rechnen oder Rechnen lassen (fpinfo-Provider)

Heutige Prozessoren bieten eine eigene Floating Point (FP)-Arithmetik. Nur wenige FP-Be-

**Link**

► [1] DtraceToolkit:  
<http://www.opensolaris.org/os/community/dtrace/dtracetoolkit/>

fehle müssen durch den Kernel simuliert werden. Welche Befehle das sind, hängt stark von dem tatsächlichen Microprozessor ab. Nutzt eine Applikation häufig einen simulierten Befehl, kann dies zu Performance-Einbußen führen.

Anzeigen lassen sich entsprechende Statistiken mit dem `kstat`- oder dem `trapstat`-Befehl. Der `fpuinfo`-Provider stellt die dazugehörigen Messpunkte bereit.

**Spurensuche für Unentwegte (fbt-Provider)**

Die meisten Funktionen im Kernel können durch einen Entry- und einen Return-Messpunkt analysiert werden. Bereitgestellt werden diese Messpunkte durch den Function Boundary Tracing (fbt)-Provider. Dieser Provider liefert auf unserem System über 33.000 Messpunkte für 144 verschiedene Module.

Für ein gutes Verständnis dieses Providers sind Kernel-Architekturkenntnisse notwendig. In dem oben erwähnten DTraceToolKit werden diese Provider für die Netzwerkanalyse und für die Analyse des Pufferverhaltens bei Dateisystemzugriffen verwendet. Folgender Einzeiler (hier aus Platzgründen zweizeilig dargestellt) listet alle Funktionsaufrufe aus dem `ufs`-Modul:

```
dtrace -n fbt:ufs::entry \
' {@a[probefunc] = count() }'
```

**Fazit**

Der Artikel stellte die unterschiedlichen Provider vor und zeigt einige Anwendungsbeispiele. Für eine aussagekräftige Analyse müssen die richtigen Messpunkte gefunden werden. Welche Provider für bestimmte Fragestellungen weiterhelfen können, wurde in diesem Artikel deutlich.

Neben der Kenntnis der Provider und damit dem Wissen von der Existenz der Messpunkte können weitere Eigenschaften der Programmiersprache D ergründet werden. Dies wird in einer der nächsten ORDIX News folgen.

Markus Schreier (info@ordix.de).

```
1 #!/usr/sbin/dtrac -qs
2 proc:::exec
3 {
4     self->parent = execname;
5 }
6 proc:::exec-success
7 /self->parent != NULL/
8 {
9     @[self->parent, execname] = count();
10    self->parent = NULL;
11 }
12 proc:::exec-failure
13 /self->parent != NULL/
14 {
15    self->parent = NULL;
16 }
17 END
18 {
19    printf("%-20s %-20s %s\n", "WHO", "WHAT", "COUNT");
20    printa("%-20s %-20s %d\n", @);
21 }
```

Abb. 7: Beispiel für den `proc`-Provider: Statistik, über `Exec`-Aufrufe.

**Glossar**

<b>Awk</b>	Programmiersprache zur Bearbeitung und Auswertung von Textdateien. Sie ist benannt nach den Initialen ihrer Programmierer: Alfred V. Aho, Peter J. Weinberger und Brian W. Kernighan.
<b>exec</b>	System Call, mit dem ein bestehender Prozess durch ein anderes Programm überlagert wird.
<b>fork</b>	System Call, mit dem sich ein Prozess „verdoppelt“. Soll z. B. ein neues Programm gestartet werden, so wird der bestehende (aufrufende) Prozess erst mit <code>fork()</code> geklont, bevor der Klon dann mittels <code>exec()</code> durch das aufgerufene Programm überlagert wird.
<b>Management Information Base (MIB)</b>	MIB ist eine hierarchisch aufgebaute Datenbank, in der Daten zur Verwaltung von Netzwerkkomponenten hinterlegt sind. Management Software kann über das Simple Network Management Protokoll (SNMP) bestimmte Zweige in der MIB abfragen oder Werte ändern und somit das System steuern.
<b>Prozess</b>	Programm, welches in den Hauptspeicher geladen wurde und gerade abgearbeitet wird.
<b>System activity report (sar)</b>	Analyseprogramm unter System V.5 zum Ermitteln der momentanen Systemaktivität.
<b>System call</b>	Aufruf einer Kernel-Funktion durch einen Prozess im User Space.
<b>System Space</b>	Abläufe, die vom Kernel durchgeführt werden, finden im System Space statt.
<b>Thread</b>	Teil eines Prozesses. Eine 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 („multithreading“).
<b>Truss</b>	Analyseprogramm unter System V.4 zum Protokollieren von System Calls eines Prozesses.
<b>User Space</b>	Abläufe, die ohne Beihilfe des Kernels durchgeführt werden, finden im User Space statt. Fast alle Programme werden im User Space ausgeführt.



# Oracle und XML (Teil I): Ein besonderer Cocktail

Relationale Datenbanken sind unumstritten im Einsatz, wenn es um die Verwaltung großer Datenmengen geht. XML hat sich als Format für den Datenaustausch zwischen Anwendungen etabliert. Oracle hat die Bedeutung von XML erkannt und bietet deshalb eine eigene XML-Schnittstelle an.



Dieser Artikel richtet sich an Datenbankentwickler und Software-Architekten.

XML-Dokumente können in Oracle auf verschiedene Arten gespeichert werden:

1. Die einfachste Methode: Das XML-Dokument wird als Dokument in eine CLOB-Spalte übernommen.
2. Die zweite Methode, bei der Oracle das XML-Dokument analysiert und die Daten in Tabellen ablegt, erläutern wir in den Abschnitten „Parse von XML-Dokumenten“ und „XML SQL Utility“.
3. Die dritte Methode, bei der Oracle das XML-Dokument als XML-Type ablegt, werden wir in Teil II genauer betrachten.

Im Gegenzug zum Speichern können Daten aus Datenbanktabellen, die mittels SQL-Abfrage ermittelt wurden, auch wieder in XML-Dokumente umgewandelt werden. Abbildung 1 gibt einen Überblick über alle Komponenten, die in den beiden Teilen dieses Artikels vorgestellt werden.

## Speichern in CLOB-Feldern

Die einfachste Methode, XML-Dokumente in einer Datenbank zu halten, sind CLOB-Felder. Wenn man XML-Daten in einem CLOB-Feld abspeichert, bedeutet dies, dass sie genau so abgelegt werden, wie sie geliefert werden. Daher kann beim Wiederherstellen des Dokuments kein Daten- oder Informationsverlust entstehen.

Diese Speichermethode ist immer dann von Vorteil, wenn die Struktur der XML-Daten veränderlich ist, weil eine aufwändige Migration der Datenbankobjekte entfallen kann. Allerdings hat die Speicherung von XML-Daten in CLOB-Feldern auch Nachteile. So ist die Unterstützung der Datenbank bei der echten Verarbeitung die-

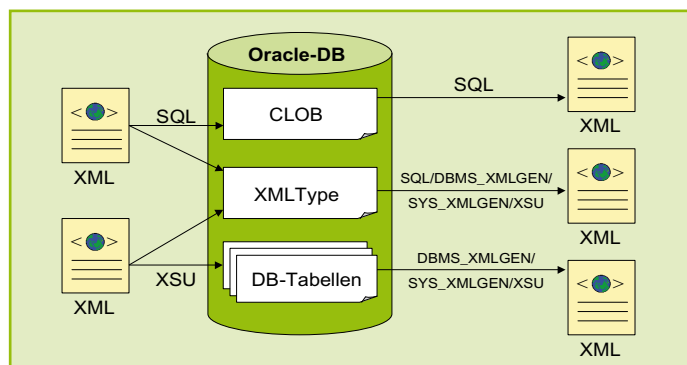


Abb. 1: Überblick über sämtliche Komponenten, die in diesem Artikel vorgestellt werden.

ser Dokumente nur sehr begrenzt. Außerdem muss bei einem Update immer das komplette Dokument ersetzt werden, was nicht unbedingt der performanteste Weg ist. Auch die begrenzte Möglichkeit einer effizienten Indizierung ist hier zu nennen. Sie ist in diesem Fall nur über Oracle-Text möglich.

## XML generieren: Das Paket DBMS\_XMLGEN

Mit dem Paket `DBMS_XMLGEN`, welches in Oracle in der Standardinstallation zur Verfügung steht, lassen sich XML-Dokumente aus einer SQL-Abfrage generieren und ausgeben, aber auch in einem CLOB-Feld abspeichern.

In Abbildung 2 ist die Prozedur `xml_to_clob` dargestellt, mit deren Hilfe wir aus den Daten einer Tabelle `auto` ein XML-Dokument erstellen möchten. Die Tabelle `auto` soll Autonamen enthalten. Dazu legen wir die Tabelle mit zwei Spalten an. Zum einen die Spalte `AUTO_ID (NUMBER)`, die eine eindeutige ID enthält und zum anderen die Spalte `AUTO_NAME (VARCHAR2)`, die den Markennamen des Autos enthält.

Um das XML-Dokument zu generieren, verwenden wir Funktionen, die das Paket `DBMS_XMLGEN` zur Verfügung stellt:

- **newContext:** Mit dieser Funktion setzt man den Kontext, indem man die Datenbankabfrage als String übergibt. So erhält man ein Context-Handle, welches als Parameter für die Funktion `getXML` benötigt wird.
- **setRowSetTag:** Diese Funktion definiert das Root-Element des XML-Dokuments. In unserem Beispiel soll das Root-Element den Namen `Autoliste` haben.
- **setRowTag:** Mit der Funktion `setRowTag` kann man das Tag, welches eine Ergebniszeile aus der Abfrage repräsentiert, definieren. In unserem Beispiel soll das Tag für jede Ergebniszeile den Namen `Auto` haben.
- **getXML:** Diese Funktion ist dafür da, aus der Abfrage ein XML-Dokument zu generieren. Dazu übergibt man der Funktion das Context-Handle, welches die Abfrage enthält. Diese Funktion liefert das XML-Dokument in Form eines CLOBs zurück.

In unserem Beispiel verwenden wir all diese Funktionen. Zunächst setzen wir den Kontext, welchem die Datenbankabfrage `select * from auto` übergeben wird. Diese Abfrage selektiert die Daten, die wir zum Erzeugen unseres XML-Dokuments verwenden wollen, aus der Datenbank.

Anschließend definieren wir die Tags für jede Ergebniszeile mit der Funktion `setRowTag` und das Root-Element mit der Funktion `setRowSetTag`.

Anschließend rufen wir die Funktion `getXML` auf und speichern das Ergebnis in der Variablen `erg`. Den Inhalt dieser Variablen speichern wir nun in die Tabelle `temp_clob` (`temp_clob` besteht aus einem Feld vom Typ CLOB).

Dieses ist nötig, da wir einen CLOB in unserer Prozedur nicht ohne weiteres ausgeben können. Daher speichern wir das Ergebnis zunächst in der Tabelle ab und selektieren es dann wieder aus der Datenbank.

Durch das Selektieren bekommen wir das in Abbildung 3 dargestellte Ergebnis ausgegeben. Wir bekommen ein XML-Dokument, welches, wie durch die Funktion `setRowSetTag` definiert, `Autoliste` als Root-Element enthält.

Jede Ergebniszeile wird, wie durch die Funktion `setRowTag` definiert, durch das Tag `<Auto>` dargestellt. Innerhalb der Ergebniszeile werden die eigentlichen Daten dargestellt. Diese Tags sind genauso benannt wie die Spalten der Tabelle, nämlich `AUTO_NAME` und `AUTO_ID`.

### Parsen von XML-Dokumenten

Mit den PL/SQL-Packages `XMLParser` und `XMLDOM` stellt Oracle Funktionen zum Parsen und Transformieren von XML-Dokumenten in der Datenbank zur Verfügung. Mit dem Paket `XMLParser` wird das Dokument analysiert und der DOM-Baum aufgebaut, der im Hauptspeicher zur Verfügung steht.

Das Dokument wird vom Parser geparkt, wodurch sicher gestellt werden kann, dass das Dokument „wohlgeformt“ (`wellformed`) ist. Wenn eine DTD vorliegt und die entsprechende Funktionalität des Parsers vom Entwickler eingeschaltet ist, wird gegen diese auch noch validiert.

Oracle stellt Parser für PL/SQL, Java, C++ und C zur Verfügung. Nun wollen wir uns die Funktionalität des Parsens eines XML-Dokumentes innerhalb der Datenbank genauer ansehen (siehe Abbildung 4).

```
CREATE OR REPLACE PROCEDURE xml_to_clob
IS
  -- Anlegen der Variable, für den Context-Handle
  qryCtx DBMS_XMLGEN.ctxHandle;
  -- Anlegen der Variable erg vom Typ CLOB. Diese
  -- Variable soll das Ergebnis abspeichern
  erg CLOB;
BEGIN
  -- Kontext mit der Abfrage, aus der das XML
  -- generiert werden soll, erzeugen
  qryCtx := dbms_xmlgen.newContext
    ('SELECT * FROM auto');
  -- ROW durch 'Auto' ersetzen:
  dbms_xmlgen.setRowTag(qryCtx, 'Auto');
  -- ROWSET durch 'Autoliste' ersetzen:
  dbms_xmlgen.setRowSetTag(qryCtx, 'Autoliste');
  erg := dbms_xmlgen.getXML(qryCtx);
  -- das generierte XML-Dokument in der
  -- Tabelle temp_clob speichern.
  INSERT INTO temp_clob VALUES (erg);
  -- Context schließen
  dbms_xmlgen.closeContext(qryCtx);
END;
/
```

Abb. 2: Generieren von XML aus einer SQL-Abfrage mit DBMS\_XMLGEN.

```
<?xml version="1.0" ?>
<Autoliste>
  <Auto>
    <AUTO_ID>1</AUTO_ID>
    <AUTO_NAME>Alfa Romeo 156</AUTO_NAME>
  </Auto>
  <Auto>
    <AUTO_ID>2</AUTO_ID>
    <AUTO_NAME>Alfa Romeo 147</AUTO_NAME>
  </Auto>
</Autoliste>
```

Abb. 3: Ergebnis einer Abfrage mit DBMS\_XMLGEN.

```
CREATE OR REPLACE FUNCTION parse_XMLDoc
(xmlDoc IN CLOB,
wellformed OUT BOOLEAN)
RETURN xmlDom.DOMDocument IS
  v_xmlDoc xmlDom.DOMDocument;
  v_parser xmlparser.Parser;
  parserError EXCEPTION;
  PRAGMA EXCEPTION_INIT (parserError, -20100);
BEGIN
  v_parser := xmlparser.newParser;
  xmlparser.ParseClob(v_parser, xmlDoc);
  v_xmlDoc := xmlparser.getDocument(v_parser);
  xmlparser.freeParser(v_parser);
  wellformed := TRUE;
  RETURN v_xmlDoc;
EXCEPTION
  WHEN parserError THEN
    xmlparser.freeParser(v_parser);
    wellformed := FALSE;
END;
```

Abb. 4: Parsen eines XML-Dokumentes innerhalb der Datenbank.



Wir legen die Prozedur *parse\_XMLDoc* an, die ein XML-Dokument parsen soll, welches ihr als CLOB übergeben wird. Die Prozedur gibt das geparsete DOM-Dokument zurück. Zusätzlich wird ein Boolean zurückgeliefert, der aussagt, ob die Datei wohlgeformt ist oder nicht.

Zunächst werden die lokalen Variablen deklariert, die wir benötigen. Variable *v\_xmlDoc* soll den Rückgabewert der Prozedur auf-

```
DECLARE
  -- Objekte definieren
  queryCtx DBMS_XMLQuery.ctxType;
  ergebnis CLOB;
BEGIN
  -- neuen Kontext anlegen und SQL-Statement setzen
  queryCtx := DBMS_XMLQuery.newContext
    ('SELECT AUTO_ID, AUTO_NAME FROM auto');
  -- SQL-Statement ausführen und mit der Funktion
  -- getXML das XML-Dokument generieren.
  -- Das XML-Dokument wird in der Ergebnis-
  -- variablen erg vom Typ CLOB abgespeichert
  ergebnis := DBMS_XMLQuery.getXML(queryCtx);
  -- in Tabelle abspeichern
  INSERT INTO temp_clob VALUES (erg);
  -- Kontext schließen
  DBMS_XMLQuery.closeContext(queryCtx);
END;
/
```

Abb. 5: Generierung eines XML-Dokuments mit XSU.

```
<?xml version="1.0" ?>
<ROWSET>
  <ROW num="1">
    <AUTO_ID>1</AUTO_ID>
    <AUTO_NAME>Alfa Romeo 156</AUTO_NAME>
  </ROW>
  <ROW num="2">
    <AUTO_ID>2</AUTO_ID>
    <AUTO_NAME>Alfa Romeo 147</AUTO_NAME>
  </ROW>
</ROWSET>
```

Abb. 6: XML-Dokument, welches mit XSU generiert wurde.

```
CREATE OR REPLACE PROCEDURE
  insert_auto(pXmlDoc IN VARCHAR2) IS
  insCtx DBMS_XMLSave.ctxType;
  rows NUMBER;
BEGIN
  -- neuen Kontext anlegen und dabei den Namen der
  -- Tabelle als Parameter übergeben
  insCtx := DBMS_XMLSave.newContext('auto');
  -- Definition des Elementnamens, der den Inhalt
  -- einer jeden Tabellenzeile enthält
  DBMS_XMLSave.setRowTag(insCtx, 'Auto');
  -- Insert in Tabelle auto ausführen - zurückgegeben
  -- wird die Anzahl der eingefügten Zeilen
  rows := DBMS_XMLSave.insertXML(insCtx, pXmlDoc);
  -- Kontext schließen
  DBMS_XMLSave.closeContext(insCtx);
END;
/
```

Abb. 7: Insert über XSU.

nehmen und *v\_parser* ist für die Parser-Instanz vorgesehen.

Um Fehler, die beim Parsen auftreten können, abzufangen, wird die Exception *parserError* definiert und auf die Fehlernummer -20100 abgebildet. Innerhalb der Prozedur erzeugen wir mit der Funktion *newParser* eine neue Parser-Instanz.

Zunächst parsen wir das XML-Dokument, welches der Prozedur übergeben wurde, mit der Funktion *parseCLOB*, der als Parameter die Parser-Instanz und das XML-Dokument übergeben werden. Anschließend wird die Parser-Instanz wieder freigegeben. Ist innerhalb dieses Vorgangs keine Exception aufgetreten, ist das Parsen fehlerfrei durchgelaufen und das Dokument somit wohlgeformt.

Also setzen wir unseren Rückgabeparameter auf *true*. Falls ein *parserError* auftritt wird dieser im Exception-Block abgefangen und zunächst die Parser-Instanz wieder freigegeben. Anschließend setzen wir den Rückgabeparameter auf *false*.

### XML SQL Utility (XSU)

Die XML SQL Utility (XSU) stellt seit Oracle 8i eine Möglichkeit dar, XML-Daten mit Oracle zu verwalten bzw. XML-Daten aus der Oracle-Datenbank zu generieren. Dabei kann XSU mit Daten aus Tabellen oder Views XML-Dokumente generieren.

Des Weiteren kann XSU Daten aus XML-Dokumenten extrahieren, um sie in Tabellen oder Views einzufügen. Auch das Extrahieren von Daten aus XML-Dokumenten für das Updaten oder Löschen von Zeilen in Tabellen oder Views steht zur Verfügung.

Außerdem sind seit Oracle 9i noch weitere Funktionen hinzugekommen, wie z. B. das Generieren eines XML-Schemas oder die Unterstützung von XML-Attributen.

XSU kann auf drei verschiedene Weisen verwendet werden:

- Mit Hilfe eines kommandozeilenorientierten Java-Tools
- Mit Hilfe von PL/SQL-Aufrufen
- Mit einem Tool innerhalb des SQL-Worksheets

Im Folgenden schauen wir uns an, wie XSU innerhalb von PL/SQL eingesetzt werden kann.

Oracle stellt zwei Pakete für die Verwendung von XSU in der Datenbank zur Verfügung:

```

DECLARE
myChar VARCHAR2(2000);
BEGIN
MyChar := '<?xml version="1.0" ?>
<Autoliste>
<Auto>
  <Auto_ID>7</Auto_ID>
  <Auto_Name>
    Jaguar S-Type
  </Auto_Name>
</Auto>
</Autoliste>';
insert_Auto(myChar);
END;
/

```

Abb. 8: Aufruf der Insert-Prozedur für den Insert über XSU.

Mit dem ersten Paket, *DBMS\_XMLQuery*, hat man die Möglichkeit, eine XML-Ausgabe zu erzeugen. Die Verarbeitung wird in Abbildung 5 dargestellt.

Das Vorgehen ist hier ähnlich der Verwendung des Pakets *DBMS\_XMLGEN*. Auch die Funktionen, die im Paket *DBMS\_XMLQuery* zur Verfügung stehen, ähneln denen aus dem Paket *DBMS\_XMLGEN*. Das Beispiel bezieht sich wieder auf die Tabelle *auto*, die unter *DBMS\_XMLGEN* beschrieben wurde.

Auch beim zweiten Paket, *DBMS\_XMLQuery*, erzeugen wir zunächst einen Context-Handle, der die Abfrage enthält. Das geschieht über die Funktion *newContext* des Pakets. Dann erzeugen wir unser Ergebnis, indem wir die Funktion *getXML* aufrufen und ihr den Context-Handle übergeben. Dieses Ergebnis speichern wir in die Tabelle *temp\_clob*, um uns später durch einfaches Selektieren die Ausgabe des Ergebnisses zu vereinfachen.

Wenn wir diese Prozedur nun ausführen und das Ergebnis, welches in der Tabelle *temp\_clob* abgespeichert wurde, selektieren, bekommen wir die Ausgabe, die in Abbildung 6 dargestellt ist.

Beim Betrachten dieses Ergebnisses sehen wir, dass das Root-Element *ROWSET* und jedes Element, welches eine Ergebniszeile repräsentiert, *ROW* heißt. Dies sind die Default-Tags, wenn man die Tags für das *RowSet* bzw. das *RowSetTag* nicht, wie in unserem Beispiel zu *DBMS\_XMLGEN*, definiert hat. Dies ist aber natürlich auch über Funktionen des Pakets *DBMS\_XMLQuery* möglich. Auf diese Funktionen sind wir hier nicht eingegangen.

Mit *DBMS\_XMLSave* kann man mittels einer XML-Eingabe einfügen, updaten und löschen. Dabei ist ganz besonders darauf zu achten,

## Glossar

- CLOB** CLOB (Character Large Object) ist der (Oracle) Datentyp für ein Datenbankfeld zur Speicherung von großen Textdaten (bis zu 4 GB).
- DTD** Die Dokumenttypdefinition (englisch: „Document Type Definition“) ist eine Deklaration in SGML- und XML-Dokumenten, die die Struktur eines solchen Dokuments festlegt.
- Parser** Programme oder Programmteile, die XML-Daten auslesen, interpretieren und gegebenenfalls auf Gültigkeit prüfen.
- Oracle Text** Oracle Text ist ein Bestandteil von Oracle 9i und 10g. Oracle Text bietet umfassende Möglichkeiten der Textverarbeitung. So z. B. Suche (Volltext-, Kontext-, Ranking usw.), Filterung, Klassifikation usw. Oracle Text unterstützt verschiedene Datenformate (HTML, XML, PDF usw.)
- XML** Die Extensible Markup Language (XML) ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur.
- XML Type** Ab Oracle 9i steht XMLType als neuer Datentyp zur Verfügung. Durch diesen Datentyp werden SQL-Operationen auf XML-Inhalte genauso ermöglicht wie XML-Operationen auf SQL-Inhalte. Oracle stellt spezielle Funktionen zum Erzeugen, Extrahieren und Indizieren von XML-Daten, welche in XMLType-Strukturen liegen, zur Verfügung.

dass die Zieltabelle mit der XML-Struktur übereinstimmt, da es sonst bei der Verarbeitung zu Fehlern kommt. Die Zuordnung der Spalten zu den Elementen erfolgt über die Metadaten. Sprich: die Tags, die die einzufügenden Daten enthalten, müssen genauso heißen, wie die Spalten in der Datenbank, in denen sie abgespeichert werden sollen.

Unser Beispiel in Abbildung 7 beinhaltet den Quelltext der Prozedur *insert\_auto*. Hierbei ist zu beachten, dass die Prozedur ein XML-Dokument in Form eines VARCHARs übergeben bekommt. Die weitere Verarbeitung ist analog zur Verarbeitung von *DBMS\_XMLQuery*.

Aufgerufen wird diese Prozedur über eine anonyme Prozedur, die das zu übergebene XML-Dokument definiert (siehe Abbildung 8).

## Fazit

Oracle stellt eine stolze Anzahl von Komponenten zur Verfügung, die der Verwaltung und der Generierung von XML-Daten dienen. Die Prozeduren, die Oracle zum Generieren von XML-Dokumenten zur Verfügung stellt, stellen eine enorme Arbeitserleichterung dar, wenn man aus Daten der Datenbank XML-Dokumente erzeugen muss.

Mit Hilfe des Datentyps XMLType bieten sich viele weitere Möglichkeiten, die wir in einer der nächsten ORDIX News betrachten werden.

Auch das Abspeichern und Verwalten von XML-Daten in der Datenbank ist dank der von Oracle zur Verfügung gestellten Mittel um einiges performanter geworden. Abschließend lässt sich also sagen, dass Oracle mit dem XML-Support einen großen Teil zu einer einfachen Verarbeitung von XML in der Datenbank beigetragen hat.

Kathrin Hammerschmidt ([info@ordix.de](mailto:info@ordix.de)).