



Stefan Kinnen
Leiter der Development
Community

Liebe Mitglieder der Deutschen ORACLE-Anwendergruppe,
liebe Leserinnen und Leser,

da hat sich der DOAG-Vorstand genau das Richtige überlegt: Die Aufteilung der inhaltlichen Arbeit in vier Communities war eine gute Entscheidung. Im Februar konnte das Leitungsteam der Development Community ideal die fachliche Kompetenz der sieben SIG-Leiter in zielorientierte und inhaltlich abgestimmte Arbeit koordinieren.

Die erste Fachkonferenz DOAG 2012 BI rückt nun schnell näher. Die Qualität des Programms und die Tatsache, dass die zehn Ausstellungsplätze extrem schnell vergeben waren, sorgen für Vorfreude auf den 18. April 2012 in München.

Bei der neuen, eintägigen Fachkonferenz DOAG 2012 Development am 14. Juni in Bonn werden wir dann diese Frage beantworten: „Software-Entwicklung auf Basis von Oracle Tools und Technologien – Wohin geht die Reise?“ Dabei führen wir den zu Jahresbeginn in Berlin aufgenommenen Schwung der Sub-Community ‚Mobile Computing‘ weiter und schauen uns natürlich die Neuigkeiten in den beiden wichtigsten Technologien Apex und ADF an.

Das Schwerpunktthema dieser Ausgabe ist Apex. Joel Kallman hat als Produktverantwortlicher der Oracle Corporation im Rahmen der DOAG 2011 Konferenz + Ausstellung bereits im November viele tiefe Einblicke in die Produkt-Entwicklung, die Positionierung im Oracle Stack, die Möglichkeiten des Cloud und Mobile Computing sowie in die weitere Roadmap für 2012 gegeben. Die Artikel in diesem Heft zeigen deutlich, dass wir in der deutschsprachigen Apex-Community auf höchstem Niveau unterwegs sind.

In diesem Sinne wünsche ich Ihnen viel Spaß beim Lesen und Erproben der neu gewonnenen Erkenntnisse.

Ihr



Schneller zum Wesentlichen!

Einfach, verständlich, vollständig: Mit HUNKLER machen Sie Business Intelligence vom Start weg richtig.

- Integrierte, optimal abgestimmte Komplettlösungen für jeden Bedarf
- Zielgruppengenaue Reportvorlagen
- Robuste Technologiebasis (z. B. Oracle BI Server, Oracle Data Integrator)
- Stark verkürzte Projektzeiten
- Flexibel, skalierbar, investitionssicher
- Spezielle Lösung für SAP R/3
- Kooperation mit SAP-Spezialist NewFrontiers (www.newfrontiers.com)

ORACLE Platinum Partner

Partner von
NewFrontiers
10 Years

Best Solutions Based on Oracle
HUNKLER
GmbH & Co. KG

Hauptsitz Karlsruhe
Geschäftsstelle Bodensee

Bannwaldallee 32
Fritz-Reichle-Ring 2

76185 Karlsruhe
78315 Radolfzell

Tel. 0721-490 16-0
Tel. 07732-939 14-00

Fax 0721-490 16-29
Fax 07732-939 14-04

info@hunkler.de
www.hunkler.de

- 3 Editorial
- 5 Spotlight
- 6 „Aus Sicht von Oracle ist das bestimmt sehr sinnvoll und nützlich ...“
Interview mit Ingo Schwarzer, Chief Technology Officer der DB Systel GmbH

Apex

- 9 Apex – das Framework in der Datenbank
Prof. Dr. Horst Heineck
- 16 Mehrsprachige Apex-Anwendungen mit Hilfe von dynamischen Übersetzungen aus der Datenbank heraus
Christian Piasecki
- 20 Standardisierung mit Apex
Thomas Zielbauer
- 25 Erstellung eines Apex Plug-ins
Tobias Arnhold
- 29 Versteckte Juwelen: kaum bekannte Apex-Features
Carsten Czarski
- 35 Komfortable LOVs für Apex
Michael Schmid
- 39 Einsatz von Dropdown-Menüs in Apex-Anwendungen
Artur Rollow

- 43 SSO zwischen Eclipse RCP und Apex: Vertrauen auf den ersten Blick
Anton Schlegel
- 46 Sicherheitsaspekte einer Apex-Installation
André Schulte
- 51 Layout und dynamische Elemente für Apex-Anwendungen
Markus Dötsch
- 57 Apex für mobile Clients
Peter Raganitsch

Tipps und Tricks

- 62 Heute: Toolbar, Menüs und Hotkeys
Gerd Volberg

Aus der DOAG

- 34 Inserentenverzeichnis
- 61 Wir begrüßen unsere neuen Mitglieder
- 62 Impressum
- 63 Neuigkeiten aus dem Verein
- 66 Vorschau
- 66 DOAG-Termine

DOAG Veranstaltungsempfehlung für Apex-Entwickler:

- **18./19. April 2012**
Berliner Expertenseminar mit Peter Raganitsch „Oracle Apex mit Plug-ins erweitern“
Dieser Workshop zeigt was mit Plug-ins alles möglich ist, welche Funktionen Ihre Oberfläche auf einmal anbieten kann. Der Trainer, Peter Raganitsch, ist jahrelang erfahrener Apex-Entwickler und Autor zahlreicher Plug-ins.
- **14. Juni 2012**
DOAG 2012 Development – Fachkonferenz für Entwickler
Das Motto: „Software-Entwicklung auf Basis von Oracle Tools und Technologien – Wohin geht die Reise?“ Die Konferenz bietet vier parallele Streams, davon ein ganztägiger Stream zum Thema Apex.
- **26. September 2012**
SIG Development. Alles Apex oder was – Cloud Computing im Oracle Umfeld
Aufgrund der Vielzahl der eingereichten Vorträge zur Development Konferenz, die nicht alle Berücksichtigung finden konnten, plant die DOAG eine Zusatz-Veranstaltung zum Thema Apex im süd-deutschen Raum.

Näheres auf www.doag.org



Interview mit Ingo Schwarzer, Chief Technology Officer der DB Systel GmbH



Mit Apex lassen sich schnell und sicher Anwendungen entwickeln, Seite 9



Anforderungen mobiler Anwendungen an Apex und deren technische Umsetzung, Seite 57



Mittwoch, 8. Februar 2012

Am Rande des Interviews mit der DOAG News (siehe Seite 6) betont Ingo Schwarzer, Chief Technology Officer der Deutschen Bahn System GmbH, seine Multi-Vendor-Strategie. Er versucht den Zukäufen von Oracle gegenzusteuern, um nicht nur von einem einzigen Anbieter abhängig zu sein.

Donnerstag, 9. Februar 2012

Die diesjährige Beiratssitzung findet mit der Rekordbeteiligung von mehr als 60 DOAG-Aktiven in der Nähe von Potsdam statt (siehe Seite 63). Nach der traditionellen Präsentation der Zahlen aus dem vergangenen Jahr folgt ein besonderer Höhepunkt. Der Gastreferent Urs Meier, ehemaliger FIFA-Schiedsrichter und Management-Trainer, zeigt Parallelitäten zwischen dem Profifußball und dem IT-Business auf.

Freitag, 10. Februar 2012

Die DOAG-Communities „Datenbanken“, „Development“, „Infrastruktur/Middleware“ und „Business Solutions“ tagen auf der Beiratssitzung. Es werden unterschiedlichste Themen intensiv diskutiert und erste Maßnahmen beschlossen.

Montag, 13. Februar 2012

Fried Saacke, Vorstand und Geschäftsführer der DOAG, und Wolfgang Taschner, Chefredakteur der DOAG News, treffen sich mit einigen Microsoft-Verantwortlichen in deren Deutschland-Zentrale, um über eine Zusammenarbeit im Interessenverbund der Java User Groups e.V. (iJUG), bei dem auch die DOAG Mitglied ist, zu sprechen. Fried Saacke und Wolfgang Taschner sind sehr überrascht von den umfangreichen Java-Aktivitäten bei Microsoft. Im Laufe des Gesprächs entsteht der Eindruck, als wolle sich Microsoft stärker als ihr IT-Konkurrent IBM beim iJUG engagieren.

Montag, 20. Februar 2012

Fried Saacke besucht Richard Frammelsberger von Oracle University. Ein wichtiger Punkt ist die Aktivität von Oracle University im Rahmen der DOAG 2012 Konferenz + Ausstellung. Richard Frammelsberger sagt zu, dass Oracle University in diesem Jahr am Schultag ein besonderes Highlight präsentieren wird, um damit auch die teilweise enttäuschten Teilnehmer der Oracle-University-Schulung auf der letztjährigen Konferenz wieder zu überzeugen.

Freitag, 24. Februar 2012

Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und Fried Saacke, Vorstand und Geschäftsführer der DOAG, sind beim Notar, um die Änderungen der letzten Vorstandswahlen im Vereinsregister eintragen zu lassen.

Freitag, 2. März 2012

Die Teilnehmer der DOAG-Leitungssitzung besprechen Details zur diesjährigen DOAG 2012 Konferenz + Ausstellung sowie zum Jubiläum „25 Jahre DOAG“.

Donnerstag, 15. März 2012

Fried Saacke trifft sich in Wien mit Alexander Weichselberger, dem Vorsitzenden der Austrian Oracle User Group (AOUG), um über eine Zusammenarbeit mit der DOAG zu reden.

Samstag, 31. März 2012

Zum Frühbucherschluss zeichnet sich ein guter Erfolg für die DOAG 2012 Applications ab, die vom 8. bis 10. Mai 2012 in Berlin stattfinden wird.



Fried Saacke (links) und Ingo Schwarzer

Ein so großes Unternehmen wie die Deutsche Bahn braucht eine entsprechend funktionstüchtige IT. Fried Saacke, Vorstand und Geschäftsführer der DOAG, und Wolfgang Taschner, Chefredakteur der DOAG News, sprachen darüber mit Ingo Schwarzer, Chief Technology Officer der DB Systel GmbH.

„Aus Sicht von Oracle ist das bestimmt sehr sinnvoll und nützlich ...“

Was sind die größten IT-Herausforderungen bei der DB Systel GmbH?

Schwarzer: DB Systel ist der ICT-Service-Provider der Deutschen Bahn. Wir haben unter anderem ein System zur interaktiven Fahrplankonstruktion sowie das Reisenden-Informationssystem entwickelt. Außerdem verwaltet und betreibt DB Systel für den DB-Konzern ein konzernweites Bürokommunikationssystem mit 74.000 Nutzern und stellt die dafür erforderliche Software bereit. Die größten Herausforderungen sind Hochverfügbarkeit und Robustheit der Systeme.

Wie lösen Sie diese Herausforderungen?

Schwarzer: DB Systel verfügt über eine moderne ICT-Infrastruktur. So unterhalten wir hier am Standort Berlin zwei Rechenzentren mit rund 3.400 Servern unterschiedlicher Größe und Bauart. Darüber hinaus betreiben wir ein Datennetz mit rund 300.000 IP-Anschlüssen auf Basis unterschiedlichster Technologien von DSL bis hin zu Breitband-Glasfaser.

Gab es gravierende Störungen im Bahnnetz, für die Ihre Einheit verantwortlich ist?

Schwarzer: Wir sind nicht für das operative Bahnnetz verantwortlich, da wir keine Signale oder Weichen stellen. Es gab allerdings auch schon bei uns Störungen mit erheblichen Auswirkungen.

Haben Sie Ihr Ziel bereits erreicht, einen lückenlosen Service für die Deutsche Bahn zu bieten?

Schwarzer: Wir haben mittlerweile eine hohe, ITIL-konforme Service-Qualität erreicht und sind CMMI-Level-dreizehntertig. Es gibt natürlich immer Dinge, die man verbessern kann.

Wie garantieren Sie die Hochverfügbarkeit Ihrer Systeme?

Schwarzer: Das Ganze basiert unter anderem auf Clustersystemen, und wir haben Parallel-Server im Einsatz.

Welche Rolle spielt dabei Oracle?

Schwarzer: Wir haben sehr viele Datenbanken im Einsatz. Durch die ver-

schiedenen Zukäufe sind auch unsere Siebel- und PeopleSoft-Lösungen mittlerweile von Oracle.

Wie schaffen Sie es, die Systeme zu patchen, ohne die Hochverfügbarkeit zu beeinträchtigen?

Schwarzer: Erst einmal erwarten wir von unseren Herstellern, dass man während des laufenden Betriebs patchen kann. Ansonsten gibt es bei der Bahn hin und wieder abgestimmte Möglichkeiten, um für ein oder zwei Stunden solche Updates fahren zu können. Häufige Patches stellen natürlich ein großes Problem dar.

Wie zufrieden sind Sie in dieser Hinsicht mit Oracle?

Schwarzer: Es hat sich schon einiges gebessert, sodass wir mit zwei bis drei größeren Release-Änderungen im Jahr hinkommen. Wir betreiben zudem unsere Datenbank-Releases sehr lange, weil die Umstellung auf eine neue Version nicht ohne großen Migrations- und Test-Aufwand möglich ist. Jeder Patch, der bei uns eingespielt wird,

muss zuvor unter Live-Bedingungen getestet werden.

Haben Sie sich auch mit anderen Oracle-Produkten wie der Middleware oder mit Managed Systems wie Exadata beschäftigt?

Schwarzer: Wir schauen in erster Linie darauf, ob die Produkte in unsere IT-Landschaft passen. Das Prinzip Exadata gibt es ja von verschiedenen Herstellern. Wir beobachten natürlich, was Oracle macht, sind aber kein Kunde der angesprochenen Produkte.

Wie beurteilen Sie die Zukauf-Strategie von Oracle?

Schwarzer: Aus Sicht von Oracle ist das bestimmt sehr sinnvoll und nützlich. Das Problem für den Endkunden ist die immer größer werdende Abhängigkeit von einem Hersteller. Aufgrund der Zukäufe ist zwar ein besseres Zusammenspiel der Produkte zu erwarten, für mich ist allerdings die damit verbundene Abhängigkeit kritisch.

Können Sie sich vorstellen, ein Komplettsystem von der Hardware bis zu den Applikationen von einem einzigen Hersteller wie Oracle einzusetzen?

Schwarzer: Nein, unter keinen Umständen. Für uns gibt es für alle kri-

tischen Bereiche einen „second vendor“. Je größer die Abhängigkeit von einem Hersteller, desto geringer ist der Spielraum bei Lizenz-Verhandlungen.

Welche Erwartungen haben Sie an Oracle?

Schwarzer: Ich wünsche mir einen besseren Support der Open-Source-Lösungen. Auch die Schnittstellen der einzelnen Module sollten transparenter beschrieben sein. Außerdem ist das Auspielen der Marktmacht für mich nicht gerade vertrauensbildend. Hinzu kommt, dass die amerikanische Zentrale oft die lokalen Gegebenheiten in Europa nicht sonderlich berücksichtigt.

Wie hat sich aus Ihrer Sicht die Support-Qualität von Oracle in den letzten Jahren entwickelt?

Schwarzer: Abgesehen von einigen schweren Fehlern gab es bei uns kaum Probleme. Es stört mich allerdings, dass wir für alte Datenbank-Versionen gesonderte Wartungsverträge abschließen müssen, da die Migration auf aktuelle Releases zu aufwendig ist.

Was hoch ist der Anteil individuell entwickelter Software in Ihrem Haus?

Schwarzer: Viele technische Abläufe bei der Deutschen Bahn müssen wir natürlich individuell entwickeln, weil es keine entsprechenden Standard-Lösungen gibt. Dafür steht ein großer Entwicklungsbereich bereit. Im kaufmännischen und im Office-Bereich hingegen setzen wir nur Standard-Software ein. Auch die Datenbanken nutzen wir möglichst ohne große individuelle Anpassungen, um Aufwand bei der Migration zu vermeiden.

Welchen Stellenwert hat Java bei der Entwicklung individueller Software?

Schwarzer: Java ist seit vielen Jahren als Standard gesetzt.

Was ist in Ihnen vorgegangen, als Oracle Sun übernommen hat?

Schwarzer: Anfangs war ich nicht sonderlich positiv davon angetan, da es

Libelle SystemCopy



- Ohne in Ihre SAP-Umgebung einzugreifen bzw. diese zu verändern
- Ohne aufwändige Vorplanung
- Mit minimaler Durchlaufzeit
- Bei gleichbleibender Qualität der Kopie

... mit deutlich reduzierten Prozesskosten



Hans-Joachim Krüger
Chief Technology Officer
Libelle AG

Erfahren Sie mehr:
www.Libelle.com/systemcopy



Libelle

Libelle AG
Gewerbestr. 42 • 70565 Stuttgart, Germany
T +49 711 / 78335-0 • F +49 711 / 78335-148
www.Libelle.com • sales@libelle.com





bei uns eine sehr lange Open-Source-Tradition gibt. Mittlerweile sehe ich jedoch, dass Oracle sich bemüht, die Sprache Java weiterzuentwickeln.

Wie sind Ihre Erfahrungen hinsichtlich der ehemaligen Sun-Hardware?

Schwarzer: Wir haben Sun-Server im Einsatz. Ich bin gespannt, wie lange Oracle noch im Hardware-Geschäft bleiben wird.

Zur Person: Ingo Schwarzer

Ingo Schwarzer (Jahrgang 1959) ist Informatiker und hat seinen Lebensmittelpunkt in Berlin. Er begann seine Laufbahn im Rechenzentrums-Betrieb und hatte dort diverse Leitungsfunktionen wie Schichtleiter inne. Weitere Stationen waren Leiter IT/TK-Planung, Leiter Lizenzmanagement, Fachbereichsleiter Architektur und Tools sowie Infrastrukturplanung. Ingo Schwarzer ist seit 2009 CTO der DB Systel.

In welche Richtung wird sich Ihre IT in den kommenden Jahren entwickeln?

Schwarzer: Grundsätzlich läuft bei uns alles in die Richtung „...as a Service“. Zudem würde ich gerne noch etwas mutiger Open-Source-Lösungen einsetzen.

Wie kann Oracle Sie hier noch besser unterstützen?

Schwarzer: Hier gibt es keine speziellen Wünsche. Wichtig ist, dass die Dinge, die heute einigermaßen stabil laufen, dies auch in Zukunft tun.

Wie beurteilen Sie die Lizenzpolitik von Oracle hinsichtlich Virtualisierung?

Schwarzer: Oracle nutzt hier seine Monopol-Stellung eiskalt aus, ich halte das nicht für akzeptabel.

Wie stehen Sie zu der von Oracle kürzlich groß angekündigten eigenen Cloud-Strategie?

Schwarzer: Eine Public Cloud hilft uns in keiner Weise weiter, weil die erforderliche Datensicherheit nicht gegeben ist – insbesondere wenn diese in den USA gehostet wird. Zudem müsste eine solche Cloud die Anforderungen der Bahn hinsichtlich des Service-Level-Agreements erfüllen.

Wie sehen Sie den Stellenwert einer Anwendergruppe wie der DOAG?

Schwarzer: Die DOAG ist für mich ein gutes Sprachrohr gegenüber Oracle, weil sie die Probleme der Interessen der Anwender aufnimmt und gebündelt kommuniziert. Auch als Plattform für den Erfahrungsaustausch leistet die DOAG eine wichtige Arbeit. Ich würde mir wünschen, dass die DOAG über den Oracle-Horizont hinausblickt und auch Produkte anderer Hersteller ins Spiel bringt, sofern diese sinnvoller als eine Oracle-Lösung sind.

Firmenporträt: DB Systel GmbH

Die DB Systel GmbH mit Sitz in Frankfurt am Main ist mit rund 2.900 Mitarbeitern einer der führenden Anbieter von ICT-Lösungen in Deutschland. Als Tochterunternehmen der Deutschen Bahn deckt das Unternehmen dabei den gesamten Lebenszyklus von ICT-Lösungen ab, angefangen von der fachlichen Analyse über die strategische Planung und Entwicklung bis hin zu Betrieb und Optimierung. Daten und Fakten auf einen Blick:

- ICT aus einer Hand: Beratung, Planung, Entwicklung, Betrieb, Service
- Rund 2.900 Mitarbeiter (2010)
- Rund 613 Millionen Euro Umsatz (2010)
- Zwei Rechenzentren mit rund 3.400 Servern
- Vier Netzwerkleitstände
- 1,4 Petabyte Plattenspeicher/4 Petabyte Backup-Kapazität
- Rund 500 produktive IT-Anwendungen
- Datennetz mit rund 300.000 IP-Anschlüssen von DSL bis Breitband-Glasfaser
- Betrieb des konzernweiten Bürokommunikationssystems mit 74.000 Nutzern
- TK-Anlagen-Netz mit 115.000 Teilnehmeranschlüssen
- ICT-Projekte, -Anwendungsentwicklung, -Serviceleistungen, Rechenzentrumsbetrieb, Desktop Services inkl. Managementleistungen zertifiziert nach ISO 9001
- IT-Servicemanagement zertifiziert nach ISO 20000 (ITIL)
- IT-Sicherheitsmanagement zertifiziert nach ISO 27001
- Gesamte Anwendungsentwicklung zertifiziert nach CMMI Level 3

Application Express ist ein Tool für die schnelle Entwicklung von Web-Anwendungen. Es ermöglicht Ihnen die Bereitstellung von Daten für eine gemeinsame Nutzung sowie die Erstellung von Anwendungen. Auch ohne tiefgehende Programmierkenntnisse und nur mit einem Web-Browser ausgestattet lassen sich Anwendungen entwickeln und bereitstellen, die schnell und sicher sind [1].

Apex – das Framework in der Datenbank

Prof. Dr. Horst Heineck, Hochschule Hof

Mit dem Framework Oracle Application Express, kurz Apex, hat Oracle ein Entwicklungstool am Markt, das in seiner Art einzigartig ist. Der Anwender – Administrator, Entwickler oder Nutzer – greift auf Apex über seinen Web-Browser zu. Apex selbst ist vollständig in die Datenbank integriert und benötigt keine weitere Installation von Software. Einzige Ausnahme bildet die Installation eines Webservers wie der „Oracle WebLogic Server“, was aber nicht immer notwendig ist, da die Datenbank von Haus aus einen Webserver mitbringt.

Die erste Version von „HTML DB 1.5“, wie Apex früher bezeichnet wurde, erschien im Jahr 2004. 2011 wurde die bisher letzte Version „Apex 4.1“ freigegeben. Ursprünglich war das Framework dazu konzipiert worden, den Wildwuchs an Excel- und Access-Dateien in Firmen zu reduzieren. Jeder kennt den weiteren Lauf einer Excel-Datei, die an einen Kollegen weiter-

gegeben wird. Die Kopie entwickelt ein Eigenleben und wird in eine andere Richtung ergänzt, als ursprünglich vorgesehen war. Für eine Firma ist es äußerst nachteilig, wenn die wichtigen Unternehmensdaten über mehrere Excel-Dateien verteilt im Unternehmen kursieren, die untereinander strukturell verschieden aufbereitet die Daten beinhalten. In solchen Fällen ist eine zentrale Haltung der Daten existenziell. Um das zu gewährleisten, ist es notwendig, die Vorteile einer Einzelplatz-Lösung und die einer zentralen Datenbank zu verbinden [2]. Diesem Umstand entsprechend ist es mit Apex sehr einfach, Excel-Dateien im CSV-Format zu importieren und eine Applikation daraus entstehen zu lassen. Apex hat für diese und viele weitere Aufgaben eine Vielzahl von Assistenten, die den Entwickler exzellent dabei unterstützen. Gleichmaßen können Berichte aus den Anwendungen für Excel im CVS-Format exportiert werden.

Interaktive Berichte in Apex

Eine geniale Erweiterung von Apex stellen die interaktiven Berichte dar. Bei anderen Anwendungen wird das Aussehen eines Berichts durch den Entwickler vorgegeben. Die interaktiven Berichte in Apex zeichnen sich dadurch aus, dass auch der Nutzer seine Berichte nach seinen Bedürfnissen anpassen kann. Wie in Abbildung 2 zu sehen ist, sind verschiedene Anpassungen möglich. So zum Beispiel:

- Die Projektion des Berichts, also die Auswahl der anzuzeigenden Spalten
- Die Selektion einzelner Zeilen der Tabelle, die in dem Bericht angezeigt werden sollen
- Die Anzahl der darzustellenden Zeilen im Bericht
- Die verschiedenen Formen der Formatierung, inklusive der Sortierung, Berechnung von Funktionen, Diagrammdarstellung etc.
- Der Flashback auf der Tabelle
- Die Speicherung des neuen Berichts
- Die verschiedenen Formate für das Herunterladen des Berichts

Das Framework Apex

Im Lauf der Zeit hat sich Apex zu einem Framework zur Entwicklung kleiner und mittlerer datenbanknaher Anwendungen im Oracle-Umfeld etabliert. Ein weiterer Vorteil des Frameworks ist die vollständige Handhabung webbasiert in einem Browser. Damit sind heute sogar Anwendungen denkbar, die auf Smartphone und Tablet-PC laufen können [3]. Apex ist so schon für den Einsatz im „Mobile Computing“ vorbereitet.



SPLUS Ziel	Zeitpunkt	Ampel
Beende NEUE_VERLEGUNG	25.01.2012 16.06.01,470000000	🟢
Starte NEUE_VERLEGUNG	25.01.2012 16.05.58,740000000	🔴
Stop_V Verlegungen	25.01.2012 16.05.56,860000000	🟢
Start_V Verlegungen	25.01.2012 16.00.46,670000000	🔴
Beende NEUE_VERLEGUNG	23.01.2012 16.47.11,660000000	🟢

Abbildung 1: Beispiel einer Apex-Anwendung auf einem Android-Smartphone

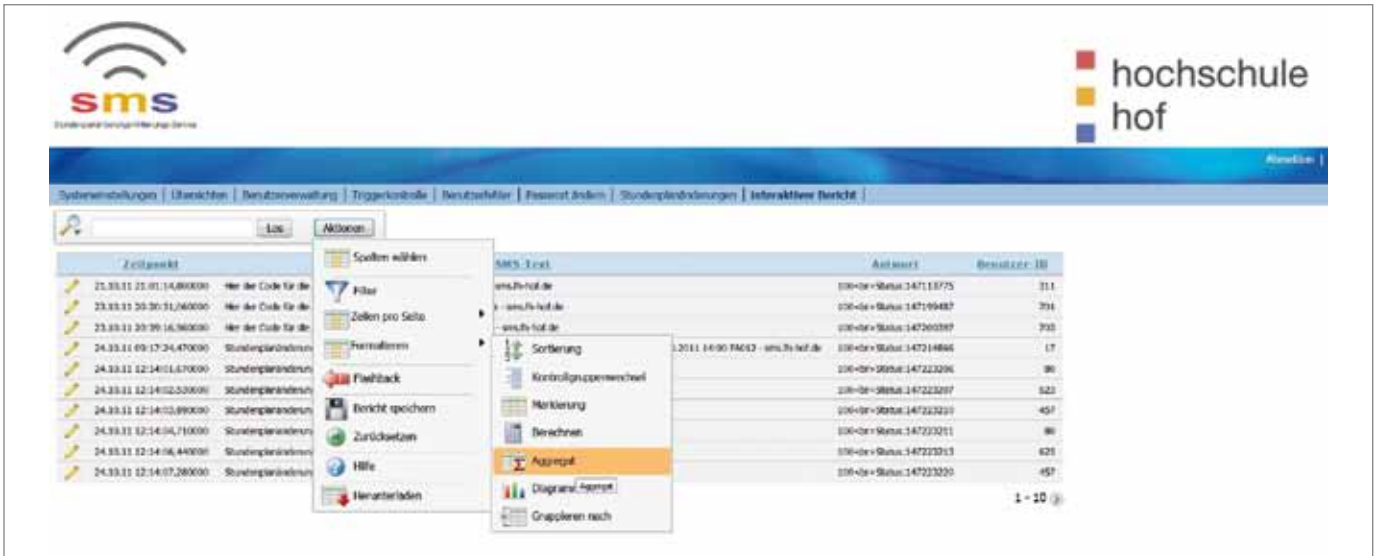


Abbildung 2: Beispiel für interaktive Berichte

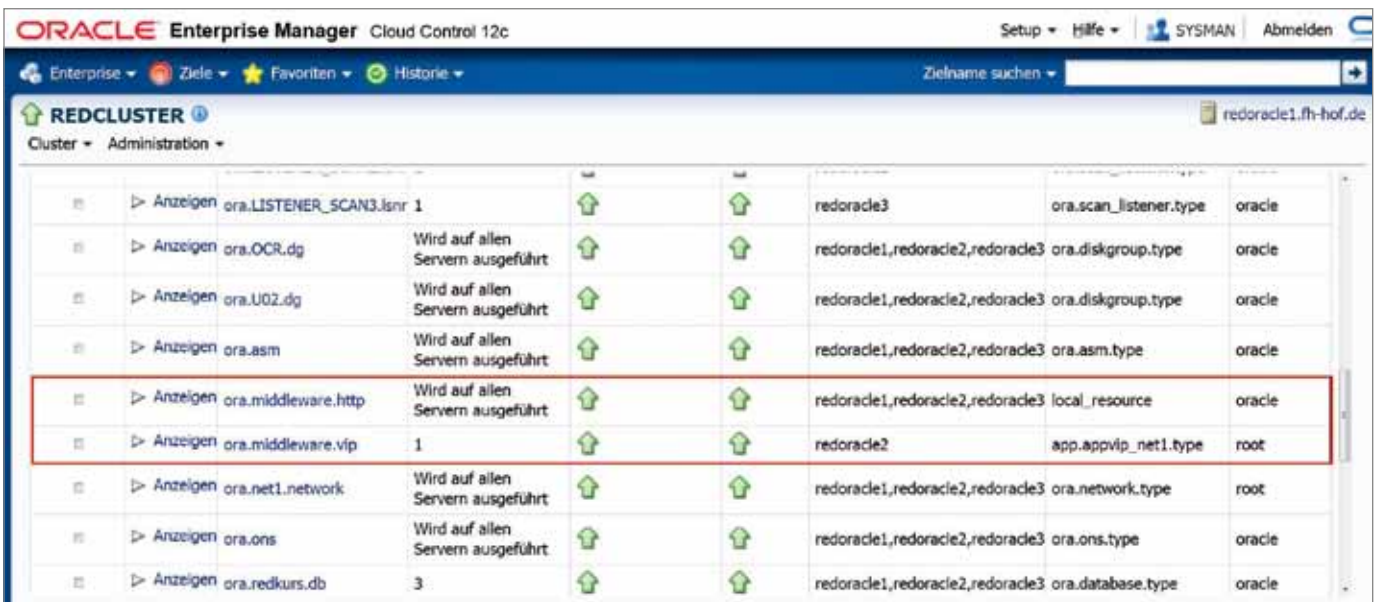


Abbildung 3: Darstellung der Cluster-Ressourcen im Enterprise Manager 12c

Anwendungen, Daten und Dateien lassen sich über die eingebauten Export-/Import-Funktionen zwischen verschiedenen Plattformen austauschen. Da Apex in jeder Oracle-Datenbank-Version ab 9i R2 enthalten und die Datenbank auf verschiedenen Plattformen verfügbar ist, kann jederzeit ein Transfer zwischen diesen vorgenommen werden. Die Anwendungen von Apex sind damit ohne Wenn und Aber plattformunabhängig. Hier zählt sich die seit langer Zeit bestehende Firmenpolitik von Oracle aus, auf verschiedenen Betriebssystem- und

Hardware-Plattformen identisch in der Funktionalität und dem Aussehen zu sein.

Apex für KMUs

Besonders für kleine und mittlere Unternehmen (KMU) hat Apex eine Chance sich durchsetzen. Grund dafür ist die Möglichkeit, auf der Oracle-Plattform eine kostengünstige Lösung maßgeschneidert für die Firma entwickeln zu können. Beispielsweise kommt dort als Betriebssystem Oracle Enterprise Linux, eventuell in der vir-

tualisierten Umgebung von VM Virtual Box, zum Einsatz. Für die Datenbank findet die Database Express Edition 11g R2 Verwendung. Bei Bedarf wird die aktuelle Version von Application Express als Upgrade eingespielt. Alle Produkte sind auf der OTN-Seite als „Free Product“ gekennzeichnet. Neben dem Know-how für die Software-Entwicklung fallen nur noch Kosten für einen handelsüblichen PC mit 64-Bit-Architektur bei der Verwendung von Enterprise Linux an.

Durch die Skalierung des Datenbank-Servers Oracle 11g R2 ergibt sich

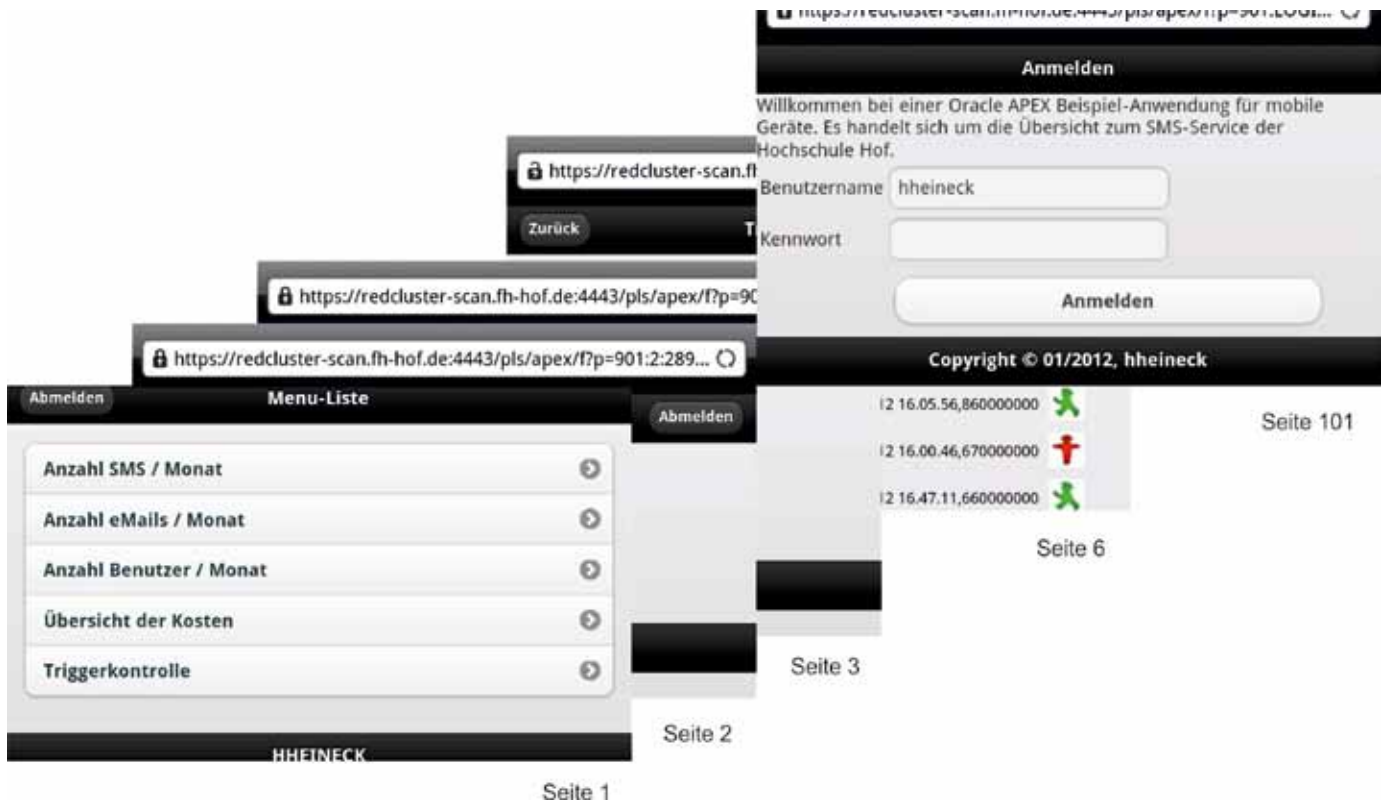


Abbildung 4: Aufbau einer Apex-Anwendung für ein Smartphone

für die Verwendung von Apex ein Einsatzfeld, ausgehend von der sehr kostengünstigen Architektur aus der genannten Darstellung bis hin zur Enterprise Edition auf einer Clustermaschine mit Oracle Real Application Clusters.

Apex im RAC-Umfeld

Wird Apex im Real-Application-Clusters-Umfeld betrieben, bietet sich natürlich sofort die Verwaltung des Webserver über die Grid-Infrastruktur, die Clusterware, an. Dabei ist es notwendig, dass zusätzliche Clusterware-Ressourcen eingerichtet sind. Für den Internet-Zugang der Nutzer auf den Webserver wird eine Anwendungs-VIP als „Cluster Resources“ eingerichtet. Diese wird in Abhängigkeit von der Netzwerk-Ressource „ora.net1.network“ erzeugt. Die Verwaltung des Webserver als Fremdsoftware, hier am Beispiel des „Oracle WebLogic Server“, wird anschließend als „Local Resources“ in Abhängigkeit von der Anwendungs-VIP realisiert. Damit ist es möglich, diese Fremdsoftware durch die Clusterware verwalten zu lassen. Es

ist aber nicht immer nötig, diese „Local Resources“ auf allen Clusterknoten zu starten. Abbildung 3 zeigt die Einbettung der genannten Ressourcen im Oracle Enterprise Manager 12c. Als Beispiel dient ein Cluster mit drei Knoten. Auf den Oracle DBA-Community-Seiten [4] gibt es Tipps, die die Vorgehensweise näher beschreiben.

Aufbau einer Apex-Anwendung

Eine Apex-Anwendung besteht genauso wie das Framework Apex selbst aus mehreren Seiten. Dabei haben zwei Seiten eine besondere Bedeutung: Objekte, die für alle Seiten der Anwendung gelten sollen, finden ihren Platz auf Seite 0. Wenn die Anwendung durch Anmeldedaten geschützt wird, erscheint beim Aufruf der Anwendung Seite 101 (siehe Abbildung 4).

Für die optische Gestaltung der Seiten sind im Repository von „Apex 4.1“ 23 verschiedene Themes zu freier Auswahl enthalten. Diese wiederum enthalten Templates für unterschiedlich Objekte auf der Seite, etwa für unterschiedliche Berichte oder Formulare, Eingabe-Elemente oder verschiedene

Buttons. Der Wechsel zwischen den einzelnen Templates ist sehr einfach möglich. Zusätzlich kann der Entwickler auch schnell eigene Themes und Templates erzeugen, um die Anwendung nach seinen eigenen oder den Firmenbedürfnissen anzupassen. So wurde nach [3] auch eine Themes-Anpassung – Änderung einzelner Templates – für mobile Geräte realisiert, wie in

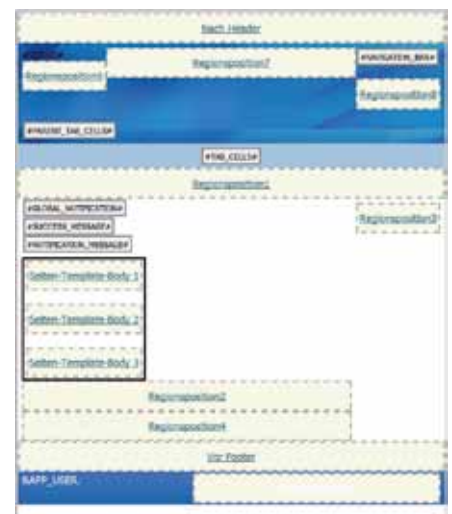


Abbildung 5: Regionen-Positionen für Themes 20 – Modern Blue

den Abbildungen 1 und 4 beispielhaft zu sehen ist.

Jede dieser Seiten ist in Regionen aufgeteilt. In Abhängigkeit von den Themes sind die Regionen unterschiedlich auf einer Seite positioniert (siehe Abbildung 5). Der Entwickler ist damit in der Lage, das Aussehen der einzelnen Seiten optimal an die Erfordernisse der Anwendung anzupassen. Es können mehrere Regionen auf derselben Position angeordnet werden. Horizontal sind maximal neun Positionen über den Parameter „Spalte“ möglich. Die Reihenfolge der einzelnen Regionen in vertikaler Ausrichtung erfolgt über den Parameter „Sequence“.

Weiterhin sind auf jeder Seite verschiedene Möglichkeiten der Navigation wie Navigationspfade, Registerkarten und Listenelemente vorhanden. Es ist auch denkbar, über Buttons mit einer hinterlegten Validierung und/oder Berechnung der Daten gleichzeitig die Navigation zu steuern.

In Abbildung 4 ist die Verwendung eines Listenelements (Seite 1) auf einem Smartphone zu sehen. Abbildung 7 stellt die Möglichkeiten der Registerkarten und Buttons dar. Die Position der Navigationspfade wurde dabei für die Copyright-Darstellung verwendet.

Neben den verschiedensten Gestaltungsmöglichkeiten, die Apex von Haus aus out-of-the-box mitbringt,



Abbildung 6: Bearbeitung der Regionsdefinition

können weitere Funktionalitäten über die Installation von Plug-ins der Anwendung hinzugefügt werden (siehe Abbildung 8). Diese lassen sich über die Seite www.apex-plugin.com bequem und kostenlos laden.

Mit Apex in die Cloud

Wenn der Begriff „Cloud Computing“ fällt, muss zunächst geklärt werden, was darunter zu verstehen ist. Die zielführende Definition ist wohl in [5] zu finden, zudem sind an gleicher Stelle fünf verschiedene Merkmale, drei verschiedene Service-Modelle und vier Deployment-Varianten aufgelistet.

Soweit die Theorie. Es stellt sich nun die Frage, was Apex an dieser Stelle leisten kann. Bekanntlich hat Apex als Basis Workspaces, in denen die verschiedensten Anwendungen zusam-

mengefasst sind. Jedem Workspace ist ein Schema in der Datenbank zugeordnet. Die Anwendungen in einem Workspace greifen in der Regel auf Daten im selben Datenbank-Schema zu. In [2] sind diese Zusammenhänge sehr gut allgemein dargestellt.

Wird zwischen der Definition von Cloud Computing und Apex ein Zusammenhang hergestellt, kann sehr schnell die Verwendung von Apex als „Entwicklungs- und Deployment-Plattformen als Service im Netzwerk“ (PaaS) in einer „Private Cloud“ abgeleitet werden. Es ist mit Apex also möglich, den verschiedenen Entwicklern in einer Firma jeweils eigene Workspaces mit allen notwendigen administrativen Rechten zur Verfügung zu stellen. Jeder Entwickler erhält so die Möglichkeit, auf einer zentral in der Datenbank gehaltenen Dateibasis sei-

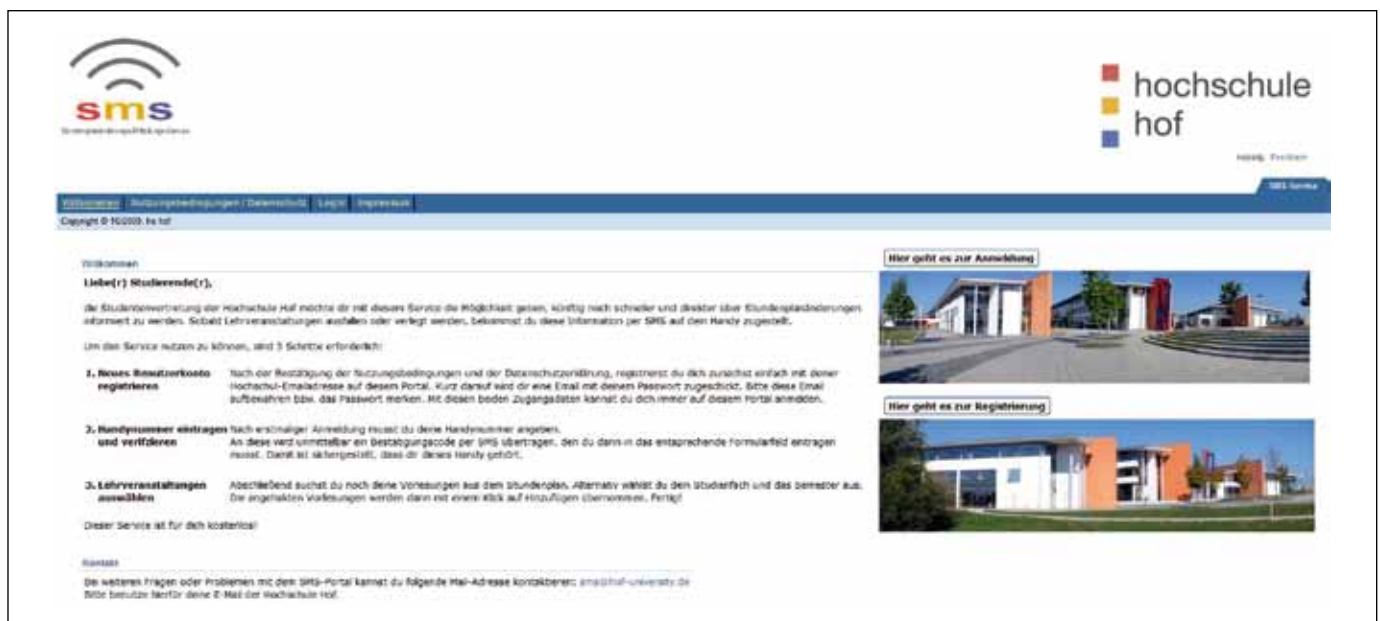


Abbildung 7: Startseite des SMS-Service der Hochschule Hof



Abbildung 8: Komponenten out-of-the-box [2]



Abbildung 9: Google Maps Plug-in [3]

ne Anwendungen zu entwickeln und den Nutzern bereitzustellen. Durch die Administration des Workspace ist ebenfalls die Verwaltung von Rechten der Nutzer möglich. In jedem Workspace ist eine komplette Nutzerverwaltung integriert.

Die Bereitstellung der Workspaces für verschiedene Entwickler ist relativ einfach durch den Apex-Administrator möglich. Besser ist jedoch eine automatische Bereitstellung der Work-

spaces, die der Entwickler jeweils selbst initiiert. Als Beispiel für einen solchen Automatismus wird in [6] ein Server für die allgemeine Nutzung durch Oracle bereitgestellt.

Da Apex vollständig „stateless“ ist, die Datenbank-Sitzung wird also nach Verarbeitung des Request wieder freigegeben und eine Datenbank-Sitzung bedient viele Apex-Sitzungen, kann mit einem relativ geringen Hardware-Aufwand eine Vielzahl von Anwendungen parallel bearbeitet werden [7]. Stand Dezember 2010 waren folgende Parameter für den Server [6] zu benennen: Es handelte sich um einen Dell PowerEdge 1950, 2 x DualCore, mit 2,33GHz und 32GByte RAM. Damit war es möglich, 4.756.868 Seitenaufrufe zu bearbeiten, was 8 Seitenaufrufen pro Sekunde für 3.958 unterschiedliche Benutzer entspricht. Zum selben Zeitpunkt waren es 11.002 Workspaces mit 37.171 Anwendungen, die der Server gehostet hat [7].

Die Anforderungen für die Bereitstellung von Apex-Workspaces in einer Firma sind absolut überschaubar. Während die Bereitstellung der Workspaces auf [6] allen allgemeinen Möglichkeiten entsprechend sehr umfangreich ist, können für eine Firma Abstriche gemacht werden. So kann beispielsweise über gültige Namensangaben die E-Mail-Adresse des Administrators des Workspace nach Vorgaben der Firma generiert werden. Für den Einsatz

High Performance für Oracle®
Entwicklungen:



Supporting
UI-Swap

devcontrol

Das intelligente Tool für den Entwicklungsprozess. Der Direct Application Impact sorgt für permanente Transparenz in der Oracle® Applikation.

Bereits während der Entwicklungsphasen wird der Source Code analysiert, versioniert und SQL Relevanzen als Metadaten gespeichert. So entstehen portable Entwicklungseinheiten, die den Entwickler von der Analyse bis zum produktiven Rollout unterstützen.

Mit dem UI-Swap unterstützt devcontrol die Migration von Oracle Forms Applikationen in neue Entwicklungswelten.

ORACLE PARTNER

imining gmbh
Berduxstraße 22
D-81245 München
Tel. +49 89/92 30 663-0
Fax +49 89/92 30 663-1
www.imining.de

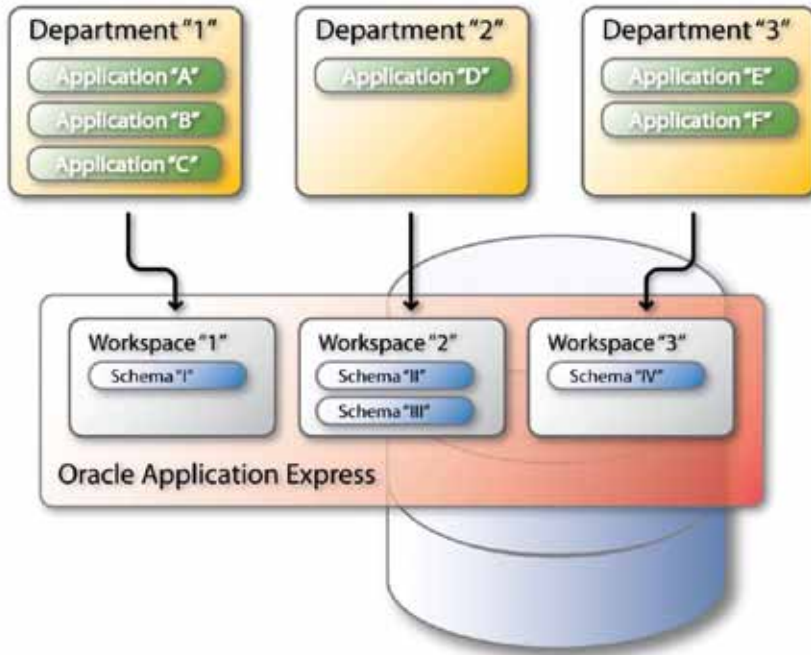


Abbildung 10: Zusammenhang zwischen Anwendung, Workspace und Datenbankschema [2]

in verschiedenen Lehrveranstaltungen an der Hochschule Hof wurde eine Apex-Anwendung geschaffen, die den Studierenden gestattet, eigene Workspaces zu generieren [8].

Fazit

Die Möglichkeit der Verwendung von Cloud Computing über Platform as a Service (PaaS) entlastet weitestgehend die Entwickler von den Verwaltungsaufgaben, die ein kommerziell anspruchsvolles Datenbanksystem wie Oracle mit sich bringt.

Die Haupthinderungsgründe scheinen in vielen Unternehmen die Unkenntnis über Apex und die Scheu vor den Kosten und dem verwaltungstechnischen Aufwand der Oracle-Produkte zu sein. Dass sich kostengünstigere Möglichkeiten ergeben, wurde jedoch dargestellt.

Im öffentlichen Dienst wie im Freistaat Bayern ist zumindest der zweite Grund flächendeckend ausgeräumt.

Quellenangabe

- [1] Infotext in Oracle Application Express, Version 4.1.0.00.32
 [2] Carsten Czarski, Schneller zum Ziel - Was kann Apex? Was ist neu?, Vortrag am 29.04.2011 im OU-Schulungszentrum München

- [3] Peter Raganitsch, Oracle Apex für mobile Endgeräte, Application Express Community, <http://www.oracle.com/webfolder/technetwork/de/community/apex/tipps/peter-raganitsch-mobile/index.html>
 [4] Sebastian Solbach, Applikationsüberwachung mit 11g R2 Grid Infrastruktur – Am Beispiel der DBConsole, DBA Community, http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/tipps/grid_dbconsole/index.html
 [5] Peter Mell Timothy Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, 2011
 [6] apex.oracle.com
 [7] Carsten Czarski, Oracle11g + Apex = Cloud Computing, Vortrag am 29.04.2011 im OU-Schulungszentrum München
 [8] <http://www.oracle.com/us/corporate/customers/customersearch/hochschule-hof-1-db-snapshot-513072.html>
 [9] Deutschsprachige Ausgabe Application Express Community, <http://www.oracle.com/webfolder/technetwork/de/community/apex/index.html>

Prof. Dr. Horst Heineck
 Hochschule Hof, Fakultät Informatik
horst.heineck@hof-university.de



Ausblick: Wie geht es weiter mit Apex?

Die aktuelle Version Apex 4.1 ist seit Sommer 2011 verfügbar. Natürlich geht die Entwicklung weiter – und das internationale Apex-Entwicklerteam ist schon mit der Arbeit an den nächsten Releases beschäftigt. Für Apex 4.2 stehen dabei folgende Themen im Vordergrund:

- Ein Schwerpunkt wird die explizite Unterstützung für Apex-Anwendungen auf mobilen Endgeräten sein. Diese wird auf dem kürzlich freigegebenen „iQuery mobile“ basieren. Apex wird bereits „Out-of-the-Box“-Themes für mobile Anwendungen mitbringen.
- Im Rahmen der Unterstützung für mobile Endgeräte sollen auch die schon lange erwarteten HTML5-Charts eingeführt werden, also Geschäftsgrafiken und Diagramme, die man auch ohne das Flash-Plug-in darstellen kann.
- Die Unterstützung für RESTful Webservices soll weiter ausgebaut werden; so ist geplant, dass sich die Webservice-Schnittstelle für SQL-Abfragen oder PL/SQL-Objekte deklarativ einrichten lässt.
- Und wie immer wird auch Apex 4.2 jede Menge Verbesserungen und Optimierungen im Detail mitbringen.

Grundlage dieser Liste ist das auf der Oracle-Webseite verfügbare „Statement of Direction“, das vom Apex-Team regelmäßig aktualisiert wird. Natürlich ist diese Liste wie immer „ohne Gewähr“ – einzelne Schwerpunkte können sich im Rahmen der Entwicklungs- und Testarbeiten nochmals ändern.

Carsten Czarski
carsten.czarski@oracle.com



Initiativ, dynamisch, anspruchsvoll.

Sie wissen, was Sie wollen: Ihre Zukunft mit viel Eigenverantwortlichkeit und Dynamik selbst gestalten. Eine hochmoderne IT-Landschaft mitprägen, an innovativen Eigenentwicklungen mitwirken und Ihren ganz persönlichen Weg gehen. Auch wir wissen, was wir wollen: Sie kennenlernen. Sie herausfordern und fördern. Sie auf Ihrem Karriereweg begleiten. Sichern Sie sich exzellente Aussichten – von Anfang an. Erfahren Sie mehr über unsere IT-Abteilung unter: www.berenberg.de/it.html

WWW.KARRIERE-BEI-BERENBERG.DE

Privatbankiers  gegründet 1590
BERENBERG BANK
Joh. Berenberg, Gossler & Co. KG

Dieser Artikel zeigt, wie man mehrsprachige Apex-Anwendungen erstellt, ohne dass für jede Sprache ein separates Exemplar der Anwendung aufgebaut sein muss. Durch Pflege der Texte in der Datenbank und eine dynamische Übersetzung zur Laufzeit lässt sich erheblicher Aufwand bei den Deployment-Prozessen und bei der Wartung von Apex-Applikationen sparen.

Mehrsprachige Apex-Anwendungen mit Hilfe von dynamischen Übersetzungen aus der Datenbank heraus

Christian Piasecki, TEAM Partner für Technologie und angewandte Methoden der Informationsverarbeitung GmbH

Die Funktion der Mehrsprachigkeit, die Apex von Hause aus mitbringt, ist im Prinzip gut gelöst. Das Verfahren, die Texte in eine Übersetzungsdatei zu exportieren und nach dem Übersetzen wieder zu importieren, ist an sich komfortabel gelöst. Diese Standard-Vorgehensweise besitzt aber einen großen Nachteil. Dadurch, dass für jede Sprache ein separates Exemplar der Anwendung erstellt wird, steigt der Aufwand bei Deployment und Wartung aller „Parallel“-Anwendungen enorm. Sobald es Änderungen an der Anwendung gibt – und mögen diese auch nur den PL/SQL-Code betreffen – sind die Anwendungen für alle unterstützten Sprachen neu zu generieren und zu deployen. Geht man hier von drei technischen Systemen pro Anwendung aus (Entwicklung, Test und Produktion) führt das zu erheblichen Mehraufwänden. Um diese zu verringern, wird hier ein alternativer Ansatz vorgestellt, der sich schon in mehreren Kundenprojekten bewährt hat.

Ziel des hier vorgestellten Ansatzes ist es, den Aufwand für die Bereit-

stellung einer mehrsprachigen Anwendung möglichst gering zu halten. Gelöst wird diese Aufgabenstellung, indem man von der Standard-Vorgehensweise abweicht und die Abwicklung der Mehrsprachigkeit in die Datenbank-Schicht verschiebt.

Kernfunktionalität in der Datenbank

Die Kernfunktionalität der Mehrsprachigkeit wird in die Datenbank ausgelagert und besteht aus einer Übersetzungstabelle und einem PL/SQL-Package. Die Übersetzungstabelle hält die Information darüber, welches in der Oberfläche darzustellende Textelement in welcher Sprache mit welchem tatsächlichen Text angezeigt wird. Das Package übernimmt die Aufgabe, diese sprachbezogenen Daten in die Übersetzungstabelle einzutragen und auszulesen.

Es bestehen nun unterschiedliche Möglichkeiten, diese Tabelle aufzubauen. Eine möglichst schlanke ist nachfolgend vorgestellt. Es reicht im Grunde aus, wenn die Übersetzungstabelle

aus drei Spalten besteht: ID, LANG, LSTRING (siehe Listing 1).

Die erste Spalte beinhaltet eine entsprechende ID, die das verwendete Textelement in der Anwendung referenziert. Die zweite Spalte wird mit einer eindeutigen Sprach-Kennung gefüllt und die dritte Spalte enthält den tatsächlichen Text in der gewählten Sprache, der dann oberflächenseitig angezeigt wird. Eine gefüllte Tabelle kann dann folgendermaßen aussehen (siehe Tabelle 1).

In dieser Tabelle werden für alle Objekte, die in der Apex-Anwendung verwendet werden, zentral die Übersetzungstexte gepflegt. Tragen unterschiedliche darzustellende Objekte die gleiche Bezeichnung, müssen die Sprach-Informationen nur einmal zentral in der zugrundeliegenden Tabelle gepflegt werden und stehen an allen zugehörigen Stellen zur Verfügung. Das Eintragen und Auslesen der Übersetzungen wird durch ein PL/SQL-Package realisiert, das im Wesentlichen aus zwei Funktionen besteht. Blendet man Tracing und Parameterüberprüfung einmal aus, reichen wenige Zeilen Code (siehe Listing 2). Sobald dieses Grundgerüst steht, kann mit der Einbindung in die Apex-Applikation begonnen werden.

Verwendung in der Apex-Oberfläche

Um eine mehrsprachige Maske, wie in den Abbildungen 1 und 2 gezeigt, zu erhalten, kann die Apex-Applikation ganz normal entwickelt werden. Die

```
create table T_LANG_STRING
( ID          VARCHAR2(40) not null,
  LANG        VARCHAR2(20) not null,
  LSTRING     VARCHAR2(4000) not null
);

alter table T_LANG_STRING
add constraint CO_UK_LANG_STRING unique (ID, LANG);
```

Listing 1

ID	LANG	LSTRING
USER	DE	Benutzer
USER	EN	User
LANGUAGE	DE	Sprache
LANGUAGE	EN	Language

Tabelle 1: Übersetzungstabelle für den einfachen Ansatz der Mehrsprachigkeit

Abbildung 1: Benutzermaske auf Deutsch

Abbildung 2: Benutzermaske auf Englisch

Mehrsprachigkeit kommt erst bei der Beschriftung der Labels, Buttons und Tabellen-Überschriften zum Tragen.

Die einzelnen Oberflächen-Objekte werden wie bewährt erstellt. Neu hinzu kommt, dass zu jedem Element der Oberfläche ein Element für den Beschriftungstext generiert wird. Die Elemente für die Beschriftungen werden in einer separaten Region erstellt und sind vom Typ „Ausgeblendet“. Als Quelltyp wird für die „Beschriftungselemente“ die Option „PL/SQL-Funktions-Body“ ausgewählt und mit dem folgenden Funktionsaufruf (exemplarisch für das Element „P3_USER_LANG“) gefüllt:

```
begin
return pk_ui.get_label(p_id =>
,USER', p_lang => :F1_LANG);
end;
```

Zwei Punkte der Implementierung sind jetzt noch offen:

1. Was steht in dem globalen Element „F1_LANG“ und wie und wo wird das Element gefüllt?

2. Angenommen „F1_LANG“ hat den Wert „DE“, dann steht in „P3_USER_LANG“ jetzt der Inhalt „Benutzer“. Doch wie wird dieser Inhalt als Beschriftung verwendet?

Füllen der Beschriftung

In dem Element „P3_USER“ muss als Inhalt für das Label Folgendes eingetragen werden: „&P3_USER_LANG.“. So wird sichergestellt, dass der Wert des Elements „P3_USER_LANG“ als Beschriftung verwendet wird (siehe Abbildung 3). Dieses Vorgehen kann natürlich auch auf alle anderen Seiten-Objekte wie Region, Buttons, Seitenüberschriften etc. und auch auf Applikations-Elemente wie Navigationen oder Registerkarten angewendet werden. Zu unterscheiden ist nur, dass bei einigen das Feld „Label“ und bei anderen das Feld „Titel“ zu füllen ist. Auch eine mehrsprachige Darstellung von Tabellenüberschriften, wie in den Abbildungen 4 und 5 gezeigt, lässt sich so auf einfache Weise realisieren.

```
create or replace package body
pk_lang_string is

function get_label
(
p_id in varchar2,
p_lang in varchar2
) return varchar2 is

v_output varchar2(4000);

begin
select lstring
into v_output
from t_lang_string
where lang = p_lang
and id = p_id;
return v_output;
end get_label;

function add_lstring
(
p_id in varchar2,
p_lang in varchar2,
p_lstring in varchar2
) return boolean is

v_exists number;

begin
begin
select 1
into v_exists
from t_lang_string ls
where ls.id = p_id
and ls.lang = p_lang;
exception
when no_data_found then
v_exists := 0;
end;

if v_exists > 0
then
return false;
else
insert into t_lang_string
(id,
lang,
lstring)
values
(p_id,
p_lang,
p_lstring);

return true;
end if;
end add_lstring;

end pk_lang_string;
```

Listing 2

Label	
Label	&P3_USER_LANG.
Horizontale/vertikale Ausrichtung	Rechts
Template	Optional Label
HTML-Tabellenzellenattribute	nowrap="nowrap"

Abbildung 3: Füllen des Labels mit dem Wert eines Elements

Benutzer	Benutzername	Sprache
ADMIN	Admin	de
TEST	test	de

Abbildung 4: Tabellenüberschriften auf Deutsch

Abbildung 6: Login

In den Berichtsattributen wird als Überschriftstyp die Option „PL/SQL“ gewählt und folgender Code hinterlegt (siehe Listing 3). Jetzt bleibt noch zu klären, wie das Anwendungselement, das die gewählte Sprache beinhaltet, gefüllt wird.

Ermitteln und Setzen der vom Benutzer gewünschten Sprache

Das praktikabelste Vorgehen ist hier, den Login-Dialog der Apex-Anwendung zu überarbeiten und ein Feld für die Sprachauswahl zu erstellen (siehe Abbildung 6). Hier kann der Benutzer zwischen den angebotenen Sprachen auswählen und diese werden bei Weiterleitung der Seite in das globale Element für die Spracheinstellung geschrieben (siehe Listing 4).

Im nächsten Schritt kann man sogar weitergehen und sicherstellen, dass der Benutzer auch die Login-Seite in der Sprache angezeigt bekommt, in der er sich beim letzten Mal ein-/ausgeloggt hat. Hierzu wird ein Prozess vor

User	Username	Language
ADMIN	Admin	de
TEST	test	de

Abbildung 5: Tabellenüberschriften auf Englisch

Formatmaske &F1_DATE_FORMAT_SHORT.

Abbildung 7: Datumsformat

dem Laden der Seite erzeugt, der sich die Daten aus einem Cookie heraus ermittelt (siehe Listing 5).

Auf diese Weise wird erreicht, dass eine Apex-Anwendung ohne Programmieraufwand schnell und flexibel auf eine andere Sprachversion umgestellt werden kann. Unter „Mehrsprachigkeit“ fällt aber nicht nur die Übersetzung von Texten.

Um dem Systembenutzer ein komfortables Arbeiten zu ermöglichen, sollten auch die Datums- und Zahlenfelder in dem für die Sprache korrekten Format dargestellt werden. Diese erweiterte Fragestellung der Mehrsprachigkeit wird in dem folgenden Absatz vorgestellt.

Länderkonforme Datumsformate

Um auch Datumsformate länderkonform darzustellen, gibt es die Möglichkeit, die Formate pro Sprache in einer Registry-Tabelle abzulegen und beim Darstellen von Datums- oder Zahlenformaten zu verwenden. Listing 6 zeigt

ein mögliches Beispiel, um ein Datumsformat auszulesen.

Ist so eine Funktion vorhanden, kann, analog zu dem Beispiel oben, ein globales Element der Anwendung damit gefüllt werden.

```

begin
return pk_ui.get_date_format
(p_lang => :F1_LANG);
end;
    
```

Für die Formatierung von Datumsfeldern kann nun auf den Wert des Elements zurückgegriffen werden (siehe Abbildung 7).

Weitere Möglichkeiten und Fazit

Die hier vorgestellte Implementierung der Mehrsprachigkeit in Apex soll als alternativer Ansatz zum Standard-Vorgehen betrachtet werden. Insbesondere lässt er sich weiter ausbauen. Auch das Handling bei den Funktionsaufrufen kann weiter vereinfacht werden, indem man zum Beispiel auf der Seite 0 eine Package-Konstante mit der Sprach-Einstellung füllt und so die Parameter-Übergabe komfortabler wird. Weitere Ausbaustufen sind denkbar und wünschenswert.

Warum empfehlen wir dieses Vorgehen zur Umsetzung von Mehrsprachigkeit?

Der reine Implementierungsaufwand für die benannten (und auch weiter ausgebauten) Formen der Mehrsprachigkeit ist zunächst erkennbar höher als bei „normalen“ Vorgehensweisen. Der Spareffekt findet sich aber im Nachhinein im Lebenszyklus der Anwendungen, etwa bei Wartung und weiterer Konfiguration. Es muss immer nur eine Version der Anwendung gepflegt und „deployed“ werden und die Erweiterung um zusätzliche Sprachen findet im Rahmen einer Datenveränderung in einer Tabelle statt und nicht im Rahmen einer Strukturänderung der Anwendung oder von Tabellen.

Positiv ist zudem, dass der Benutzer seine Sprache frei wählen kann und nicht auf die Einstellungen seines Web-Browsers festgelegt ist.

```

begin
return
pk_ui.get_label(p_id => ,USER', p_lang => :F1_LANG)
||':'|| pk_ui.get_label(p_id => ,USERNAME', p_lang => :F1_
LANG)
||':'|| pk_ui.get_label(p_id => ,LANGUAGE', p_lang => :F1_
LANG);
end;

```

Listing 3

```

declare
begin
:F1_LANG := upper(:P101_LANGUAGE);
:F1_USER := upper(:P101_USER);
end;

```

Listing 4

```

declare
v varchar2(255) := null;
c owa_cookie.cookie;
begin
c := owa_cookie.get(',LOGIN_USERNAME_COOKIE');
:P101_USER := c.vals(1);
c := owa_cookie.get(',LANG_LOCALE_COOKIE');
:P101_LANGUAGE := c.vals(1);
exception when others then
null;
end;

```

Listing 5

```

function get_date_format(p_lang varchar2) return varchar2 is
v_ret varchar2(100);
begin
v_ret := pk_mds.get_string_value(p_path => ,dateformat.supported.'
||p_lang);
return v_ret;
exception
when others then
v_ret := pk_mds.get_string_value(p_path => ,dateformat.de-
fault');
return v_ret;
end get_date_format;

```

Listing 6



Christian Piasecki
TEAM GmbH
cpi@team-pb.de

PROMATIS Appliances
Prozessoptimierung & Simulation

Oracle Applications
Oracle BI Suite
Usability
Enterprise 2.0

Enterprise Content Management
Accelerate-Mittelstandslösungen
Fusion Applications

Business Intelligence Applications
Managed Services
Oracle Infrastruktur

Oracle E-Business Suite
Oracle BPM Suite
Application Integration Architecture
Social BPM

Oracle CRM On Demand

Hier sind wir zuhause

Unser Alleinstellungsmerkmal: Intelligente Geschäftsprozesse und beste Oracle Applikations- und Technologiekompetenz aus einer Hand. Als Oracle Pionier und Platinum Partner bieten wir mehr als 15 Jahre erfolgreiche Projektarbeit im gehobenen Mittelstand und in global tätigen Großunternehmen.

Unsere Vorgehensweise orientiert sich an den Geschäftsprozessen unserer Kunden. Nicht Technologieinnovationen sind unser Ziel, sondern Prozess- und Serviceinnovationen, die unseren Kunden den Vorsprung im Markt sichern. Über Jahre gereifte Vorgehensmodelle, leistungsfähige Softwarewerkzeuge und ausgefeilte Best Practice-Lösungen garantieren Wirtschaftlichkeit und effektives Risikomanagement.



PROMATIS software GmbH
Tel.: +49 7243 2179-0 · Fax: +49 7243 2179-99
www.promatis.de · hq@promatis.de
Ettlingen/Baden · Hamburg · Berlin

Die Forderungen der Fachabteilungen nach Prozess-Abbildung in IT-Systemen haben in den letzten Jahren stark zugenommen. Dabei steht nicht nur die schnelle Umsetzung, sondern auch die Flexibilität für nachträgliche Anpassungen im Fokus. Apex ermöglicht es, genau diese Anforderungen umzusetzen. Allerdings sollte auch dort nie der Grundsatz der Einheitlichkeit aus Sicht des Anwenders verloren gehen.

Standardisierung mit Apex

Thomas Zielbauer, MAN Truck & Bus AG

Findet der Anwender in neuen Applikationen eine gewohnte Benutzerführung, reduziert sich der Schulungsbedarf beziehungsweise die Schulungsintensität erheblich. Es liegt also nahe, über eine Standardisierung der Anwendungen in Apex nachzudenken. In Apex gibt es mehrere Möglichkeiten diesen Ansatz anzugehen.

Wir könnten die Anforderungen des Fachbereichs in jeweils separaten Anwendungen mit demselben Look & Feel (Theme) umsetzen, was es uns ermöglicht, ein einheitliches Design mit jedoch getrennten Funktionalitäten an den Kunden zurückzugeben. Oder aber wir verfolgen den Ansatz der absoluten Standardisierung und integrieren verschiedene Projekte in eine Art „Framework-Anwendung“. Das hätte den Vorteil, dass Benutzer mit mehreren Berechtigungen sich nicht durch unterschiedliche Anwendungen bewegen müssten. Ebenso könnte standardisierte Funktionalität auf alle dort integrierten Teilsysteme vererbt werden. Natürlich würde sich die übergreifende Systemanmeldung auch mit einem Single-Sign-on-Ansatz decken, jedoch bringt der Framework-Ansatz Vorteile mit sich, die sich in der Praxis mit Oracle Apex als eindeutig interessanter herausstellen.

Dieser Ansatz der UI-Standardisierung brachte das Unternehmen des Autors vor einigen Jahren auf die Idee, mit Apex eine Framework-Applikation zu programmieren, also eine Anwendung, in der unterschiedlichste Projekte und Kundenanforderungen umgesetzt werden können. Eine Anwendung, die es auch den Entwicklern vereinfachen würde, schnell und flexibel auf Kunden-Anforderungen zu reagieren – aber

dabei auch niemals den definierten Standard verlassen zu müssen. Sozusagen Individualität im Standard.

Konzept-Erstellung

Diese und weitere Fragen prägten das Konzept grundlegend: Wie viele Variablen braucht der Fachbereich tatsächlich und wie soll der Zugriff darauf erfolgen? Wie kann eine Navigation umgesetzt werden, die es uns erlaubt zu bestimmen, was (beispielsweise Gemeinkosten oder Qualitätsdaten) über wen (entspricht dem Auswertungsobjekt wie etwa Kostenstelle oder Abteilung) angesehen werden kann? Mithilfe welcher Komponenten wird sich das alles umsetzen lassen?

Das endgültige Konzept gliederte sich in vier funktionale Bereiche (siehe Abbildung 1):

1. Variablen, die mithilfe einer jQuery-Funktion ein- und ausblendbar sein sollen
2. Eine Navigationsleiste, die ein jQuery-basiertes Dropdown-Menü enthalten soll (dynamisch, abhän-

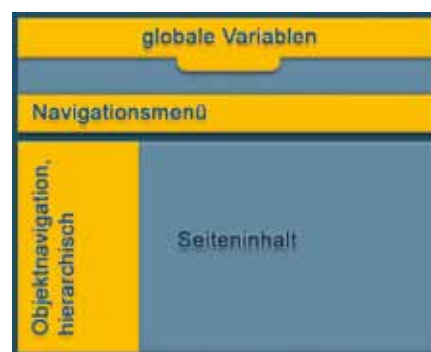


Abbildung 1: Das strukturierte, visuelle Konzept

gig von der Rollenzuweisung des angemeldeten Benutzers)

3. Die Auswertungsobjekt-Navigation, die die Organisationsstruktur der MAN Truck & Bus AG als Baumstruktur darstellt und erst bei Änderungsbedarf eingeblendet werden soll
4. Ausreichend Platz für den eigentlichen Inhalt der aktuellen Apex-Seite

Umsetzung des Konzepts in Apex

Der Schlüssel ist ein eigenes Theme und natürlich die Globalisierungsmöglichkeit in Apex mithilfe der Seite 0. Inhalte, die auf Seite 0 in Apex erstellt werden, projizieren sich selbst auf jede geöffnete Seite. Regionen, die beispielsweise im Seiten-Template definiert sind, lassen sich auf Seite 0 mit Leben füllen. Auf diese Weise entstehen das individuelle Benutzermenü, die Kostenstellen-Hierarchie als Baumstruktur und die „globalen“ Variablen auf jeder Seite in der Anwendung. Wie die Abbildungen 2 und 3 zeigen, ist das gesamte visuelle Konzept im minimierten Zustand sehr kompakt gehalten. Dadurch gewinnt man Platz für den eigentlichen Seiteninhalt.

Umgesetzt wurden die ein-/ausblendbaren Ebenen als „<div> HTML“-Elemente. Die Cascading-Stylesheet-Eigenschaft (CSS) „display“ wechselt durch das Klick-Ereignis bei diesen Elementen per JavaScript von „none“ auf „block“ und umgekehrt. Dafür lässt sich wunderbar die in jQuery integrierte Funktion „toggle“ nutzen, die mit oder ohne Animation verwendet werden kann. Die HTML-Elemente selbst wurden per CSS absolut positioniert, um nicht durch die weiteren HTML-



Abbildung 2: Eine neue, leere Seite inkl. der Seite 0, Regionen (minimiert)

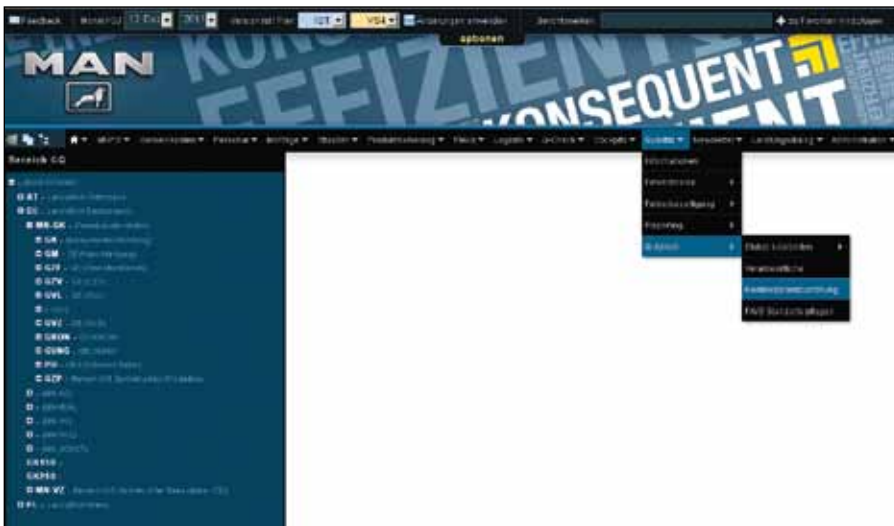


Abbildung 3: Siehe Abbildung 2, jedoch mit eingeblendeten Regionen von Seite 0



Abbildung 4: News & mehr auf der Einstiegsseite

Elemente der Seite unbeabsichtigt verschoben zu werden.

Damit die Sichtbarkeit der ausblendbaren Bausteine nicht durch nachfolgende Elemente eingeschränkt wird, kommt die CSS-Eigenschaft „z-index“ zum Einsatz. Damit lässt sich die Reihenfolge bei den sich überlap-

penden HTML-Elementen festlegen. Ein hierarchisch vorhergehendes Element kann damit im Vordergrund stehen und ein nachfolgendes Element überdecken, obwohl dessen eigentliche Darstellung im Hintergrund wäre.

Lediglich bei der Region für globale Variablen wurde auf ein leicht ver-

ändertes Konzept gesetzt. Am Kopf der Seite sollten aus Design-Gründen überlappende Elemente vermieden werden. Das Logo sollte stets sichtbar bleiben. Deswegen wurde der gesamte Bereich der globalen Variablen in den negativen Bereich der Seite nach oben verschoben. Hierfür lässt sich eine Positionsangabe mit CSS durch die Eigenschaft „top“ verwenden. „Top“ liegt bei einem Wert von „0“ und absoluter Positionierung an oberster Stelle im Browser. Bei einem negativen Wert verschwindet das Element sogar komplett aus dem Zeichenbereich des Browsers. Dieses Verhalten wurde genutzt, um mithilfe von JavaScript bei Klick auf „Optionen“ den nicht sichtbaren Bereich der globalen Variablen langsam nach unten zu fahren. Das hat einen angenehmen Animations-effekt zur Folge, bei dem die gesamte Seite nach unten verschoben wird. Das Logo bleibt dadurch jederzeit vollständig sichtbar und die Seite wirkt weniger starr.

Eine gemeinsame Schnittstelle

Die Framework-Applikation stellt inzwischen zahlreiche Projekte von unterschiedlichsten Fachabteilungen dar. Durch den hohen Grad an Personalisierung und verschiedenste Inhalte in der Anwendung kam die Idee nach einer gemeinsamen Schnittstelle auf. Etwas, das den kleinsten gemeinsamen Nenner widerspiegeln würde, also eine Art „Willkommenseite“. Die Gemeinsamkeit aller Anwender – abgesehen davon, dass sie dasselbe System verwenden – liegt darin, dass sie alle Mitarbeiter der MAN Truck & Bus AG sind. Diese Gemeinsamkeit wollten wir prägen (siehe Abbildung 4).

Die Willkommenseite enthält aktuell für jeden Anwender verständliche Informationen über das Unternehmen und das System selbst. Da die Fachbereiche unterschiedliche Sichten auf das Unternehmen haben, war es uns umso wichtiger, eine gemeinsame Sicht zu schaffen. Gerade bei großen Unternehmen wird die soziale Komponente und die Personalisierung in IT-Systemen zukünftig eine nicht zu unterschätzende Rolle einnehmen.

Da es das Ziel war, unternehmensweit einige wenige Kennzahlen aufzubereiten, die lediglich zur stärkeren Identifizierung mit dem Unternehmen beitragen sollten, fiel die Entscheidung auf animierte Business-Objects-Xcelsius-Diagramme. Diese werden über eine sich täglich ändernde XML-Datei automatisiert mit Daten befüllt. Per JavaScript und CSS wurde eine Art Diagrammwürfel in die Einstiegsseite integriert. Eine Seite des Würfels enthält weitere Navigationsmöglichkeiten in verschiedene Richtungen. Damit können unterschiedliche Diagramme animiert gewechselt werden. Mit dieser Technik konnten mehrere Diagramme auf dem Platz von einem in die Startseite integriert werden. Selbstverständlich enthält diese Einstiegsseite aber auch funktionale Bereiche wie beispielsweise eine Übersicht darüber, welche Daten sich derzeit im System befinden, oder auch eine Möglichkeit zur Passwortänderung.

Berechtigungskonzept der Anwendung

Zur Steuerung der Rechte empfiehlt es sich, den Apex-Standard zu benutzen. Dieser ist vielseitig anwendbar und sollte die meisten Anforderungen erfüllen. Die Überprüfung der einzel-

nen Rollen wurde sowohl auf der Seite selbst als auch in unserem Navigationsmenü platziert. Damit lässt sich nicht nur die Anwendung absichern, sondern auch das Menü individuell für den Anwender erzeugen. Diese Form der Personalisierung trägt erheblich zur Übersichtlichkeit der Anwendung bei. Die Überprüfung der Berechtigung kann ohne Bedenken auf einmal je Session reduziert werden. Gerade Anwendungen mit vielen Navigationszugriffen oder Seitenwechseln profitieren hiervon. Für den Anwender entsteht dadurch lediglich ein annehmbarer Nachteil: Wurde einem Anwender eine neue Rolle zugesprochen, wird diese erst nach erneuter Anmeldung aktiv (siehe Abbildung 5).

Rückwirkende Betrachtung

Die Framework-Anwendung ist seit knapp zwei Jahren im produktiven Betrieb mit vereinzelt Design-Anpassungen. Einige Projekte brachten neue Funktionen mit sich, die in den Framework-Standard integriert werden konnten. Die Anzahl der darin umgesetzten Projekte aus den Fachabteilungen nahm konsequent zu. Das Wachstum der Anwendung erfolgte also mehrdimensional. Aus Sicht des An-

wenders trat ein, was als Ziel definiert war: Sehr hohe Akzeptanz und nahezu kein Schulungsbedarf. Aus Entwicklersicht wurden die Erwartungen ebenso erfüllt: Neue Anforderungen können sehr zielgerichtet umgesetzt werden und im Standard integrierte Funktionen wie beispielsweise der Warnhinweis, dass eine Seite nicht ausreichend dokumentiert wurde, tragen erheblich zur Qualität der Anwendung bei (siehe Abbildung 6).

Integration eines Kundenprojekts in die Framework-Applikation

Ein interessantes Projekt aus dem Fachbereich war die Analyse von Qualitätsdaten. Als Datenquelle diente ein auf einer Oracle-Datenbank basierendes Data Warehouse. Das System sollte in der Lage sein, Produktionsfehlermeldungen zu überwachen und zu katalogisieren. Sobald aus der Datenmenge ersichtlich wurde, dass etwa ein Problem bei einer Konstruktionsgruppe vorlag, sollte der Prozess zum Abstellen des Fehlers ausgelöst werden. Die Protokollierung des Abstellprozesses sollte durch das System vollzogen werden. Während des Abstellprozesses trat ein zeitlich definiertes Regelwerk in Kraft, das den Anwender anhand festgelegter Standards durch den Prozess begleitete. Außerdem sollte es dem Kunden möglich sein, die einzelnen Prozessschritte mit Kollegen zu teilen. Das Hauptaugenmerk lag damit von Anfang an auf optimierter Analysefunktionalität, softwareseitiger Prozess-Unterstützung sowie einer klaren Kommunikation des Systems mit dem Anwender.

Die Katalogisierung des Datenbestandes erfolgte mithilfe von Oracle-Text. Um eine komfortable Suche mit bestmöglichen Suchergebnissen zu realisieren, wurde auf Schlüsselmerkmale ein Textindex gelegt. Der Textindex, der mit dem nachfolgenden Befehl erzeugt wurde, gibt bei dessen Verwendung eine Suchergebnisbewertung (score) zurück.

```
create index IDX_INDEXNAME on
TABELLE (INDEXSPALTE) indextype
is ctxsys.context
```

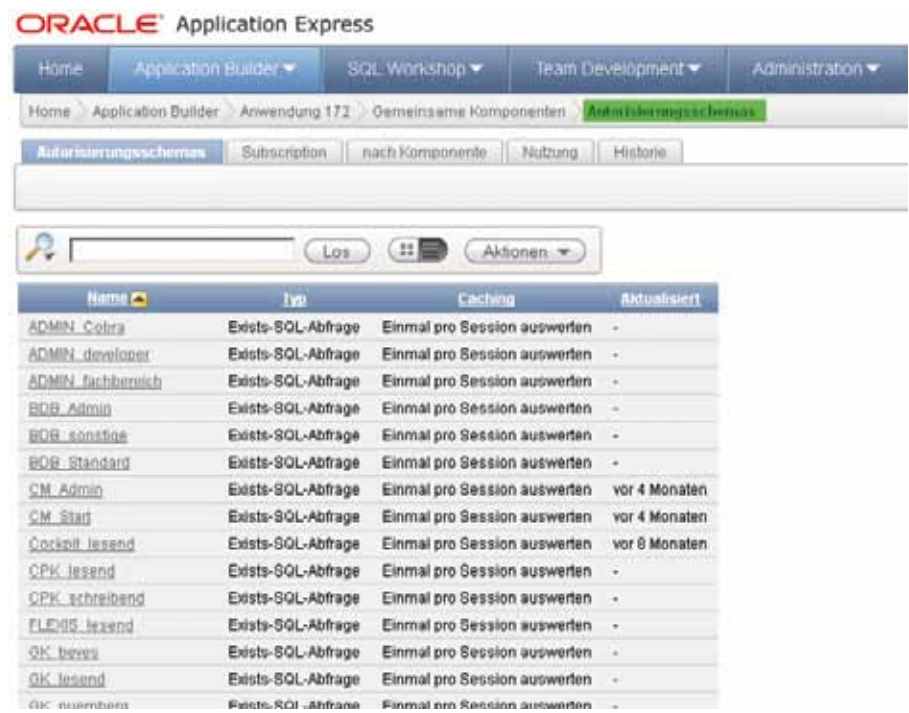


Abbildung 5: Autorisierungsschemata in Apex



Abbildung 6: Fehlerbehandlung serienmäßig



Abbildung 7: AnyChart5 bietet zahlreiche Einsatzmöglichkeiten

```
select
  SCORE(1) as Punkte,
  SPALTE_A,
  SPALTE_B,
  INDEXSPALTE
from TABELLE
where
  contains(INDEXSPALTE, '%' || :P1_Apex_TEXT_ELEMENT_SUCHE || '%', 1) > 0
```

Listing 1

Dieser „Score“ bewertet den Grad an Entsprechung des Ergebnisses zum Suchbegriff in Punkten. „Score“ lässt sich wie eine normale Spalte sortieren. Damit können die am ehesten dem Suchbegriff entsprechenden Ergebnisse zuerst aufgelistet werden. Eine Abfrage nach einem Suchbegriff erfolgt über den SQL-Befehl „contains“ (siehe Listing 1).

Nachdem die gezielte Navigation innerhalb der Daten mit der Umsetzung der Suchfunktion einen wichtigen Schritt nach vorn gemacht hatte, galt es, diesen sinnvoll in die Anwendung zu integrieren. An dieser Stelle nahm der Framework-Gedanke wieder konkrete Formen an: Die benötigten Regionen der Suchfunktion wurden auf Seite 0 gelegt. Allerdings wurde die Publizierung auf bestimmte Seiten beschränkt, da diese nur auf einigen benötigt war.

Die Suchfunktion wurde aufgrund der hohen Qualität der Ergebnisse in einige Berichte als Filtermöglichkeit aufgenommen. Ziel war es unter anderem, ein grafisches Monitoring eines eindeutigen Fehlerbildes zu ermöglichen.

Da die Suchfunktion so programmiert wurde, dass sie, sobald ein Wert selektiert wurde, auch ein konkretes Fehlerbild zurückgeben konnte, wurde dieses in Variablen auf Seite 0 abgelegt. Dadurch war das konkrete, ausgewählte Fehlerbild wiederum für die Filterung der Daten aus dem Data Warehouse griffbereit. Die Abfrage erfolgte also gezielt und damit äußerst performant. Nun war es möglich, auf

Verwalten Sie Ihre Requirements mit IBM Rational DOORS?
Haben Sie ein PLM System von Oracle im Einsatz?
Lassen Sie sich den Mehrwert durch eine Integration nicht entgehen!
Besuchen Sie uns auf der DOAG 2012 Applications.

X
XOPENDOOR

ORACLE Gold Partner

usb GmbH • Betastr. 13a • 85774 Unterföhring • Telefon 089 / 99894283
http://www.usb-muc.de • info@usb-muc.de



Abbildung 8: Verwenden eines „Region Display Selectors“ für eine Region



Abbildung 9: Die Auswirkung der JavaScript-Anpassung

diese gefilterte Datenmenge ein Reporting nach Wunsch des Fachbereichs aufsetzen.

Apex bietet im Standard bereits eine breite Masse an Möglichkeiten zur Umsetzung eines grafischen Reporting. In Version 4.1 enthält Apex mit AnyChart 5 eine äußerst mächtige Komponente, um Flash-Diagramme dynamisch zu erzeugen. Für die Erstellung dieser ist es möglich, den integrierten Assistenten zu verwenden oder – sofern Kenntnisse über die AnyChart-5-XML-Syntax vorhanden sind – auch den Code beliebig selbst zu schreiben. Aufgrund der besseren Anpassungsmöglichkeiten fiel die Entscheidung, für das grafische Reporting die XML-Syntax selbst zu codieren und Apex lediglich das Füllen des Templates mit Daten zu überlassen (siehe Abbildung 7).

Da das Reporting so konstruiert wurde, dass der Fachbereich verschiedene Sichten auf eine gefilterte Menge an Basisdaten anwenden konnte, lag es nahe, die Datenmenge in eine Collection zu schreiben. Diese konnte nun von den unterschiedlichen Berichten einer Seite als Datenbasis angesprochen werden. Der Performance-Gewinn war dadurch enorm. Um die Seite aber nicht mit zu vielen Berichten zu überladen, wurde ein Standard-Fea-

```
/* ausblenden aller display-selector regionen */
apex.jquery(, #SHOW_ALL').hide();
apex.jquery(, #DISP_REG_1').hide();
apex.jquery(, #DISP_REG_2').hide();
/* einblenden von Region 1 */
apex.jquery(, #DISP_REG_1').show();
/* Initialisierung Region 1 */
apex.jquery(, li.apex-rds-first').removeClass(„apex-rds-selected“);
apex.jquery(, li.apex-rds-first').hide();
apex.jquery(, a[href=„#DISP_REG_1“]).closest(, li').
addClass(„apex-rds-selected“);
```

Listing 2

ture in Apex verwendet: der „Region Display Selector“. Dieser ermöglicht es, bestimmte Regionen einer Seite ein- und ausblendbar zu machen. Haben wir beispielsweise zwei Regionen auf einer Seite und aktivieren für beide den „Region Display Selector“, kann man entweder Region 1, Region 2 oder alle Regionen zugleich anzeigen lassen. Zu beachten ist nur, dass auf der Apex-Seite eine neue Region vom Typ „Region Display Selector“ erstellt werden muss. Diese spezielle Region übernimmt die Steuerung der Sichtbarkeit der Regionen im User Interface. Da auch diese Region vollständig anpassbar ist, lässt sie sich per CSS wunderbar in eine Liste mit Registerkarten-Optik umwandeln.

Wie Abbildung 8 zeigt, wird die Region einfach per Einstellung ein-/ausblendbar gemacht. In der Abbildung ist aber auch das Feld „Static ID“ hervorgehoben. Diese wurde für unsere Regionen fest hinterlegt. Der Grund hierfür ist schnell erklärt: Im Apex-Standard öffnet sich eine Seite mit mehreren ein-/ausblendbaren Regionen immer im Modus „Show all“. Damit würden alle Diagramme beim Öffnen der Seite auf einmal sichtbar sein. Die Seite würde dadurch allerdings überladen wirken. Ziel war, dass zum einen der „Display Region Selector“-Eintrag „Show all“ komplett verschwindet und zum anderen, dass nur eine Region auf einmal angezeigt wird. Da die Region IDs fest definiert wurden, war es möglich, über folgenden JavaScript-Code (siehe Listing 2), der beim Laden der Seite ausgeführt wird, die Sichtbarkeit zu steuern (siehe Abbildung 9).

Mit dieser Methode fiel es nicht schwer, die Anforderungen des Fachbereichs gebündelt, aber dennoch übersichtlich auf einer Themenseite darzustellen. Der Anwender hatte hierbei lediglich eine einmalige, kurze Wartezeit für das Bereitstellen der Daten. Sobald diese geladen wurden, konnte ohne Ladezeiten zwischen den Sichten gewechselt und damit die Qualitätsdaten analysiert werden.

Fazit

Das kurz diskutierte Projekt ist ein gutes Beispiel für die erfolgreiche Integration einer Data-Warehouse-Analyse über eine standardisierte Apex-Anwendung. Es zeigt sehr deutlich, mit welchen Mitteln bereits innerhalb des Apex-Standards gearbeitet werden kann. Außerdem bietet es einen kleinen Einblick in die Möglichkeiten, die durch Anpassungen der Apex-Features entstehen können. Apex bietet nahezu keine Grenzen – auch nicht bei der Definition eigener Standards.

Thomas Zielbauer
MAN Truck & Bus AG
thomas.zielbauer@man.eu



In diesem Beispiel wird die Entwicklung eines Apex-Region-Type-Plug-ins vorgestellt. Um eine inhaltlich möglichst umfangreiche Lösung abzubilden, kommt die jQuery-Image-Slider-Technik zum Einsatz.

Erstellung eines Apex Plug-ins

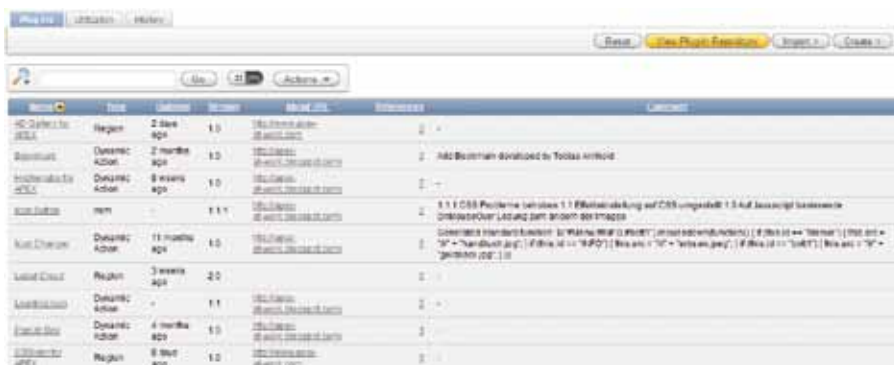
Tobias Arnhold, IT-Consulting

Warum benötigt man eigentlich Apex Plug-ins? Sie machen die Wiederverwendung von Web- und SQL-Techniken modularer einsetzbar. Das heißt, es muss kein PL/SQL-, kein JavaScript-Code und auch keine Datei kopiert beziehungsweise angepasst werden, wenn eine bestimmte Funktionalität wiederverwendet werden soll.

Wann lohnt sich der Aufbau eines Plug-ins? Er lohnt sich immer dann, wenn man Codes innerhalb einer Apex-Umgebung häufig wiederverwenden will. Auch wenn nachträglich Erweiterungen an dem Plug-in vorzunehmen sind, ist die Aktualisierung im Rahmen der Apex-Umgebung recht einfach realisierbar. Es gibt die folgenden Plug-in-Typen: Region-Type-Plug-ins, Item-Type-Plug-ins, Dynamic-Action-Plug-ins, Process-Type-Plug-ins, Authorization-Type-Plug-ins und Authentication-Type-Plug-ins.

Ein Apex Plug-in erstellen

Unter „Shared Components“ findet man den Punkt „User Interface -> Plug-ins“. Nach einem Klick auf „Plug-ins“ erscheint die Übersicht aller erstellten und installierten Plug-ins.



Name	Type	Version	Status	Plug-in URL	Released
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0
10-Default-UI	Region	2.0.0	1.0	10-Default-UI	1.0

Abbildung 1: Übersicht der Plug-ins

Mit einem Klick auf „Create ->“ gelangt man zur Maske für die Erstellung eines Apex Plug-ins. Folgende Menüreiter stehen zur Verfügung:

- **Name**
Beschreibung des Plug-in-Namens und -Typs
- **Subscription**
Definition eines Master-Plug-ins, von dem aus zentral aktualisiert werden soll. Weitere Informationen dazu unter <http://tinyurl.com/6tfqohv>
- **Settings**
File-Prefix definiert, ob die verwendeten Dateien aus der Oracle-Datenbank oder dem Webserver geladen werden.
- **Source**
Definition der PL/SQL-Plug-in-Logik. In diesem Bereich sind die PL/SQL-Funktionen angelegt, die für das Ausführen der Plug-in-Prozesse notwendig sind.
- **Callbacks**
Funktionsnamen, die zum Aufruf des Plug-ins notwendig sind. Der „Render Function Name“ ist zwingend notwendig, da über diesen Funktionsnamen auf die Startfunktion im „Source“-Bereich zugegriffen wird.

- **Standard Attributes**
Zusätzliche festdefinierte Attribute-Optionen zur Verwendung des Plug-ins. Die Attribute variieren zwischen Plug-in-Typen.
- **Custom Attributes**
Benutzerdefinierte Attribute, die beim Anwenden des Plug-ins innerhalb der Apex-Anwendung gesetzt werden. Diese Attribute finden innerhalb des PL/SQL-Codes Verwendung.
- **Files**
Upload-Bereich von JavaScript- und CSS-Files, die innerhalb des Plug-ins verwendet werden. Dateien sind in der Datenbank abgelegt.
- **Events**
Erstellung von benutzerdefinierten Events
- **Information**
Infos zur Version und der URL des Herstellers
- **Help Text**
Hilfe-Text zum Plug-in
- **Comments**
Entwicklerkommentare

Der PL/SQL-Code des Plug-ins

Neben den vielen Einstellungsmöglichkeiten kommt es vor allem auf den PL/SQL-Code des Plug-ins an. Dabei stellt sich die Frage: Was unterscheidet diesen Code von normalen PL/SQL-Funktionen?

Der Aufbau der Hauptfunktion ist mit festen Übergabe-Parametern vorgegeben und beinhaltet auch einen fest vordefinierten Rückgabewert. Dabei unterscheidet sich die Benennung der Parameter und des Typs je nach Plug-in-Art. Der Name der Funktion kann frei gewählt werden und ist später im Feld „Render Function Name“ zu hinterlegen.

Um ein besseres Verständnis vom Aufbau zu bekommen, wurde für dieses Beispiel eine Art leere Hauptfunktion erstellt. Diese generiert lediglich eine DIV-Box mit einem Beispieltext und dem Wert des einzigen Übergabeparameters (siehe Listing 1).

Innerhalb des Codes sind verschiedene Kommentare hinterlegt, die auf einzelne Teilbereiche aufmerksam machen sollen:

1. Funktionsaufruf mit Definition der Parameter

Aufbau des oben erwähnten Funktionsaufrufs

2. Plug-in-Attribute per Variable übergeben

Innerhalb des „Declare“-Bereichs sind die Parameter und die „Custom Attributes“ zu definieren. Dabei wird der Typ des im Plug-in definierten Attributs verwendet: „apex_application_page_regions.attribute_01%type“. Außerdem wird der Parameter an eine sprechende

Variable übergeben: Sprechende Variable „l_text“ und Übergabewert „p_region.attribute_01“

3. Laden einer JavaScript-Datei

Funktionsaufruf, um eine JavaScript-Datei zu laden. Beim Aufruf einer CSS-Datei unterscheidet sich die zu verwendende Funktion nur im Namen „apex_css.add_file (...)“;

4. Generierung des Inhalts für die Darstellung auf dem Bildschirm

Die Erstellung des Outputs kann beliebig komplex werden, da dies vom Inhalt beziehungsweise vom Ergebnis des Plug-ins abhängig ist. Hier ist besonders die Nutzung der Variablen „p_region.static_id“ hervorzuheben. Sie ermöglicht es, den vom Apex-Item gelieferten Regionsnamen im Quellcode zu verwenden. Innerhalb des Items kann der Regionsname vom Entwickler definiert werden oder es wird automatisch ein eindeutiger Name generiert. Dies hat den Vorteil, dass das Plug-in mehrfach auf der Seite aufgerufen

werden kann, das erstellte Element jedoch immer einen eindeutigen Namen besitzt.

5. Debug-Informationen generieren

Der gezeigte Code kann an den jeweils relevanten Stellen verwendet werden, um entscheidende Debug-Informationen hinzuzufügen. Diese Informationen können nur vom Entwickler ausgelesen werden.

Sobald man sich an die Art und Weise der Plug-in-Funktionslogik gewöhnt hat, ist es möglich, komplexere Plug-ins mit wenig Aufwand zu erstellen.

Erstellung des jQuery-Image-Slider-Plug-ins

Im vorangegangenen Beispiel wurde eine simple DIV-Box generiert. Das folgende Beispiel geht nun auf die Erstellung eines jQuery-Image-Slider-Plug-ins ein. Der jQuery-Image-S3Slider blendet verschiedene Bilder nacheinander mit einem grafischen

```
-- 1. Funktionsaufruf mit Definition der Parameter
function render_s3slider (
    p_region          in apex_Plug-in.t_region,
    p_Plug-in         in apex_Plug-in.t_Plug-in,
    p_is_printer_friendly in boolean
) return apex_Plug-in.t_region_render_result is

    -- 2. Plug-in Attribute per Variable übergeben
    l_text          apex_application_page_regions.attribute_01%type := p_region.attribute_01;
begin
    -- 3. Laden einer Javascript Datei
    apex_javascript.add_library (
        p_name      => ',s3slider',
        p_directory => p_Plug-in.file_prefix,
        p_version   => null
    );

    -- 4. Generierung des Inhaltes für die Darstellung auf dem Bildschirm
    sys.htp.p(<div id="||apex_Plug-in_util.escape(p_region.static_id, true)||'_slider">');
    sys.htp.p('Mein erstes Plug-in mit Einstellungswert für l_text: ' || to_char(l_text));
    sys.htp.p(</div>');

    -- 5. Debuginformationen generieren
    if wwv_flow.g_debug then
        wwv_flow.debug(',DIV container finished.');
```

Listing 1

```

select
  id_col as id,
  ,#OWNER#.my_image_display?p_image_id=' || to_char(id) as image,
  text,
  description,
  ,left' as position
from my_image_table

```

Listing 2

```

function com_aaw_s3slider(pRegionId, pPlug-inName, pItemList) {
  var aItemList;
  apex.event.trigger(
    $x(pRegionId+"_slider"),
    „apexbeforerefresh“
  );

  var lAjaxRequest = new htmldb_Get(
    null,
    $v(,pFlowId'),
    „PLUG-IN=“ + pPlug-inName,
    $v(,pFlowStepId')
  );
  if (pItemList != „“) {
    aItemList = pItemList.split(„,“);
    for (var i=0;i<aItemList.length;i++) {
      lAjaxRequest.add(aItemList[i], $v(aItemList[i]));
    }
  }
  lAjaxRequest.addParam(,p_debug', $v(,pdebug'));
  var lAjaxResult = null;
  lAjaxResult = lAjaxRequest.get();
  $x(pRegionId+"_slider").innerHTML = lAjaxResult;
  eval(pRegionId+'_run_slider()');
  apex.event.trigger(
    $x(pRegionId+"_slider"),
    „apexafterrefresh“
  );
}

```

Listing 3

Übergang ein. Außerdem ist es möglich, individuelle Texte dem Bild hinzufügen

Um solch eine Lösung umzusetzen, muss der Plug-in-Code entsprechend erweitert werden. Dazu die Vorstellung der einzelnen Plug-in-Bereiche mit den hier verwendeten Werten:

- **Name**
Es werden die Namen definiert und der Plug-in-Typ ausgewählt. Bei dem Feld „Internal Name“ sollte ein möglichst eindeutiger Name verwendet werden. Als Name „S3Slider for Apex“, als Internal Name „COM.AAW.S3SLIDER“ und als Type „Region“.
- **Subscription und Settings**
Diese Werte werden nicht verändert, da kein Master-Plug-in existiert

und die Dateien aus der Datenbank geladen werden sollen.

- **PL/SQL-Code**
Es folgt der etwas komplexere PL/SQL-Code. Als Code-Basis wurde die sehr gut aufbereitete Logik von Carsten Czarski, Oracle Deutschland, verwendet, die er in seinem Label-Cloud-Plug-in nutzt. Auf Basis dieser Grundlage war es möglich, innerhalb von ein paar Stunden eine funktionsfähige Implementierung in Apex zu schaffen. Der Code nutzt zwei Funktionen:

– *render_s3slider*

Dieser Teil wird beim Start der Seite einmalig aufgerufen, um die JavaScript-, CSS- und HTML-Komponenten des Slider zu la-

den. Zum Schluss wird die zweite Funktion aufgerufen, die entsprechend Inhalte ausliest. Hier ist die Bedeutung der Plug-in-Attribute hervorzuheben, da diese an allen entscheidenden Stellen verwendet werden, um dynamisch und eindeutig die Komponenten des Image-Slider abzubilden.

– *render_s3slider_ajax*

Dieser Part selektiert die Daten des übergebenen Select und gibt die Inhalte formatiert aus. Als wichtigste Komponente kommt hier der SQL-Handler ins Spiel. Dieser wandelt das vom Apex-Entwickler erstellte Select in eine Cursor-ähnliche Funktionalität um, zum zeilenbasierenden Auslesen und Verarbeiten der Werte. Dadurch sind ein besseres Fehlerhandling und eine einfache Verwendung der Spalten möglich.

Das entsprechende Listing steht unter <http://www.doag.org/go/doagnews/asc> zum Download bereit.

Callbacks

Hier sind die PL/SQL-Funktionen referenziert, damit diese während der Laufzeit aufgerufen werden können. Der Render-Function-Name ist „render_s3slider“ und der AJAX-Function-Name lautet „render_s3slider_ajax“.

Standard Attributes

Bei der Definition der Standard-Attribute ist es wichtig, Häkchen bei „Region Source is SQL Statement“ und bei „Region Source Required“ zu setzen. Dadurch trägt der Entwickler ein entsprechendes Select ein. Das hier angegebene Beispiel nutzt genau fünf Spalten, weshalb auch unter „Minimum und Maximum Columns“ jeweils „5“ eingetragen werden muss. Listing 2 zeigt einen Beispiel-Select.

Custom Attributes

In diesem Bereich werden die einzelnen Attribute je nach Typ definiert. Im Beispiel kommen Text-, Number- und Select-Listen-Felder vor. Bei den Select-

Listen muss zusätzlich die mögliche Auswahl vordefiniert sein.

Files

Es werden das Originalprogramm und eine eigens erstellte Erweiterung hochgeladen (aaw_s3slider.js, s3slider.js). Beides wird dann im PL/SQL-Code aufgerufen und im Browser ausgeführt. Die Erweiterung macht das dynamische Refreshen der Region während der Laufzeit möglich, indem beim Aufruf der Funktion ein Datenbank-Request erstellt und der neue Wert aus der Datenbank selektiert wird. Listing 3, Seite 27, zeigt den JavaScript-Code.

Soweit zu den wichtigsten Informationen, die für dieses Plug-in relevant sind. Die letzten Bereiche (Events, Information, Help Text, Comments) werden entweder nicht genutzt oder

nur informativ gefüllt (beispielsweise Information -> Version: 1.0). Beim Ausführen des Plug-ins mit einem entsprechenden Select erscheint nun eine ansehnlich inszenierte Apex-Implementation des jQuery-S3Slider. Das Ergebnis ist online unter <http://apex.oracle.com/pls/apex/f?p=65560:12> zu sehen. Neben diesem Beispiel existiert auch eine einfache Anleitung zum Erstellen eines Dynamic-Action-Plug-ins unter <http://tinyurl.com/7lrstme>.

Wichtige Links zum Thema

- SQL-Selektionsbeispiel für das Plug-in: <http://tinyurl.com/7dxbp7l>
- Original S3Slider-Beispiel: <http://tinyurl.com/6be3af>
- Generelle Apex-Dokumentation: <http://tinyurl.com/898drb8>

- Apex_PLUG-IN_UTIL-Dokumentation: <http://tinyurl.com/7efamfy> Quellcode-Vorlage für das Label-Cloud-Plug-in von Carsten Czarski: <http://tinyurl.com/7c696yb>
- Buch über Apex Plug-ins: <http://tinyurl.com/6lmtsfr>

Tobias Arnhold
tobias-arnhold@hotmail.de
<http://www.apex-at-work.com>



„Bereit für die Cloud und gerüstet für Mobile Apps ...“

Bei der letzten Oracle OpenWorld eröffnete Joel Kallman seine Keynote mit den Worten „I was involved in Apex Development since the first line of code – and since that moment we were ready for the cloud.“ Das war 1999. Als Nachfolger einiger anderer Ansätze wurde Apex zunächst eher kritisch beobachtet und nicht wirklich angenommen. Heute stellt Joel Kallman völlig zurecht fest, dass Apex zum Industrie-Standard geworden ist, der auch die Entwicklung und den Betrieb von unternehmenskritischen Applikationen abdeckt und ein modernes Software Engineering ermöglicht. Wie er als Produktverantwortlicher die Entwicklung von Apex, aber auch Test, Dokumentation, Q&A und Produktsupport gewährleistet, beschreibt Kallman sehr schön in seinem Interview auf DOAG.tv unter <http://www.doag.org/home/doagtv.html>. Zwei Kriterien machen Apex heute so erfolgreich:

- Beim visionären Schwenk auf Cloud Computing sind künftig Entwicklungsumgebungen gefragt, deren Infrastruktur völlig dezentral und Standort-unabhängig ist. Wer weiß, dass Apex mit Apex entwickelt wird und das Team dabei über den ganzen Erdball verteilt sitzt, versteht auch, dass diese Eigenschaften erfüllt sind. Als Bestandteil der Datenbank und mit lediglich einem Browser auf der Client-Seite ist Apex ideal gestaltet.
- Beim Software-Design ist heute schon die Forderung nach „Any Devices“ maßgeblich und dominiert klar vor weiteren funktionalen Erweiterungen. Dass Apex heute in der Lage ist, Applikationen für mobile Endgeräte zu erstellen, hat auch die DOAG schon mehrfach vorgestellt, diskutiert und dokumentiert. Mit der 2012 erscheinenden Apex Version 4.2 und der darin integrierten Funktionalität von jQuery Mobile sieht sich Oracle mit Apex bestens gerüstet für das Mobile Computing.

Patrick Wolf ist einer der Mitglieder des Apex-Entwicklungsteams. Er weist bei der Frage nach einem Migrationsweg aus Oracle Forms wohl zu Recht darauf hin, dass neben der optimalen Architektur auch das Wissen und die Logik aus Forms-Anwendungen sehr gut übertragen werden können.

SQL und PL/SQL sind ebenfalls Bestandteile von Apex. Der Umstieg ist somit für Entwickler einfacher und schneller. Anstelle einer automatisierten Migration sollte man den Wechsel von Forms nach Apex jedoch als Re-Engineering betrachten. Transaktionen lassen sich nicht unbedingt „1:1“ abbilden. Die schrittweise Migration und Übertragung der Logik ist zwar vereinfacht, muss und sollte aber auch im neuen Zielsystem überdacht werden.

Stefan Kinnen, Leiter der Development Community
stefan.kinnen@doag.org

Seit dem ersten Release von HTML DB zusammen mit Oracle10g im Jahr 2003 ist eine Menge passiert. HTML DB heißt schon seit einiger Zeit Apex; die Version 4.1 ist verfügbar und es gibt eine Fülle neuer Features und Möglichkeiten. Wie bei jedem Software-Produkt gibt es inzwischen auch in Apex Features, die zwar sehr nützlich sind, jedoch in Artikeln, Blogs oder Howto-Dokumenten kaum erwähnt werden. Dieser Artikel stellt einige selten genutzte, aber sehr nützliche Apex-Features vor.

Versteckte Juwelen: kaum bekannte Apex-Features

Carsten Czarski, ORACLE Deutschland B.V. & Co. KG

Format-Masken für Berichte

Es ist allgemein bekannt, dass man eine numerische oder Datumsspalte in einem Apex-Bericht formatieren kann. Dabei werden die aus SQL bekannten Format-Masken wie „9G999D99“ oder „DD-MM-YYYY“ verwendet. Apex bietet jedoch zusätzliche Format-Masken an, welche die Lesbarkeit eines Berichts massiv verbessern können:

- „PCT_GRAPH“ stellt einen Prozentwert als Balkengrafik dar

- „SINCE“ übersetzt ein Datum in einen Text
- „FILESIZE“ übersetzt eine Zahl in eine lesbare Dateigröße (KB, MB, GB, TB)

Der Bekanntheitsgrad von „PCT_GRAPH“ hat sich seit der Version 4.0 verbessert, da es als Schnellauswahl direkt unterhalb des Eingabefelds für die Format-Maske bereitsteht (siehe Abbildung 1). Es ist geeignet für Zahlen zwischen 0 und 100. Durch Doppelpunkte getrennt werden zunächst die Vordergrundfarbe, danach die Hinter-

grundfarbe und schließlich die Breite des Balkens in Pixel angegeben. Der Balken wird von rechts her gefüllt – möchte man es von links her haben, kann man in der Berichtsabfrage den Wert der Tabellenspalte von 100 abziehen. Der Prozentbalken wird mit HTML-Code erzeugt; da dieser ab Apex 4.0 standardmäßig maskiert wird, ist es wichtig, die Darstellung der Spalte auf „Standardberichtsspalte“ umzustellen.

Die Format-Masken „SINCE“ und „FILESIZE“ sind noch einfacher zu nutzen: Das Schlüsselwort wird einfach bei „Number / Date Format“ eingetragen und es muss nichts mehr angehängt werden. „SINCE“ ist sogar multilingual und unterstützt alle Sprachen, für die eine Übersetzung der Apex-Entwicklungsumgebung installiert ist. Ist die Apex-Anwendung beispielsweise auf „deutsch“ eingestellt und die deutsche Übersetzung für den Application Builder installiert, so werden die Texte auch auf Deutsch dargestellt (siehe Abbildung 2).

Die Vorteile der eingebauten Format-Masken liegen klar auf der Hand: Erstens muss man sie nicht selbst entwickeln und zweitens wirken sie sich nicht negativ auf Sortier- und Suchfunktionen aus. Alle drei Format-Masken stehen auch auf PL/SQL-Ebene als Funktionen im Package „Apex_UTIL“ bereit:

- Apex_UTIL.GET_SINCE
- Apex_UTIL.FILESIZE_MASK
- Apex_UTIL.HTML_PCT_GRAPH_MASK

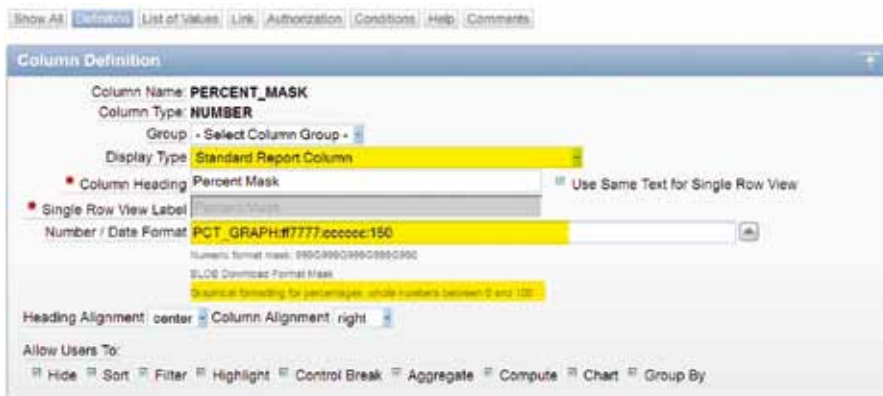


Abbildung 1: Format-Maske „PCT_GRAPH“

Formatmasken					
Percent	Percent Mask	Datum	Datum Mask	Filesize	Filesize Mask
15	<div style="width: 15%; background-color: #ff9900; border: 1px solid #ccc;"></div>	27.01.81	vor 30,8 Jahren	1782376761762	2TB
85	<div style="width: 85%; background-color: #ff9900; border: 1px solid #ccc;"></div>	06.11.11	vor 2 Wochen	176237	172KB

Abbildung 2: Format-Masken in Aktion

Schnell-Auswahlen nutzen

Schnell-Auswahlen sind Links unterhalb von Eingabefeldern in Formularen – beim Klick darauf wird das Eingabefeld automatisch gefüllt. Im Application Builder werden die Schnell-Auswahlen sehr intensiv genutzt, zum Beispiel bei den Format-Masken (siehe auch Abbildung 1). Apex erlaubt Schnell-Auswahlen auch in eigenen Anwendungen, und zwar für alle Formular-Elemente. Navigiert man zu den Einstellungen eines Elements, so gibt es den Abschnitt „Schnell-Auswahl“ (Quick Picks); die Abbildungen 3 und 4 zeigen das anhand eines Beispiels. Das Formular ist sofort wesentlich leichter zu bedienen.

Management der Apex-Komponenten

Viele der weniger bekannten Features beschäftigen sich mit dem Management der Apex-Komponenten. Auch Features, die nicht direkt zur Benutzeroberfläche einer Anwendung beitragen, gehören nahezu immer zu den „kaum bekannten“ Funktionen.

Erstell-Optionen

Erstell-Optionen (Build Options) erlauben es, mehrere Komponenten einer Apex-Anwendung auf einmal zu aktivieren oder zu deaktivieren, ohne dass dazu eigene Bedingungen erforderlich sind. Dazu ist zuerst eine neue Erstel-Option zu erzeugen, was in den „gemeinsamen Komponenten“ und dort unter „Erstell-Optionen“ geschieht. Das Beispiel in Abbildung 5 erzeugt eine neue Erstel-Option mit dem Status „Exclude“. Mit dieser Erstel-Option können nun alle Anwendungskomponenten versehen werden (siehe Abbildung 6). Solange der Status auf „Exclude“ steht, werden sie zur Laufzeit nicht ausgeführt. Erstel-Optionen werden auch beim Export berücksichtigt; es ist möglich, eine Anwendung mit oder ohne Komponenten einer bestimmten Erstel-Option zu exportieren.

Erstell-Optionen sind sehr zu empfehlen, wenn innerhalb der Anwen-

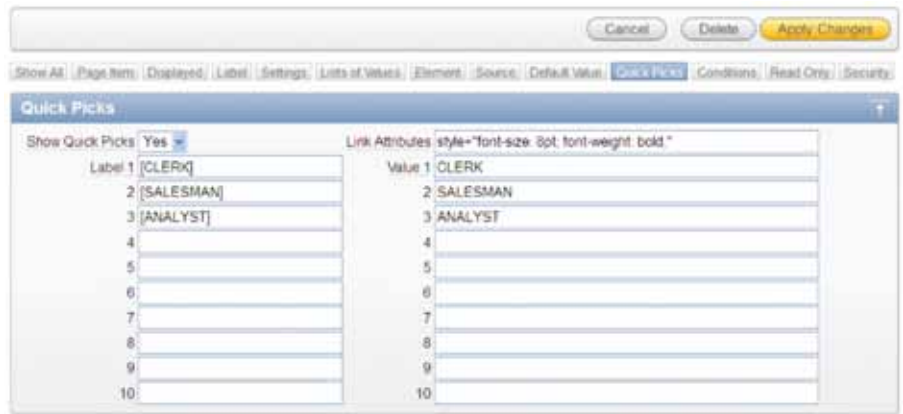


Abbildung 3: Einrichten einer Schnell-Auswahl



Abbildung 4: Schnell-Auswahlen in Aktion



Abbildung 5: Eine neue Erstel-Option mit dem Status „EXCLUDE“

nung eine neu eingebaute Funktionalität aus vielen Komponenten auf mehreren Seiten besteht – das Aktivieren oder Deaktivieren wird mit einfachem Setzen der Erstel-Option auf „Include“ oder „Exclude“ realisiert. Anwendungskomponenten dokumentieren, zu welchem Zweck und gebe-

nenfalls zu welcher Zeit eine Komponente eingeführt wurde.

Kommandozeilen-Export und Apex_APPLICATION_INSTALL

Wenn es darum geht, Anwendungskomponenten auf einen anderen Ser-

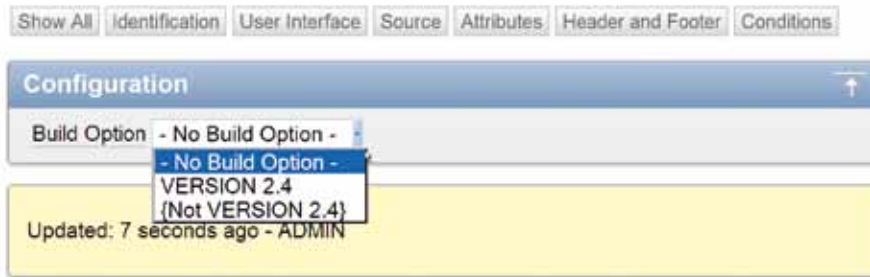


Abbildung 6: Erstell-Option bei einer Apex-Komponente einstellen

```
$ apxexp -db localhost:1521:orcl \
        -user scott -password **** \
        -instance

Exporting Application 100:'Bildbearbeitung'
  Completed at Tue Nov 22 11:25:05 CET 2011
Exporting Application 102:'Sample Application'
  Completed at Tue Nov 22 11:25:07 CET 2011
Exporting Application 109:'ODD Editions'
  Completed at Tue Nov 22 11:25:08 CET 2011
:
$
```

Listing 1: Export einer Apex-Anwendung von der Kommandozeile

ver zu übertragen oder in ein Versions-Kontrollsystem wie CVS oder Subversion einzustellen, so ist der Anwendungsexport das Mittel der Wahl. Aus dem Application Builder heraus hat wahrscheinlich jeder schon mal eine Anwendung exportiert. Weniger bekannt ist, dass Apex auch ein Kommandozeilen-Werkzeug mitliefert, das nicht nur den Export einzelner Anwendungen, sondern auch den aller Anwendungen eines Workspace oder gar der Apex-Instanz erlaubt (siehe Listing 1). Das Aufsetzen eines Jobs, der täglich alle Anwendungen exportiert und ins Versions-Kontrollsystem eincheckt, ist dann nur noch eine Kleinigkeit. Das Kommandozeilen-Werkzeug muss allerdings erst eingerichtet werden. Wie das geht, ist in einem Tipp der deutschsprachigen Apex-Community unter <http://tinyurl.com/apxcm-dexp> beschrieben.

Ein weiteres Werkzeug ist der „Apex Export Splitter“, der eine Exportdatei in ihre Komponenten zerlegt – es entsteht eine Verzeichnisstruktur, in der eine ei-

KeepTool mit neuer Version 10

Das handliche Werkzeug für Oracle™-Datenbanken



Zahlreiche neue Funktionen, z.B.

- Darstellung Ihrer Daten als Pivottabelle, ggf. mehrstufig.
- Praktische Hinweistexte bei der Datenerfassung.
- Überwachung und Steuerung der Optimizer-Statistiken.
- Data Pump-Schnittstelle.
- Jumplist für den Windows 7™ Taskbar.

Laden Sie die kostenlose Testversion unter www.keeptool.com herunter.



keeptool

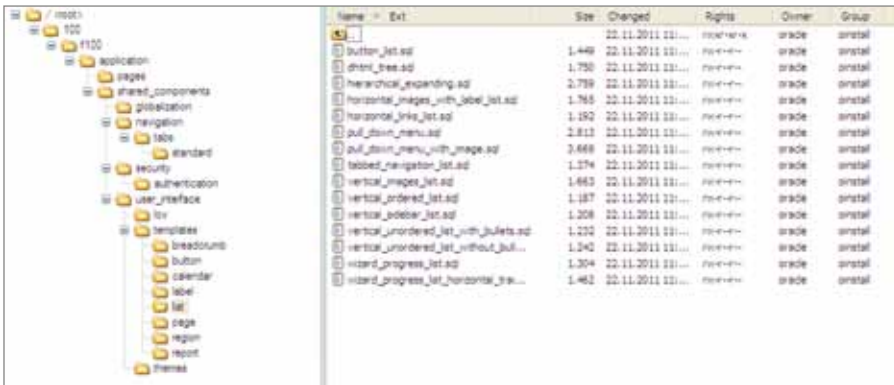


Abbildung 7: Der Apex Export Splitter zerlegt eine Export-Datei

```
SQL> begin
  2 apex_application_install.set_workspace_id(3138423103500405);
  3 apex_application_install.set_application_id(9999999);
  4 apex_application_install.set_schema('SCOTT');
  5 -- Offset auf eine zufällige große Zahl setzen ...
  6 apex_application_install.set_offset(2763898718743);
  7 end;
  8 /
```

PL/SQL procedure successfully completed.

```
SQL> @app_17378/pages/page_00021.sql
...PAGE 21: Interaktive Berichte
```

```
SQL> commit;
```

Commit complete.

Listing 2: Eine Apex-Seite wird mit SQL*Plus erfolgreich in eine andere Anwendung importiert

gene Exportdatei für jede Apex-Komponente enthalten ist (siehe Abbildung 7). Auch dies ist im Tipp der deutschsprachigen Apex-Community näher erklärt.

Kombiniert man dies mit dem erwähnten täglichen Exportieren und Einchecken ins Versions-Management, so bekommen Apex-Entwickler Zugriff auf tägliche Versionsstände für einzelne Apex-Komponenten. Das Wiederherstellen einer versehentlich gelöschten Apex-Komponente ist damit überhaupt kein Problem mehr.

In diesem Zusammenhang ist das in Apex 4.0 eingeführte PL/SQL-Paket „Apex_APPLICATION_INSTALL“ zu erwähnen. Wie die meisten Entwickler wissen, ist eine Apex-Exportdatei ein SQL-Skript, das prinzipiell auch mit SQL*Plus eingespielt werden kann. Sobald man jedoch versucht, eine Anwendung in einen anderen Workspace

oder eine Anwendungskomponente in eine andere Anwendung zu importieren, schlägt ein solcher Import mit einer Fehlermeldung fehl.

„Apex_APPLICATION_INSTALL“ erlaubt es, für einen Import mit SQL*Plus die Workspace-ID, die Applikations-ID und andere Einstellungen vor dem Import zu setzen. Diese Einstellungen gehen den in der Exportdatei enthaltenen Einstellungen vor. Listing 2 zeigt, wie eine Anwendungsseite aus einer anderen Anwendung erfolgreich mit SQL*Plus in die Anwendung „999999“ importiert werden kann.

Voraussetzung für die Nutzung von „Apex_APPLICATION_INSTALL“ ist allerdings, dass die Apex-Exportdatei mit der Version 4.0 oder höher erstellt wurde. Ist sie älter, sollte sie zunächst in Apex 4.0 oder höher importiert und daraus wieder exportiert werden.

URL-Syntax

Als Apex-Entwickler baut man fast ständig Apex-URLs nach dem Schema „f?p=“ zusammen. Und auch hier hat man mehr Möglichkeiten, als den meisten Entwicklern bewusst ist. So stehen viele Entwickler vor dem Problem, den Inhalt eines Elements per URL setzen zu wollen, was unproblematisch ist, jedoch soll im Inhalt ein Doppelpunkt oder ein Komma enthalten sein. Beides sind allerdings spezielle Zeichen – so ist der Doppelpunkt das Trennzeichen für die einzelnen URL-Bestandteile.

Mit dem zusätzlichen URL-Parameter „p_sep“ kann man das Trennzeichen selbst definieren. Die Übergabe des Inhalts „Name: Apex“ per URL für das Element „P1_PRODUKT“ funktioniert dann wie in Listing 3.

```
f?p=100!1!&SESSION.!!!P1_
PRODUKT!Produkt:Apex&p_sep=!
```

Listing 3: Apex-URL mit einem anderen Trennzeichen

Dem Komma, mit dem die Werte für mehrere Elemente voneinander getrennt sind, kann ein Backslash (\) vorangestellt werden. Es gibt also kaum Einschränkungen für die Zeichen, die per URL übergeben werden können.

Wenn man den URL-Parameter „p_trace“ auf „YES“ stellt, erzeugt Apex für das Seiten-Rendering eine SQL-Trace-Datei. Diese wird in der „user_dump_destination“ des Datenbank-Servers abgelegt und enthält sehr detaillierte Informationen zu den einzelnen SQL- und PL/SQL-Anweisungen, die von Apex ausgeführt wurden. Typischerweise wird „p_trace“ bei der Diagnose von Performance-Problemen eingesetzt.

Ab Apex 4.0 kann man die Sprach-einstellung mit dem URL-Parameter „p_lang“ einstellen – das ist wesentlich einfacher als das umständliche Einstellen der „Language Preference“ im Browser. Mit „p_lang“ lassen sich verschiedene URLs zur Anwendung direkt so konstruieren, dass fremdsprachige Nutzer gleich in der richtigen Sprache landen.



Abbildung 8: Die Powersuche

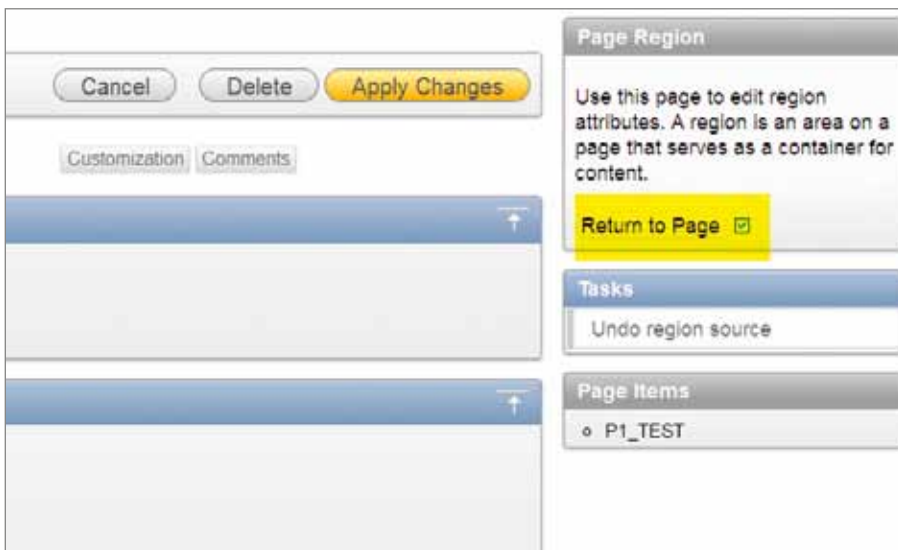


Abbildung 9: Return to Page: Beim Speichern der Änderungen landet man wieder auf der gleichen Seite



Abbildung 10: Exportieren aus der Vergangenheit

Wertelisten

Generell sollten Wertelisten möglichst dynamisch definiert sein, denn statische Wertelisten existieren nur in Apex und können nur im Application Builder angepasst werden. Ein Fremdschlüssel, der die Werteliste auch im Datenmodell durchsetzt, ist logischerweise ebenfalls nicht möglich. Standardbeispiele für statische Wertelisten sind „Ja/Nein“ oder „Frau/Herr“, wobei sich über letzteres Beispiel sicherlich streiten lässt, denn es gibt ja noch weitere Anredeformen.

Bei der Definition einer statischen Werteliste werden die einzelnen Einträge durch ein Komma getrennt, Anzeige- und Rückgabewert innerhalb eines Eintrags durch ein Semikolon. Beim Schlüsselwort „STATIC“ sortiert Apex die Einträge automatisch, bei „STATIC2“ sind die Einträge in der definierten Reihenfolge dargestellt.

Die einzelnen Trennzeichen können jedoch selbst definiert werden: Sollen Komma oder Semikolon in den Einträgen zum Tragen kommen, braucht man andere Trennzeichen. Listing 4 zeigt Beispiele zur Definition statischer Wertelisten.

```
STATIC:Ja;Y,Nein;N
STATIC2:Ja;Y,Nein;N
STATIC(#,%):Ja,
wirklich!%Y#Nein; ich will
nicht
STATIC2(#,%):Ja,
wirklich!%Y#Nein; ich will
nicht
```

Listing 4: Syntax für statische Wertelisten

Der Application Builder

Mit Apex 4.0 wurde eine wertvolle Navigationshilfe in den Application Builder eingeführt. Oben rechts liegt das Eingabefeld für „Powersuche“ (siehe Abbildung 8). Mit dem hier eingegebenen Begriff wird die gesamte Anwendungsdefinition durchsucht. Das ist sehr nützlich, wenn man zum Beispiel wissen möchte, wo die Werteliste „LOV_KUNDEN“ verwendet wird. Die Powersuche kann aber noch mehr: Gibt man zum Beispiel die Apex-Syn-

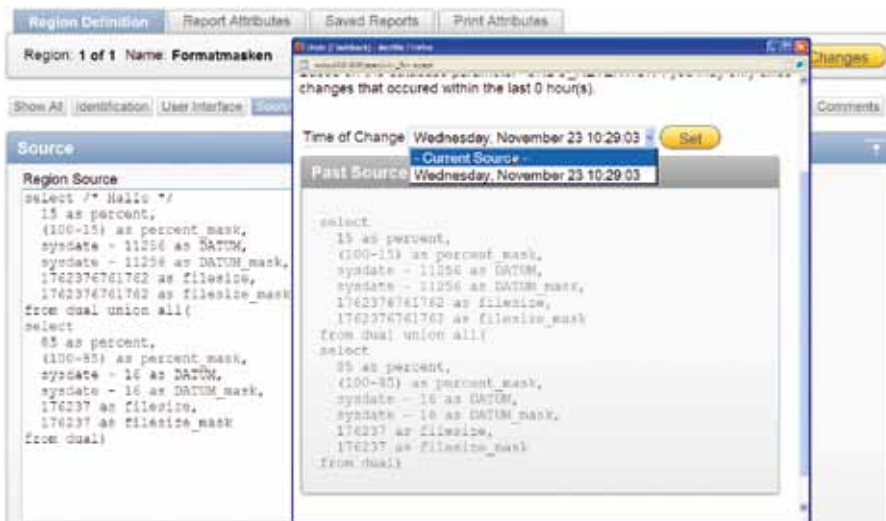


Abbildung 11: Änderungen an der Regionsquelle rückgängig machen

tax „[Anwendungs-ID]:[Page-ID]“ ein, so navigiert der Application Builder direkt dorthin. Das klappt natürlich nur dann, wenn sich die Anwendung im gleichen Workspace befindet, andernfalls landet man auf der Login-Seite.

Vielfach ist man in der Situation, das Aussehen eines Templates oder einer Region ändern zu wollen. Dabei ist es üblich, dass man sich die durchgeführten Änderungen immer wieder ansieht. Typischerweise ist die Anwendung dabei in einem zweiten Browser-Fenster geöffnet.

Das in Abbildung 9 dargestellte „Return to Page“ ist hier extrem nützlich – bei Anklicken wird man nach dem Speichern der Änderungen nicht mehr zurück zur Seitenübersicht, sondern nochmals zur gleichen Seite geführt. Man kann also direkt die nächste Änderung machen. Das spart eine Menge Klicks bei der Arbeit.

Blick in die Vergangenheit

Apex nutzt sehr intensiv das Flashback Query. Damit kann man eine Anwendung so exportieren, wie sie beispielsweise vor fünf Minuten aussah (siehe Abbildung 10). Wie weit man in die Vergangenheit zurückgehen kann, hängt von der Konfiguration der Datenbank (des UNDO-Tablespace) ab, Zeiträume bis circa zehn Minuten sollten aber auf nahezu allen Systemen möglich sein. Mit einem solchen Ex-

port lassen sich versehentlich gelöschte Komponenten einfach wiederherstellen – vorausgesetzt, man bemerkt den Fehler schnell genug.

Undo-Funktionen mithilfe von Flashback sind auch in den Dialogen zur Regionsbearbeitung vorhanden. Hat man sein Berichts-SQL gerade umfangreich geändert und es funktioniert nicht mehr, kommt man sehr einfach zur alten Version zurück. Klickt man auf der rechten Seite „Undo Region Source“ an, so erscheint der in Abbildung 11 dargestellte Dialog.

Flashback Query ist auch bei interaktiven Berichten standardmäßig aktiviert. Endanwender können sich die Daten also ansehen, so wie sie vor wenigen Minuten aussahen.

Fazit

Apex bietet inzwischen eine Fülle von Funktionen und Möglichkeiten an. Wie immer gibt es dabei Funktionen, die in Vorträgen, Konferenzen und Howto-Dokumenten im Mittelpunkt stehen, und solche, die kaum bekannt sind. Gerade diese sind aber oft sehr nützlich und deren Kenntnis kann im Bedarfsfall einiges an Entwicklungsaufwand einsparen.

Weitere Informationen

Auf den Webseiten von Oracle und im Internet ist umfangreiches Materi-

al auch in deutscher Sprache vorhanden:

1. Deutschsprachige Apex-Community: <http://tinyurl.com/apexcommunity>
2. Apex im OTN: <http://otn.oracle.com/apex>
3. Oracle-Datenbank auf Twitter: <http://twitter.com/OracleBUDB>

Carsten Czarski
 ORACLE Deutschland B.V. & Co. KG
 carsten.czarski@oracle.com
<http://twitter.com/cczarski>
<http://sql-plsql-de.blogspot.com>



Unsere Inserenten	
BERENBERG BANK www.berenberg.de	S. 15
Hunkler GmbH & Co. KG www.hunkler.de	S. 3
WIN-Verlag GmbH & Co. KG win-verlag.de	U 2
imining gmbh www.imining.de	S.13
KeepTool GmbH www.keeptool.com	S. 31
Libelle AG www.libelle.com	S. 7
MuniQsoft GmbH www.muniqsoft.de	S. 55
ORACLE Deutschland B.V. & Co. KG www.oracle.com	U 3
PROMATIS software GmbH www.promatis.de	S. 19
Trivadis GmbH www.trivadis.de	U 4
usb GmbH www.usb-muc.com	S. 23

Mit dem Itemtyp „Pop-up-List of Values (LOV)“ stellt Apex eine Möglichkeit zur Verfügung, um den Benutzer einen Wert aus dem Datenbestand zur Verwendung in einem Formular auswählen zu lassen.

Komfortable LOVs für Apex

Michael Schmid, Trivadis GmbH

Für viele einfache Anwendungsfälle ist der Mechanismus ausreichend. Im Vergleich zu der analogen Funktionalität von Oracle Forms wird der Entwickler allerdings die Möglichkeit, mehrere Werte an das aufrufende Formular zurückzugeben, vergeblich suchen. Dieser Artikel beschreibt eine Methode, wie sich komplexe und benutzerfreundliche Pop-up-LOVs in Apex realisieren lassen, die den Vergleich mit den LOVs von Oracle Forms nicht zu scheuen brauchen.

Bei der Erstellung benutzerfreundlicher Formulare ergibt sich häufig die Problematik, dass dem Benutzer zur Pflege einer Fremdschlüssel-Beziehung eine komfortable Möglichkeit zur Auswahl des entsprechenden Masterschlüssels geboten werden sollte. Zudem sollen fast immer im Formular zusätzliche Informationen zum jeweiligen Schlüssel angezeigt werden. Ein solches Apex-Formular, das auf dem EMP/DEPT-Schema basiert, könnte etwa wie in Abbildung 1 dargestellt aussehen.

Leider lässt sich das Füllen des „Deptno“-Schlüssels und der zusätzlichen Informationen („Dname“, „Loc“) dazu nicht mit dem im Lieferumfang von Apex befindlichen Pop-up-LOV-Mechanismus realisieren, denn damit ist es nur möglich, ein einziges Feld zu füllen. Außerdem lassen sich in dem Standard-Pop-up zwar einfache Filterungen durchführen, die Anzeige beschränkt sich jedoch auf nur eine einzige Spalte (siehe Abbildung 2).

Erstellen der Pop-up-Seite

Zur Lösung dieses Problems erstellen wir zunächst eine eigene Apex-Seite 210, die wir aus der Formularseite 200 heraus als Pop-up-Seite öffnen und die

einen interaktiven Report (IR) enthält, der zur Auswahl des Master-Datensatzes dient. Die Filtermöglichkeiten der interaktiven Reports von Apex sind so umfangreich und komfortabel, dass damit auch anspruchsvolle Benutzer zufrieden sind. Die Abfrage, die dem Report zugrunde liegt, sollte alle Spalten selektieren, die wir später verarbeiten beziehungsweise zurückgeben wollen. In unserem Fall der DEPT-Tabelle wählen wir mit „select deptno, dname, loc from dept“ alle Spalten.

Als Page-Template nehmen wir für diese Seite „Popup“ und für die Report-Region empfiehlt sich das Region-Template „Region without Title“. Zusätzlich erstellen wir im IR eine „Link Column“, die den Aufruf der (noch

von uns zu implementierenden) JavaScript-Funktion „passBack“ bewirkt und dabei die Werte des jeweiligen Datensatzes übergibt (siehe Abbildung 3).

Verbinden des Formulars mit der Pop-up-Seite

Um die Verbindung zwischen der Seite mit dem Formular und der Pop-up-Seite herzustellen, machen wir uns eine sehr nützliche JavaScript-Funktion namens „showModalDialog“ zunutze. Diese Funktion ist unter HTML 5 spezifiziert, funktioniert allerdings glücklicherweise schon seit geraumer Zeit in allen verbreiteten Browsern. So unterstützt der Internet Explorer diese Funktion schon seit Version 4, Firefox

Abbildung 1: Eingabemaske für einen EMP-Datensatz mit Zusatz-Informationen zum Department

Abbildung 2: Apex „Pop-up LOV“

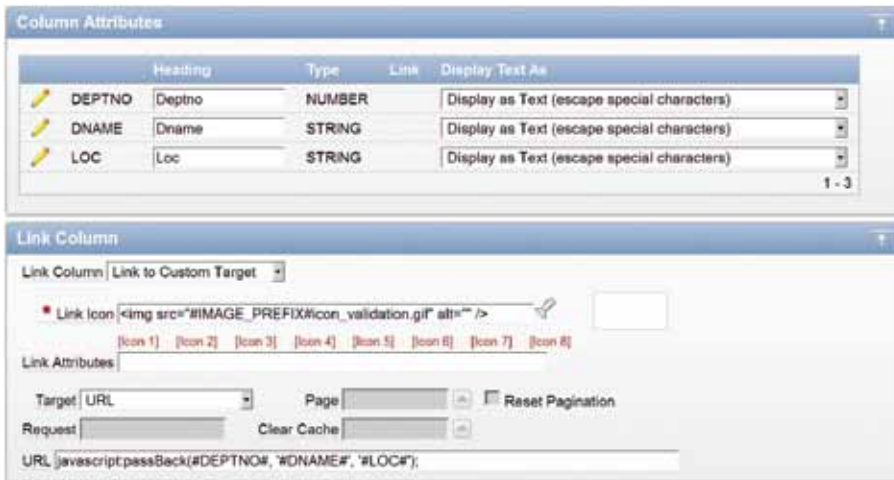


Abbildung 3: Die Einstellungen für die „Link Column“ des interaktiven Reports

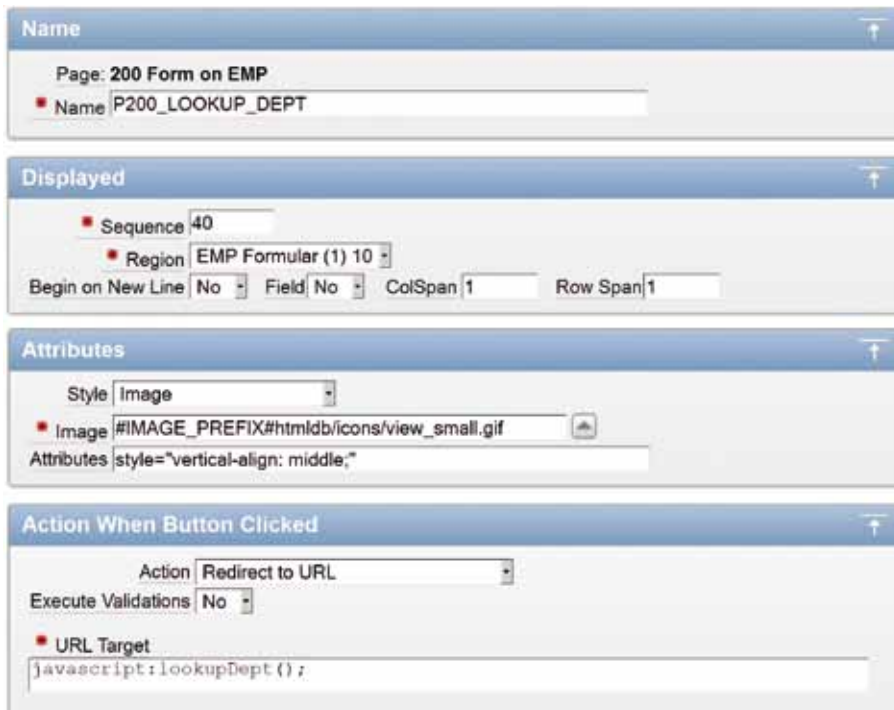


Abbildung 4: Die Einstellungen für den Button zum Öffnen der Pop-up-Seite

```
function lookupDept() {
    var ret =
        window.showModalDialog(
            „f?p=“ + $(„pFlowId“) + „:210:“ + $(„pInstance“),
            null,
            „dialogWidth: 620px; dialogHeight: 420px; „ +
            „resizable: yes“
        );
    if(ret) {
        $s(„P200_DEPTNO“, ret.deptno);
        $s(„P200_DNAME“, ret.dname);
        $s(„P200_LOC“, ret.loc);
    }
}
```

Listing 1: JavaScript-Funktion zum Öffnen des Pop-up-LOVs

beherrscht sie seit Version 3 und aktuelle Versionen von Chrome und Safari kommen damit ebenfalls zurecht (siehe auch die Links am Ende dieses Artikels).

Die Funktion „showModalDialog“ ermöglicht es, eine Web-Seite als modalen Dialog zu öffnen. Die URL der zu öffnenden Seite wird als erster Parameter übergeben. Mithilfe des zweiten Parameters lässt sich ein (komplexes) JavaScript-Objekt übergeben und mit dem dritten Parameter kann das Erscheinungsbild des neuen Fensters beeinflusst werden. Als Rückgabewert kann zum Beispiel sowohl ein String als auch ein komplexes Objekt geliefert werden, das die Pop-up-Seite generiert hat.

Auf der Formulareseite 200 erstellen wir zum Öffnen der Pop-up-LOV-Seite 210 unter „Function and Global Variable Declaration“ eine Funktion „lookupDept“ mit dem entsprechenden Code aus Listing 1.

Die Bedeutung der Item-Namen auf der Formulareseite 200 lässt sich leicht erschließen. Durch den Aufruf von „showModalDialog“ wird die Seite 210 in einem modalen Fenster geöffnet, das heißt der Benutzer kann erst wieder zum Formular zurückkehren, nachdem das Pop-up-Fenster verlassen wurde.

Zusätzlich legen wir die Größe des neuen Fensters mit 620 x 420 Pixeln fest und gestatten dem Benutzer, die Fenstergröße anzupassen. Da wir in diesem Fall von der Möglichkeit, ein Objekt zu übergeben, keinen Gebrauch machen, ist der zweite Parameter auf „null“ gesetzt. Die Ausführung des JavaScript-Codes blockiert am Aufruf von „showModalDialog“ und wartet, bis das Pop-up-Fenster geschlossen wird.

Das Rückgabeobjekt wird in der lokalen Variablen „ret“ gespeichert. Falls ein Wert zurückgegeben wurde, lesen wir die Werte „deptno“, „dname“ und „loc“ aus dem Objekt aus und legen sie in den entsprechenden Items ab.

Der Aufruf der Funktion „lookupDept“ erfolgt über einen Item-Button, den wir beispielsweise direkt neben dem Item „P200_DEPTNO“ platzieren (siehe Abbildungen 1 und 4).

Rückgabe der gewählten Werte

Nun müssen wir uns noch um die Rückgabe der Werte von der Pop-up-Seite 210 zurück zur Formularseite 200 kümmern. Dazu erstellen wir auf der Pop-up-Seite unter „Function and Global Variable Declaration“ eine Funktion namens „passBack“, auf die wir ja schon in unserem IR in der „Link Column“ referenzieren (siehe Listing 2).

Die Funktion erstellt aus den übergebenen Parametern ein JavaScript-Objekt und legt es in dem Property „returnValue“ des Window-Objekts ab. Anschließend wird das Pop-up-Fenster geschlossen. Der Wert des Property „returnValue“ wird an den Aufrufer zurückgegeben – in unserem Fall also an die lokale Variable „ret“ der Funktion „lookupDept“ auf der Formularseite. Empfehlenswert ist zusätzlich noch die Erstellung eines Buttons zum Schließen des Pop-up-Fensters, ohne eine Auswahl treffen zu müssen (siehe Abbildung 5). Diesen Button platzieren wir rechts neben der Suchleiste des IR (siehe Abbildung 6).

Weitere Ausbaumöglichkeiten

Die Interaktion zwischen unserer Formularseite und der Pop-up-LOV-Seite ist nun komplett implementiert – zusätzlich könnten wir unsere Pop-up-LOV-Seite noch um die Fähigkeit erweitern, neue Departments anzulegen. Ein solches Feature ist bei den Anwendern gern gesehen. Da es sich bei dieser Seite 210 um eine vollwertige Apex-Seite handelt, erstellen wir dazu eine Formularregion („Neues DEPT“) mit den benötigten Items („P210_DEPTNO“, „P210_DNAME“, „P210_LOC“) und legen einen entsprechenden Prozess („Process Row of DEPT“) an, der die eigentliche Speicherung übernimmt. Ferner ist noch ein Branch zurück zur selben Seite nötig (siehe Abbildung 7).

Zusätzlich sind noch zwei Änderungen beziehungsweise Erweiterungen am Page-Template „Popup“ vorzunehmen. Um zu verhindern, dass sich beim Internet Explorer durch das Abschicken des Formulars ein neues Fenster öffnet, stellen wir sicher, dass

Abbildung 5: Die Einstellungen für den Button zum Schließen der Pop-up-Seite

Deptno	Dname	Loc
✓ 10	ACCOUNTING	NEW YORK
✓ 20	RESEARCH	DALLAS
✓ 30	SALES	CHICAGO
✓ 40	OPERATIONS	BOSTON

Abbildung 6: Die Pop-up-Seite mit IR und Schließen-Button

```
function passBack(deptno, dname, loc) {
    window.returnValue =
        { deptno: deptno, dname: dname, loc: loc };
    window.close();
}
```

Listing 2: JavaScript-Funktion zur Rückgabe der Werte

```
<div class="success" id="success-message" style="width: 400px;">
  
  #SUCCESS_MESSAGE#
</div>
```

Listing 3: Sub-Template „Success Message“

der HTML-Head-Tag den Code „<base target=“_self“>“ enthält. Zum Schluss passen wir noch die Sub-Templates „Success Message“ (siehe Listing 3) und „Notification“ (siehe Listing 4) an. Damit ist schließlich unser Pop-up-LOV für die Departments fertig (siehe Abbildung 8).

Fazit

Mit der beschriebenen Methode ist es möglich, benutzerfreundliche LOVs in Apex zu erstellen und der wahrscheinlich größte Vorteil dabei ist: Überall in unserer Applikation, wo wir ein Look-up zum Beispiel für ein Department benötigen, können wir dieselbe Pop-up-Seite wiederverwenden und auch der Aufruf mit „showModalDialog“ ließe sich in eine JavaScript-Bibliothek auslagern. Einzig die weitere Verarbeitung des zurückgegebenen Objekts ist seitenspezifisch zu implementieren. Und dies eröffnet die Möglichkeit, die Resultate etwa in Items oder in Felder eines tabellarischen Formulars zu schreiben oder andere komplexere Verarbeitungsschritte mit JavaScript durchzuführen.

Weitere Informationen

- [1] Dokumentation zur JavaScript-Funktion „showModalDialog“ im Mozilla Developer Network: <https://developer.mozilla.org/en/DOM%3Awindow.showModalDialog>
- [2] Dokumentation zur JavaScript-Funktion „showModalDialog“ im Microsoft Developer Network: [http://msdn.microsoft.com/en-us/library/ms536759\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536759(VS.85).aspx)

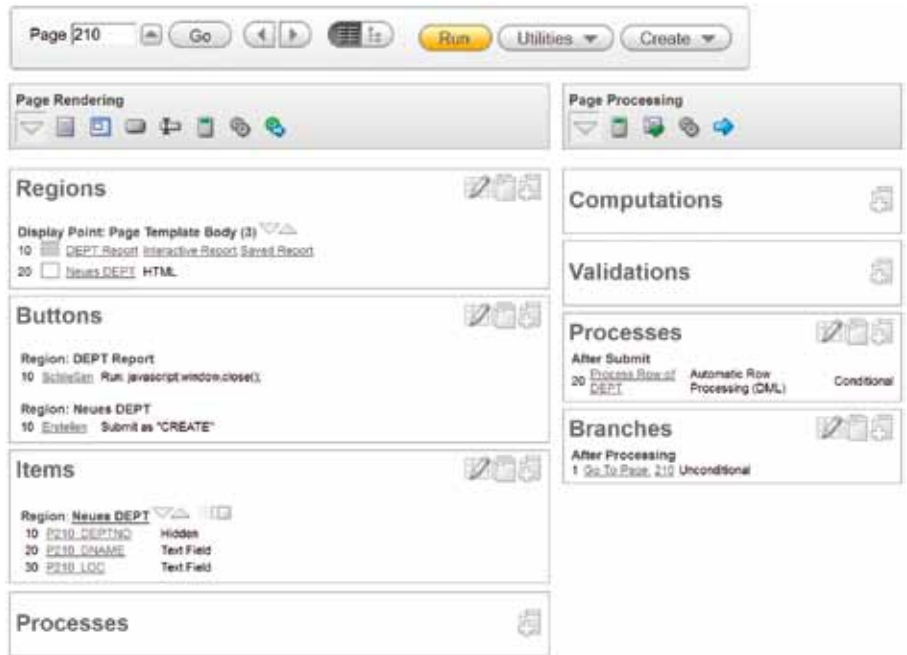


Abbildung 7: Die Seitendefinition der Pop-up-Seite mit den benötigten Komponenten zum Erstellen eines Departments



Abbildung 8: Das fertige Pop-up-LOV nach der Erstellung eines neuen Departments

Michael Schmid
Trivadis GmbH
michael.schmid@trivadis.com



```
<div class="notification" id="notification-message" style="width: 400px;">
  
  #MESSAGE#
</div>
```

Listing 4: Sub-Template „Notification“

Das Apex-Framework bietet Entwicklern bereits eine beträchtliche Anzahl von Seitenelementen wie „Shuttle“ oder „Date Picker“. Leider gehören Dropdown-Menüs nicht zum Standard-Funktionsumfang. Einzellösungen für solche Menüs in Web-Anwendungen gibt es in vielfältiger Ausprägung. Dieser Artikel zeigt einen generischen Ansatz für die Verwendung von Dropdown-Menüs in Apex-Anwendungen.

Einsatz von Dropdown-Menüs in Apex-Anwendungen

Artur Rollow, OPITZ CONSULTING GmbH

Eine Link-Spalte ist typischerweise ein Bestandteil jedes Interactive Reports. Verbreitet sind beispielsweise Löschk-Aktionen oder die Navigation auf Detailseiten. Für den Anwender ist dies praktisch, da er die gewünschte Aktion direkt auf der Übersichtsseite ausführen kann und nicht auf eine Detailseite navigieren muss. In der Vergangenheit forderten die Anwender deshalb häufig eine Möglichkeit zur Ausführung mehrerer Aktionen pro Datensatz. Erfahrungsgemäß sollten aber maximal zwei Link-Spalten pro Report verwendet werden. Kommt es zur Überschreitung dieses Wertes, wirkt die Funktionsauswahl für den Anwender überladen und unüberschaubar. Zusätzlich verdrängen die Link-Spalten die eigentlichen Report-Daten auf die rechte Seite des Monitors. Eine Alternative hierzu bietet ein sogenanntes „Dropdown-Menü“. In diesem erfolgt eine Zusammenfassung aller Datensatz-Aktionen. Der Vorteil liegt darin, dass der Einsatz von nur einer Link-Spalte notwen-

dig ist. Ein Dropdown-Menü ist keine Standard-Komponente der Apex-Entwicklungsumgebung. Der Artikel stellt deshalb einen eigenen Lösungsansatz vor.

Das vorgestellte Dropdown-Menü (siehe Abbildung 1) wurde im Rahmen eines Projekts zur Verbesserung der Benutzerfreundlichkeit einer Apex-Anwendung erfolgreich eingesetzt. Die Anforderungen an das Menü sind:

- *Wiederverwendbarkeit*
Der entwickelte Quellcode soll für beliebig viele Menüs verwendbar sein. Neue Menüs sollen deklarativ erstellt werden.
- *Konfigurierbarkeit*
Änderungen an der Beschriftung der Einträge sollen ohne Entwickler stattfinden
- *Untermenüs*
Um das Dropdown-Menü möglichst kompakt zu halten, sollten Untermenüs möglich sein
- *Performance*
Das Dropdown-Menü soll sich im Browserfenster flüssig aufbauen

empfängt vom Server den generierten HTML-Code und stellt ihn in dem Browserfenster dar.

2. Die serverseitige Komponente besteht aus weiteren Unterkomponenten – dem Anwendungsprozess und einer „Pipelined Table Function“. Serverseitig wird der HTML-Code für das Menü erstellt. Die Menü-Attribute sind in den Datenbank-Tabellen konfigurierbar hinterlegt. Unter anderem wird eine ausführbare Aktion festgelegt. Diese definiert, was bei einem Klick passiert, beispielsweise EDIT, DELETE etc.

Um den gesamten HTML-Code der Seite nicht unnötig aufzublähen und somit eine gute Performance zu gewährleisten, muss das Dropdown-Menü für den angeklickten Datensatz zur Laufzeit generiert werden.

JavaScript-Funktion

Beim Klick auf ein Menü-Icon wird der clientseitigen Komponente mitgeteilt, welches Menü für welchen Datensatz aufgerufen werden soll. Hierfür wurde die JS-Funktion „ApexMenu.generateMenu“ (siehe Listing 1) erstellt. Folgende Parameterliste liegt vor: Mit „pThis“ wird der Primärschlüssel des angeklickten Datensatzes übergeben. „pMenuId“ identifiziert ein konkretes Menü. Der Client kommuniziert mit dem Server über einen Ajax-Request, um den HTML-Code für das Menü erzeugen zu lassen. Die Verwendung der asynchronen Datenübertragung verhindert beim Klick auf ein Menü-Icon ein erneutes Laden der Seite.

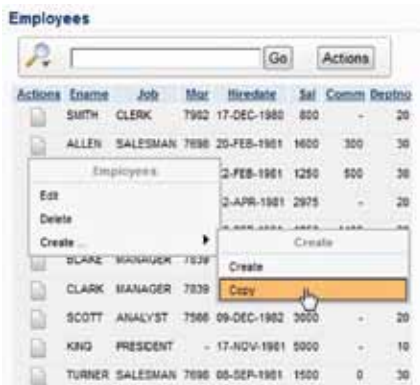


Abbildung 1: Dropdown-Menü im Einsatz

Technische Realisierung

Zur Umsetzung der Anforderungen wurden zwei Komponenten implementiert:

1. Mittels einer clientseitigen Komponente, einer JavaScript-Funktion, wird ein Request an den Server abgesetzt, mit dem eine Prozedur zum Erzeugen des Menü-HTML-Codes aufgerufen wird. Die JS-Funktion

Auf der Serverseite erfolgt der Aufruf des Apex-Anwendungsprozesses „GENERATE_MENU“. Dabei wird die aktuelle Parameterliste der JS-Funktion an den Datenbank-Server weitergereicht. Damit im Server die Parameter verwendet werden können, sind zwei Anwendungselemente angelegt (F101_MENU_ID und F101_OBJ_ID). Der erzeugte HTML-Code wird mit der jQuery-Syntax „\$(, #body').append(gReturn)“ dem aktuellen DOM-Baum hinzugefügt. Durch den weiteren Funktionsaufruf „ApexMenu.appMenuMultiOpen“ wird das Dropdown-Menü im Browser-Fenster dargestellt. Aus Gründen der Performance lohnt sich die Überlegung, beim erneuten Aufruf desselben Menüs den bereits vorliegenden HTML-Code wiederzuverwenden statt diesen über einen Ajax-Request neu zu generieren.

Anwendungsprozess

Der Aufbau des serverseitigen Anwendungsprozesses „GENERATE_MENU“ (siehe Listing 2) ist übersichtlich. Es findet ein Aufruf der Pipelined Table Function „GEN_MENU“ statt. Dabei werden die vom Client übergebenen Parameter mithilfe der Anwendungselemente weitergereicht. Die Menü-Einträge werden zeilenweise zurückgeliefert. Die gesamte Ergebnismenge wird zu einem kompletten Dropdown-Menü zusammengefasst und mit dem „HTP.PRN“-Aufruf zurück an den Client übergeben.

Table Function

Das Listing für den nachfolgenden Text steht unter <http://www.doag.org/go/doagnews/drop> zum Download bereit. Die Table Function „GEN_MENU“ (siehe Listing 3) stellt die Haupt-Komponente des gesamten Dropdown-Mechanismus dar. Jede mit „PIPE ROW“ zurückgelieferte Zeile entspricht einem HTML-Tag. Im PL/SQL-Code ist dafür eine festdefinierte HTML-Vorlage mit „- und „-Elementen hinterlegt. Die endgültige Struktur des Menüs hängt von der aktuellen Parameterliste ab. „MENU_ID“ legt die Anzahl der Menü-Einträge fest, die aus der Tabelle „APEX_MENU“ aus-

```
var ApexMenu = function () {
    return {
        generateMenu: function (pThis, pMenuId) {
            var get = new htmldb_Get(null, &APP_ID., ,APPLICATION_PROCESS=GENERATE_MENU', 0);
            get.add(,F101_OBJ_ID', pThis.id);
            get.add(,F101_MENU_ID', pMenuId);
            gReturn = get.get();
            $(, #body').append(gReturn);

            var dropdown_id = pMenuId + pThis.id;
            setTimeout(, ApexMenu.appMenuMultiOpen($x(\'' + pThis.id + ,\'), \\' + dropdown_id + ,\', \\'Bottom\')', 100);
            get = null;
        },
        appMenuMultiOpen: function (pThis, pThat, pPlacement) {
            var l_that = $x(pThis);
            var lMenu = $x(pThat);
            if (lMenu) {
                if (l_that != gCurrentAppMenuImage) {
                    app_AppMenuMultiClose();
                    pThis.className = g_dhtmlMenuOn;
                    dhtml_MenuOpen(l_that, pThat, false, pPlacement);
                    gCurrentAppMenuImage = l_that;
                } else {
                    dhtml_CloseAllSubMenus();
                    app_AppMenuMultiClose();
                }
            }
        }
    };
}();
```

Listing 1: JavaScript-Funktion mit Ajax-Request

```
DECLARE
    menu_html VARCHAR2 (3000);
BEGIN
    FOR item IN (SELECT *
                FROM TABLE (GEN_MENU (
                    :F101_MENU_ID,
                    2,
                    :F101_OBJ_ID)))
    LOOP
        menu_html := menu_html || item.COLUMN_VALUE;
    END LOOP;
    HTP.prn (menu_html);
END;
```

Listing 2: Anwendungsprozess mit dem Aufruf der Pipelined Table Function

gelesen werden. Die OBJ_ID, Primärschlüssel des angeklickten Datensatzes, ersetzt den Platzhalterstring aus dem Action-Eintrag, zum Beispiel wird „apex.submit(DELETE?#1#)“ zur Laufzeit zu „apex.submit(DELETE?4712)“. Beim Aufbau eines Untermenüs erfolgt ein rekursiver Aufruf der Table Function. Dies wird mit der Spalte „CHILD_

MENU_ID“ gesteuert. Für das Aussehen sowie das Ein- und Ausblenden des Menüs sorgen die in dem Apex-Framework vorhandenen CSS- und JS-Klassen.

Beispiel-Menü

Mit der Implementierung der beschriebenen Komponenten ist das Grund-

DOAG 2012 Development

Bonn | 14. Juni 2012

ERSTMALS

Fachkonferenz für
Erfahrungsaustausch
unter Software-
Entwicklern.

Software-Entwicklung auf Basis von Oracle Tools und Technologien – Wohin geht die Reise?

Veranstaltungs-Schwerpunkte

- PL/SQL und Apex
- Java und Open Source
- Forms, Reports, ADF, BI-Publisher
- BPM und Softwarearchitektur



DOAG

Deutsche ORACLE-Anwendergruppe e.V.

Infos unter development.doag.org



professional
it software &
services **pitss**[®]

iJUG
Verbund

SOUG
Swiss Oracle User Group

ORACLE[®]

BASIS
International

OPITZ CONSULTING

t&p software.
our profession.

XDEV³
Rapid Java Development

gerüst für die Verwendung der Drop-down-Menüs in Apex-Anwendungen geschaffen. Um ein Beispiel-Menü zu erstellen, genügt es, Menü-Einträge (siehe Listing 4) in der Tabelle „APEX_MENU“ (siehe Listing 3) abzuspeichern und den Interactive Report um eine Link-Spalte zu erweitern.

Zum Einbau des Dropdown-Menüs in den Interactive Report wird das vorhandene SQL-Statement (siehe Listing 5) mit einem „“-Tag um eine Link-Spalte ergänzt. Der „“-Tag enthält einen „onclick“-Event, sodass beim Anklicken des Menü-Icons die clientseitige JS-Komponente aufgerufen wird. Mit „this“ wird der Parameterliste immer der aktuelle „“-Tag als Objekt übergeben. Da das Attribut „id“ des Tags den Wert des Primärschlüssels enthält, kann in der JS-Funktion der aktuelle Datensatz ermittelt werden. Der zweite Parameter identifiziert das Dropdown-Menü in der Datenbank-Tabelle.

Weitere Features

Der vorliegende Code lässt sich mit wenigen Schritten um weitere Features wie Mehrsprachigkeit und bedingtes Anzeigen der Menü-Einträge erweitern. Das bedingte Anzeigen der Menüeinträge kann mit einer zusätzlichen „CONDITION“-Spalte in der „APEX_MENU“-Tabelle umgesetzt werden. Pro Menü-Eintrag wird eine Bedingung in SQL hinterlegt. Das SQL-Statement wird mit „EXECUTE IMMEDIATE“ zur Laufzeit ausgeführt. Wurde die Bedingung nicht erfüllt, wird der Menü-Eintrag ausgegraut und somit nicht ausführbar. Ähnlich wie bei der Action kann der Primärschlüssel durch einen Ersetzungsstring innerhalb der Bedingung verwendet werden.

Fazit

Das vorgestellte Dropdown-Menü dient als sinnvolle Ergänzung zu den vorhandenen Apex-Bordmitteln. Trotz der Erstellung des Menüs zur Laufzeit und des asynchronen Request bietet dieser Lösungsansatz eine sehr gute Performance. Nach einem einmaligen Aufwand für die Implementierung der

```
CREATE TABLE APEX_MENU
(
  , MENU_ID VARCHAR2(50) NOT NULL
  , CHILD_MENU_ID VARCHAR2(50)
  , ORDER_NO NUMBER(1) NOT NULL
  , ITEM_TEXT VARCHAR(50) NOT NULL
  , ACTION VARCHAR2(255)
);
```

Listing 3: Tabellenstruktur für Menüs

```
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Employees',null,1,'Edit','f?p=&APP_ID.:2:&APP_
SESSION.:NO::P2_EMPNO:#1#');
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Employees',NULL,2,'Delete','javascript:apex.
submit(''DELETE?#1#'')');
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Employees','Create',3,'Create',null);
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Create',null,1,'Create','javascript:apex.
submit(''CREATE?#1#'')');
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Create',null,2,'Copy','javascript:apex.
submit(''COPY?#1#'')');
```

Listing 4: Insert-Statements für Beispiel-Menü

```
SELECT ,'
AS Action,
ENAME,
JOB,
MGR,
HIREDATE,
SAL,
COMM,
DEPTNO
FROM EMP;
```

Listing 5: Erweiterung des SQL-Statements

notwendigen Komponenten lässt sich in kurzer Zeit ein Menü in eine Berichtssseite integrieren. Die client- und serverseitigen Komponenten können beliebig um weitere Funktionalitäten erweitert werden.

Artur Rollow
OPITZ CONSULTING GmbH
artur.rollow@opitz-consulting.com



Die datenbankzentrierte Web-Entwicklungsplattform Apex erfreut sich steigender Beliebtheit. Dabei wird sie nicht nur zur Neuentwicklung, sondern auch zur Erweiterung bestehender Applikationen eingesetzt. Eine häufige Anforderung in diesem Zusammenhang ist das Prinzip der Einmal-Anmeldung: Hat sich der Nutzer an der Hauptanwendung authentisiert, soll eine separate Anmeldung an der Apex-Anwendung entfallen. Dafür wird eine mögliche Lösung vorgestellt.

SSO zwischen Eclipse RCP und Apex: Vertrauen auf den ersten Blick

Anton Schlegel, PROMATIS software GmbH

Im vorliegenden Kapitel stellt der Horus Business Modeler, ein Werkzeug zur Geschäftsprozess-Modellierung (GPM), die zu erweiternde Lösung dar. Es nutzt eine Oracle-Datenbank und basiert auf der Eclipse Rich Client Plattform (RCP). Das Werkzeug soll mithilfe von Apex um „Governance, Risk and Compliance“-Funktionalitäten (GRC) erweitert werden. Eine wichtige Anforderung ist dabei das Single-Sign-on (SSO). Das bedeutet, dass sich der Nutzer nach einer einmaligen Anmeldung am GPM-Werkzeug nicht zusätzlich an der Apex-Anwendung authentisieren muss. Nach einer Erläuterung der technischen Grundlagen werden die Funktionsweise der Lösung sowie die Methoden vorgestellt, die bei der Implementierung zum Einsatz kommen. Es folgt eine detaillierte Beschreibung der technischen Umsetzung.

Technische Grundlagen

Eclipse bietet mit RCP eine modulare und erweiterbare Plattform zur Entwicklung von Desktop-Anwendungen an. Die entwickelten Anwendungen sind von der Eclipse Integrated Development Environment (IDE) unabhängig. Um SSO zwischen Eclipse RCP und Apex umzusetzen, ist es wichtig zu verstehen, wie der Zugang zu den Apex-Anwendungen geregelt ist.

Den Zugang zu einer Apex-Applikation regelt ein sogenanntes „Authentifizierungsschema“. Dieses kann unter anderem auf Datenbank-Accounts oder auf Oracle-SSO-Servern basie-

ren. Die erste Möglichkeit hat zur Folge, dass der Entwickler eine/mehrere Tabelle(n) mit Benutzernamen/Passwörtern pflegen muss. Die zweite erfordert eine Zusatzkomponente in Form eines Oracle-SSO-Servers. Bei der vorgestellten Lösung kommt dagegen eine dritte Variante zum Einsatz: ein benutzerdefiniertes Authentifizierungsschema. Dieses muss eine PL/SQL-Funktion (Authentifizierungsfunktion) enthalten, die die Logik dafür implementiert, wem der Zugang zur Anwendung gewährt wird und wem nicht. Der Artikel zeigt, wie die zu implementierende Logik ausgestaltet werden kann beziehungsweise

welche weiteren Schritte in Apex und Drittanwendungen nötig sind.

Konzept

Wie in Abbildung 1 zu erkennen ist, wird Apex von der Eclipse-Anwendung heraus mithilfe einer URL aufgerufen. Diese URL beinhaltet Parameter, die der Nutzer-Identifikation dienen. Daraufhin erfolgt eine Umleitung zur Login-Seite. Ein Seitenprozess auf der Login-Seite extrahiert die Parameter aus der URL und übergibt sie an die Authentifizierungsfunktion. Diese soll sowohl die Benutzer des GPM-Werkzeugs als auch diejenigen, die die

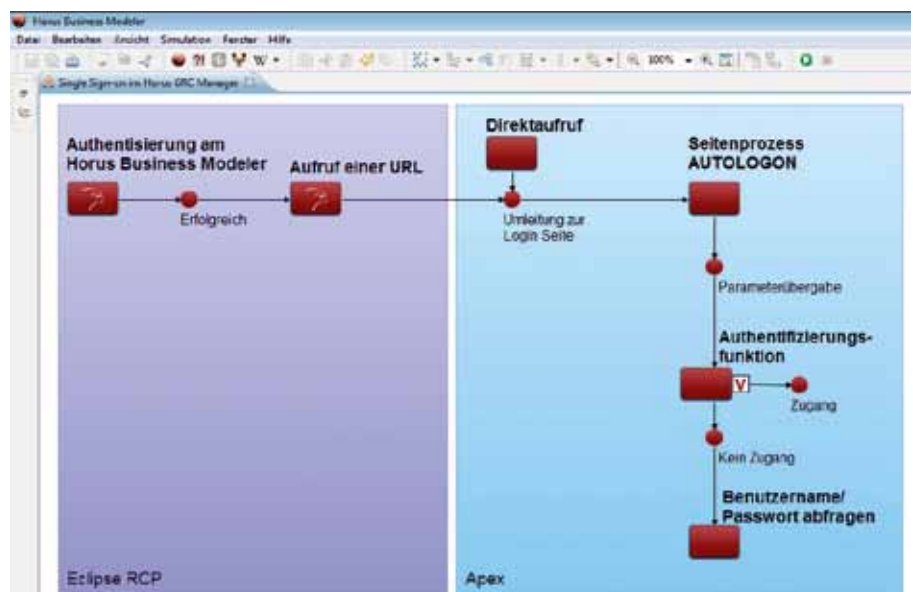


Abbildung 1: SSO-Konzeption

Apex-Anwendung direkt aufrufen, authentifizieren können.

Technische Umsetzung

Wie bereits erwähnt, werden an die Apex-Anwendung bestimmte Parameter in einer URL zur Identifikation des Nutzers übergeben. Hierbei handelt es sich zum einen um den Login-Namen des Benutzers und zum anderen um einen Hash-Wert beziehungsweise eine Prüfsumme.

Der Datensatz, der „gehasht“ wird, besteht aus dem Login-Namen, einer Sitzungs-ID und einem Zeitstempel. Die Sitzungs-ID identifiziert eindeutig einen Nutzer des GPM-Tools, der aktuell am Werkzeug angemeldet ist. Der Zeitstempel sorgt dafür, dass die Prüfsumme nach Ablauf einer definierten Zeit nicht mehr gültig ist. Die Anmeldung an der Apex-Komponente ist ohne Eingabe von Benutzername/Passwort somit nur innerhalb eines bestimmten Zeitfensters möglich.

Die Bildung der Prüfsumme führt die Eclipse-Anwendung durch. Eine mögliche Ausgestaltung lässt sich in diesem Pseudocode beschreiben: „md5_hash_function(username, session_id, timestamp)“. Eine URL, mit der jede Apex-Anwendung aufgerufen wird, ist so aufgebaut: „https://<Host>:<Port>/apex/f?p=<App>:<Seite>::<Request>::<Parameterbezeichnung>:<Parameterwert>“. Konkret könnte ein Aufruf in der Eclipse-Anwendung so aussehen: „https://horus-grc-manager:8080/apex/f?p=24313:4::AUTOLOGON::: P101_UNAME,P101_UPASS:tester1,2BC5617650F41F4A969DB4D8871256D2“.

Darüber hinaus erfordert die Umsetzung des Konzepts eine benutzerdefinierte Authentifizierungsfunktion. Diese bildet einen Hashwert nach dem gleichen Muster wie die Eclipse-Anwendung, wofür sich die Prozedur „DBMS_OBFUSCATION_TOOLKIT.MD5“ eignet. Diese setzt folgende Logik um: „Liefere dann „true“ zurück, wenn der erzeugte Hash-Wert mit dem übergebenen Hash-Wert übereinstimmt oder wenn die Kombination Benutzername/Passwort von Apex als gültig verifiziert wird“. Die Validierung des Passworts wird durch den Aufruf

```

DECLARE
v_mixed VARCHAR2(4000);
v_user VARCHAR2(4000);
v_pass VARCHAR2(4000);
v_fsp_after_login_url VARCHAR2(4000) := :FSP_AFTER_LOGIN_URL;
BEGIN
SELECT SUBSTR(v_fsp_after_login_url, instr(v_fsp_after_login_
url, 'P101_UNAME,P101_UPASS')+
LENGTH('P101_UNAME,P101_UPASS')+1)
INTO v_mixed
FROM dual;
SELECT SUBSTR(v_mixed, 1, instr(v_mixed, ',')-1) INTO v_user FROM
dual;
SELECT SUBSTR(v_mixed, instr(v_mixed, ',')+1) INTO v_pass FROM
dual;
wwv_flow_custom_auth_std.login(
P_UNAME => v_user,
P_PASSWORD => v_pass,
P_SESSION_ID => v('APP_SESSION'),
P_FLOW_PAGE => :APP_ID||':1'
);
END;

```

Listing 1

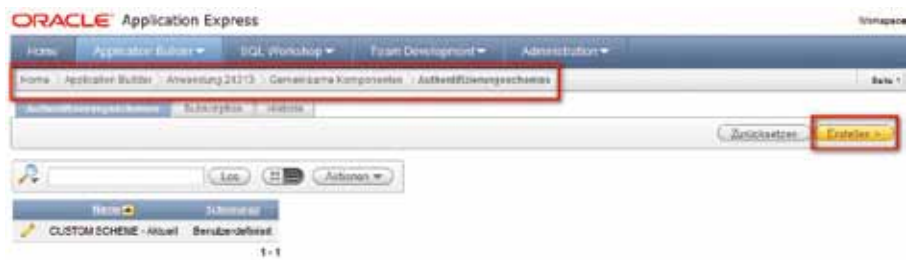


Abbildung 2: Authentifizierungsschema erstellen

der Prozedur „APEX_UTIL.IS_LOGIN_PASSWORD_VALID“ umgesetzt.

In einem weiteren Schritt ist ein Authentifizierungsschema zu erstellen, das die Funktion nutzt. Das Vorgehen unterscheidet sich je nach Apex-Version geringfügig. Abbildung 2 bezieht sich auf die Version 4.1.

Das erstellte Authentifizierungsschema ist als aktuell zu markieren, damit es von der Applikation genutzt wird. Nach seiner Erstellung wird die Login-Seite der Apex-Anwendung angepasst. Auf dieser Seite ist ein Prozess „autologon“ zu erstellen. Es muss sichergestellt sein, dass der Seitenprozess auf der Login-Seite als erstes ausgeführt wird, damit der Nutzer die Login-Seite nicht sieht (vorausgesetzt die Parameter in der URL sind valide). Der Seitenprozess soll „Vor Header“ und mit der niedrigsten Sequenz ausgeführt werden. Zu beachten ist, dass dieser Prozess nur ausgeführt wird, falls die

Anforderung (siehe „<Request>“ in der URL oben) dem Ausdruck „AUTOLOGON“ entspricht (siehe Abbildung 3).

So wird beim Aufruf der oben erwähnten URL der Prozess „autologon“ ausgeführt. Da die Ausführung jedoch bereits vor dem Header-Aufbau stattfindet, sind die Elemente „P101_UNAME“ und „P101_UPASS“, die entsprechend den Benutzernamen und die Prüfsumme beinhalten, noch nicht gesetzt. Daher müssen die Werte erst aus der Apex-Anwendungsvariablen „FSP_AFTER_LOGIN_URL“ mithilfe der Funktionen „instr“ und „substr“ extrahiert werden. Die extrahierten Werte übernimmt dann die Prozedur „wwv_flow_custom_auth_std.login“, die dann im Hintergrund die Authentifizierungsfunktion aufruft (siehe Listing 1).

Abschließend ist zu erwähnen, dass als Übertragungsprotokoll „HTTP over SSL“ zum Einsatz kommt. Sonst könnte ein potenzieller Angreifer die



Abbildung 3: Bedingungen für die Prozess-Ausführung

auf den Einsatz mit Eclipse RCP beschränkt und lässt sich somit auch auf weitere Projekte übertragen.

Parameter abfangen und sich ohne Kenntnis des Passworts mit dem Login-Namen und dem Hash-Wert an der Apex-Anwendung authentisieren. Eine verschlüsselte Übertragung verhindert derartige Angriffe.

Fazit

Mit der vorliegenden Lösung wurde ein einfacher Mechanismus zur Ein-

mal-Anmeldung zwischen Anwendungen basierend auf Eclipse RCP und Apex-Applikationen aufgezeigt. Dabei ist weder die Pflege zusätzlicher Tabellen noch eine weitere Komponente wie der Oracle-SSO-Server notwendig. Ebenso wurde die Sicherheit insofern berücksichtigt, als die Anwendungen über eine verschlüsselte Verbindung miteinander kommunizieren. Die vorgestellte Vorgehensweise ist nicht

Anton Schlegel
 PROMATIS software GmbH
 anton.schlegel@promatis.de



DOAG 2012 Applications
 Konferenz für Oracle Applications Anwender in Europa
8. – 10. Mai 2012
RAMADA Hotel Berlin-Alexanderplatz

applications.doag.org

Sponsoren

Goldsponsoren

		In Kooperation mit				
--	--	--------------------	--	--	--	--

Premiumsponsor

Die vielen Vorteile von Apex können die Fachabteilung eines Unternehmens dazu bewegen, eine Fachanwendung zur Personalverwaltung oder für andere sensible Firmendaten zu entwickeln. Spätestens dann kommt die Frage der Sicherheit einer Apex-Anwendung auf. Der Artikel beschreibt die Sicherheitsaspekte der verschiedenen Installationsvarianten von Produktiv-Umgebungen sowie wichtige Apex-Sicherheitskonzepte und deren Konfiguration.

Sicherheitsaspekte einer Apex-Installation

André Schulte, Freiberufler

Eine Apex-Anwendung kann grundsätzlich genauso sicher oder unsicher wie Applikationen basierend auf anderen Technologien betrieben werden. Je beliebter und verbreiteter eine Technik ist, desto größer wird der Personenkreis, der sich damit beschäftigt, Sicherheitslücken zu finden und auszunutzen. Apex ist genauso wie ein JEE- oder .NET-basiertes System „out of the box“ nur unzureichend gesichert und muss an Sicherheitsanforderungen angepasst werden.

Das Sicherheitsniveau des Gesamtsystems wird durch das schwächste Glied der Kette bestimmt. Dabei müssen auch das Betriebssystem, die Datenbank und das Netzwerk betrachtet werden. Das Sicherheitskonzept muss eine Sicherheits-Maintenance vorsehen. Diese definiert Aufgaben wie regelmäßige Patches, Durchsicht von Auditlogs oder Überprüfung von Rechten. Um die Komplexität einzuschränken, geht der Artikel nur auf die spezifischen Aspekte einer Apex-Sicherheitskonfiguration für eine Produktivumgebung ein. Zunächst werden allgemeine Bedrohungen einer Apex-Anwendung aufgezeigt, die grundsätzlich für alle Web-Anwendungen relevant sind.

Bedrohungen einer Apex-Anwendung

Es gibt folgende Szenarien:

- **SQL-Injection**
Der Angreifer ändert den SQL-Befehl und injiziert ein geändertes Verhalten. Dies kann zur Zerstörung der Datenbank oder zum Lesen von geschützten Daten führen. SQL-Injection entsteht durch Zeichenketten-

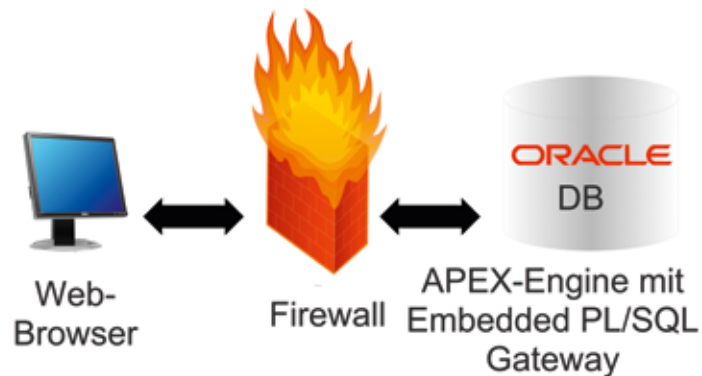


Abbildung 1: Apex-Installation mit Embedded PL/SQL Gateway

Konkatenation zu einem SQL-Befehl, bei der der Angreifer Parameter so übergibt, dass ein in seinem Sinne geänderter SQL-Befehl ausgeführt wird. Schutz bieten Bindevariablen oder, falls dies nicht möglich ist, eine Prüfung der Variablen durch das DBMS_ASSERT-Package.

- **Cross-Site Scripting (CSS)**

Ein Schadcode wird bei einer Webseite so eingeschleust, dass er auf der Client-Seite ausgeführt wird. Man unterscheidet persistente und nicht persistente Angriffe. Ein Beispiel für einen nicht persistenten Angriff ist eine präparierte URL mit einem Parameter, der per HTTP-GET-Aufruf übergeben wird. Der Browser zeigt den übergebenen Parameter ungeprüft auf der Webseite an und kann beispielsweise einen Script-Aufruf verursachen. Bei einem persistenten Angriff wird der Schadcode zum Beispiel als Blog-Eintrag in der Datenbank gespeichert und als Teil der Webseite angezeigt. Durch die Anzeige im Client-Browser wird der Schadcode ausgeführt und kann beispielsweise JavaScript-Aufrufe verursachen.

- **URL Manipulation (URL-Tampering)**

Es wird eine URL manipuliert, sodass die Web-Anwendung Inhalte anzeigt, die der Nutzer eigentlich nicht sehen soll.

In der Regel werden mit HTTP-GET übergebene Parameter abgeändert und beispielsweise eine andere Personen-ID für die Anzeige an den Server gesendet. Gegen solche Angriffe hilft die Session State Protection, die weiter unten im Artikel erläutert wird.

- **Netzwerk**

Durch das Abhören von Netzwerk-Paketen kann man an Personen-IDs, Passwörter und andere Informationen, nach denen man gezielt sucht, kommen. Auch das Manipulieren oder wiederholte Absenden von Netzwerk-Paketen sowie eine „Man in the Middle“-Angriffe müssen beachtet werden. Eine Verschlüsselung der HTTP-Verbindungen mit Secure Sockets Layer (SSL) als HTTPS schützt gegen diese Angriffe. Bei hohen Sicherheitsanforderungen kann dafür ein Zertifikat von einer Zertifizierungsstelle genutzt werden.

Installationsvarianten

Bei der Installation von Apex unterscheidet man die Installation mit dem Embedded PL/SQL Gateway (EPG) sowie die mit dem Oracle HTTP-Server (OHS) und MOD_PLSQL. Darüber hinaus kann man mit dem Apex-Listener installieren.

Beim Embedded PL/SQL Gateway läuft der Oracle-XML-DB-HTTP-Server direkt in der Oracle-Datenbank 11g. In dieser Zwei-Tier-Architektur greift der Browser des Nutzers direkt auf die Datenbank zu. Es ergibt sich weniger Netzwerk-Overhead als bei den anderen Varianten und man benötigt keine separate Server-Installation. Von Seiten der Sicherheit aus betrachtet ist die Lösung für Produktionen nicht zu empfehlen, weil der HTTP-Listener und Apex in derselben Datenbank ausgeführt werden. Es ist nicht möglich, sie zu trennen. Die Datenbank kann, wie in Abbildung 1 dargestellt, nicht hinter einer Firewall separiert werden und ist für Angreifer zugänglicher.

MOD_PLSQL ist ein Apache-Webserver-Erweiterungsmodul, mit dem man aus PL/SQL-Packages und Stored Procedures dynamische Webseiten erzeugen kann. Der Vorteil ist die Trennung des Apache-Webserver von der Datenbank, sodass die Datenbank hinter einer Firewall geschützt ist und die Sicherheit erhöht wird. Der Apex-Listener läuft in einem Java-Container. Er verbindet sich über JDBC mit der Datenbank. Listener und Datenbank können durch eine Firewall getrennt wer-

den, damit die gleiche Sicherheit wie bei Nutzung der MOD_PLSQL-Lösung entsteht (siehe Abbildung 2).

Der Apex-Listener hat jedoch noch weitergehende Sicherheitsfunktionen, die ihn zur sichersten Apex-Installationsvariante machen. Es gibt Listen für Allowed- und Blocked-Procedures, die für Prozeduren, Packages und Schema-Namen verwendet werden können. Die „Database Validation Function“ ist eine Funktion in der Datenbank, die prüft, ob eine angefragte Prozedur ausgeführt werden soll. Sie kann einen Cache auf der Listener-Seite nutzen, um den Datenbank-Roundtrip zu verhindern. Bei einer Anfrage wird zuerst die Validate-Funktion geprüft, dann die Allowed-List und zum Schluss die Blocked-List. Erst danach erfolgt der eigentliche Aufruf der Prozedur. Der Apex-Listener beinhaltet außerdem Statistiken, ein Fehler-Tracking-System und eine Protokollierung. Die Protokollierung zeigt Informationen wie geänderte Cache-Einträge der Validate-Funktion oder die Ausführungszeit der Prozedur.

Database Access Descriptor (DAD)

Wenn man keinen Apex-Listener verwendet, dann kann man den Zugriff auf die Datenbank mit dem Database Access Descriptor (DAD) einschränken. Mithilfe der Funktion „`www_flow_epg_include_mod_local`“ kann man per String-Vergleich eine benutzerdefinierte Funktion autorisieren. Die Funktion gibt „true“ für eine er-

folgreiche Autorisierung, ansonsten „false“ zurück. Listing 1 zeigt die Autorisierung der Funktion „myfunction“.

IP-Whitelist, HTTPS, File Upload, Session Timeout und Passwörter

Mit der Einstellung „Restrict Access by IP Address“ kann man den Zugang zur Apex-Instanz auf IP-Basis beschränken. Um den Parameter zu ändern, meldet man sich als Apex-Admin über die Web-Oberfläche an und geht auf „Manage Instance -> Security“. Dort kann unter „Restrict Access by IP Address“ eine kommaseparierte Liste, auch unter Nutzung von Wildcards, eingegeben werden.

Mit dem Setzen des „Require HTTPS“-Attributs auf „yes“ aktiviert man das HTTPS-Protokoll. Daten werden verschlüsselt über das Internet zum Browser des Clients übertragen. Die Einstellung kann im INTERNAL-Workspace unter „Manage Instance -> Security“ vorgenommen werden. Durch Konfigurieren des „Allow Public File Upload“-Attributs auf „no“ kann man verhindern, dass nicht authentifizierte Nutzer in Datei-Upload-Steuer-elementen Dateien hochladen können. Damit nicht geschlossene Browser beziehungsweise Nutzer, die sich nicht aus der Applikation ausloggen, ein geringeres Sicherheitsrisiko darstellen, ist es sinnvoll, den Parameter „Session Timeout“ festzulegen. Man kann dort angeben, nach wie vielen Sekunden eine ungenutzte Session abläuft und nach wie vielen Sekunden eine Session

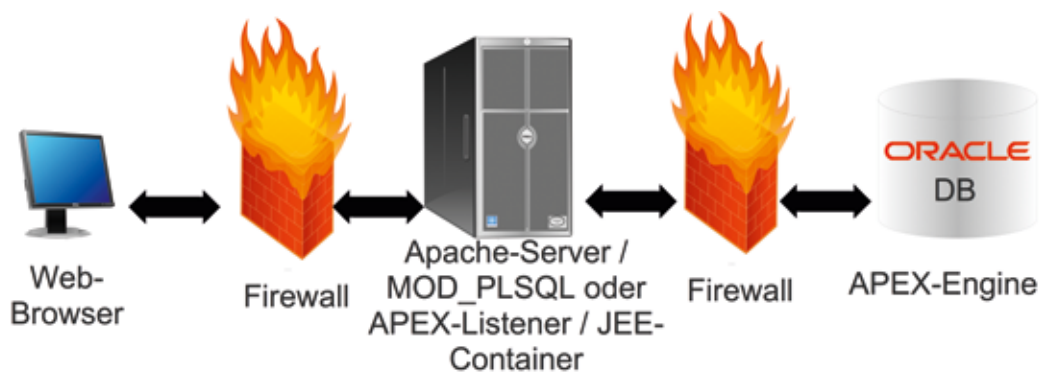


Abbildung 2: Apex-Installation mit MOD_PLSQL und Apex-Listener


```

CREATE OR REPLACE function wwv_flow_epg_include_mod_local( procedure_name in varchar2)
return boolean
is
begin
  if upper(procedure_name) in ('Apex_040100.MYFUNCTION','MYFUNCTION') then
    return TRUE;
  else
    return FALSE;
  end if;
end wwv_flow_epg_include_mod_local;
/

```

Listing 1

```

create table testapp.geheim as select * from (
select object_name, object_type ,
ROW_NUMBER() OVER(PARTITION BY object_type ORDER BY object_type, object_name) myindex
from dba_objects where object_type in (,TABLE', ,FUNCTION', ,PROCEDURE') order by object_type, object_
name) where myindex <= 5
/

```

Listing 2

beendet werden soll. Der Entwickler einer Apex-Applikation kann für die Nutzer eine Passwort-Policy festlegen. Dies ist sinnvoll, damit Nutzer nicht zu einfache Passwörter verwenden. Beispiele für Regeln sind die Verwendung mindestens eines Großbuchstaben oder mindestens einer Zahl und das Vorschreiben einer Mindestlänge des Passworts von sechs Buchstaben. Die Sicherheit kann weiterhin erhöht werden, indem das Wiederverwenden von Passwörtern unterbunden wird. Diese Funktionalität wird unter „sqlplus“ als Apex-Schema-Owner (Apex_040100) beispielsweise für sechzig Tage aktiviert: „Apex_INSTANCE_ADMIN.SET_PARAMETER(PASSWORD_HISTORY_DAYS',60);“.

Session State Protection

Mit der Verwendung von HTTPS können Pakete unterwegs nicht mehr manipuliert werden. Der Nutzer kann aber vor dem Absenden in der Browserzeile einen Parameter, etwa für eine Personen-ID oder eine Mandanten-ID, manipulieren. Apex speichert Session-Parameter serverseitig und persistent in der Datenbank. Aus diesem Grund sollte man sensitive Daten in der Session in verschlüsselter Form speichern. Im Application Builder kann, wie in Abbildung 3 dargestellt, unter

„Shared-Components -> Session State Protection“ über einen Wizard der Schutz auf verschiedenen Ebenen aktiviert werden.

Man unterscheidet zwischen „Application Item Protection“ zum Schutz individueller Items und „Page Access Protection“, um die ganze Seite zu schützen. Bei „Page Protection“ wählt man im Wizard für die Seite „Arguments Must Have Checksum“. Weitere Einstellungen sind „No Arguments Allowed“, „No URL Access“ oder „Unrestricted“. Eine mit „Arguments Must Have Checksum“ geschützte Seite hat beim Aufruf in der URL als letzten Parameter „cs=<checksum>“. Der Nutzer kann die anderen Parameter in der URL sehen, aber eine Änderung wird an der falschen Checksumme erkannt und führt zu einem Fehler. Auf diese

Weise wird das sogenannte „URL-Tampering“ verhindert.

Session State Protection und VPD

Eine Session State Protection schützt vor Manipulationen von Seiten des Clients. Ein SQL-Befehl, der eine Sicherheitslücke bei der Session State Protection findet, wird auf der Datenbank ungehindert ausgeführt. Ein weiteres Problem ist, dass eine komplexe Anwendung aufgrund eines Entwicklungsfehlers mit einem der Reports ungewollt zu viele Daten anzeigt. Die Qualitätssicherung hat den Report aber nie mit den Parametern des Anwenders, der die Lücke ausnutzt, aufgerufen. Um bei einer Lücke in der Session State Protection oder bei einem Anwendungsfehler dennoch sensible

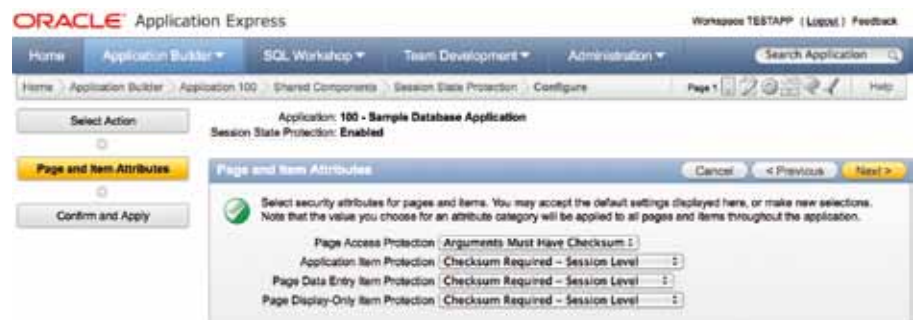


Abbildung 3: Session State Protection gegen URL-Manipulation

```

Grant EXECUTE ON DBMS_RLS TO TESTAPP; (als Nutzer sys)
Grant CREATE ANY CONTEXT TO TESTAPP; (als Nutzer sys)

create or replace context apex_context using vpd_context
/
create or replace package vpd_context is
  procedure logonUser(userid varchar2);
end vpd_context;
/

create or replace package body vpd_context is
  procedure logonUser(userid varchar2) is
  begin
    dbms_session.set_context(,apex_context', 'userid', userid);
  end logonUser;
end vpd_context;
/

create or replace function pol_geheim(obj_owner in varchar2, obj_name in varchar2) return varchar2
as
  predicate varchar2(200) := ',1=2';
begin
  if SYS_CONTEXT(,apex_context', 'userid')= ,ADMIN' then
    predicate := ',';
  elsif SYS_CONTEXT(,apex_context', 'userid')= ,TABLE_ADMIN' then
    predicate := ',OBJECT_TYPE='TABLE'';
  elsif SYS_CONTEXT(,apex_context', 'userid')= ,PROCEDURE_ADMIN' then
    predicate := ',OBJECT_TYPE='PROCEDURE'';
  end if;
  return predicate;
end;
/
BEGIN
DBMS_RLS.ADD_POLICY(
object_schema => ,testapp2',
object_name => ,geheim',
policy_name => ,secure_geheim',
policy_function => ,pol_geheim',
statement_types => 'SELECT');
END;
/

```

Listing 3

Daten zu schützen, kann zusätzlich zur Session State Protection die Virtual Private Database (VPD) genutzt werden. VPD ist nur mit der Enterprise Edition der Datenbank möglich und erzwingt den Schutz der Daten direkt bei der Ausführung des SQL-Befehls in der Datenbank.

Ein Beispiel zeigt, wie VPD mit Apex genutzt werden kann. Der grundsätzliche Gebrauch ist in der MyOracleSupport Note (ID 1352641.1) „Master Note For Oracle Virtual Private Database“ erläutert. Im Beispiel werden ein neuer Workspace namens „TESTAPP“ und in dem TESTAPP-Schema die Tabelle „geheim“ angelegt, die fünf Tabellen-, Funktions- und Prozedurnamen der

Datenbank aus dba_objects enthalten soll. Listing 2 zeigt den SQL-Befehl, der mit „sqlplus“ als „sys“ ausgeführt wird.

In Apex wird in der neuen Applikation ein einfacher Report für die Tabelle angelegt, bei dem die gesamte Tabelle mit folgendem SQL-Befehl selektiert wird: „select * from geheim“. In Apex sind die drei Nutzer „admin“, „table_admin“ und „procedure_admin“ für die Applikation angelegt. Der Nutzer „admin“ soll später alle Daten sehen können, „table_admin“ beziehungsweise „procedure_admin“ jedoch nur die Tabellen und Prozeduren aus der Tabelle „geheim“. Um VPD nutzen zu können, werden im TESTAPP-Schema ein Kontext angelegt, die Logon-Proze-

dur „vpd_context.logonUser“ für das Setzen des Kontext erstellt sowie die Policy-Funktion „pol_geheim“ für die Einhaltung der genannten Regeln implementiert (siehe Listing 3).

Die Funktion „ADD_POLICY“ aktiviert die Regel. Beim Einloggen als Nutzer „admin“ bekommt man aber noch keine Daten angezeigt, da der Kontext von Apex noch nicht gesetzt wird. Mit Apex kann man unter „Application Builder -> Edit Application Properties -> Security“ im Feld „Initialization PL/SQL Code“ die Funktion für das Setzen des Kontext konfigurieren: „vpd_context.logonUser(v('APP_USER'))“. Apex ersetzt „v('APP_USER)“ durch den tatsächlich einzuloggenden

Object Name	Object Type	Myindex	Object Name	Object Type	Myindex
ADD_JOB_HISTORY	PROCEDURE	1	ACCESS\$	TABLE	1
APEX	PROCEDURE	2	ACTION_TABLE	TABLE	2
APEX	PROCEDURE	3	ALERT_QT	TABLE	3
APEX_ADMIN	PROCEDURE	4	ANFRAGE	TABLE	4
APEX_ADMIN	PROCEDURE	5	ANFRAGEDATEN	TABLE	5

Abbildung 4: Reportdaten, links „procedure_admin“, rechts „table_admin“

Nutzer. Wenn man sich als Nutzer „table_admin“ oder „procedure_admin“ anmeldet und den Report aufruft, sieht man unterschiedliche Daten (siehe Abbildung 4). Die Regel wird allerdings nicht durch Apex, sondern durch die Datenbank bei der Ausführung des SQL-Befehls erzwungen.

Fazit

Die allgemeinen Bedrohungen einer Apex-Anwendung sind vielfältig. Genauso kreativ wie der Angreifer sollte daher auch das Sicherheitskonzept zur Abwehr von Angriffen sein. Ein ganzheitliches Konzept muss alle Sys-

tem-Komponenten berücksichtigen. Bereits bei der Entwicklung der Anwendung müssen Bedrohungen wie die SQL-Injection berücksichtigt und Bindevariablen genutzt werden. Eine Basis-Apex-Härtung umfasst die richtige Installationsvariante und das Setzen der Apex-Sicherheitsparameter. Der Aufwand für diese Maßnahmen ist überschaubar und der Sicherheitsgewinn groß. Weitere Maßnahmen zur Erhöhung der Sicherheit sind mit höherem Aufwand verbunden. Der Einsatz von Virtual Private Database erfordert beispielsweise eine Enterprise-Edition-Lizenz. Diese Lösung ist sinnvoll als zusätzlicher Schutz zur

Verhinderung der Einsicht in sensitive Daten unterhalb der Anwender (nicht dem Datenbank-Administrator).

André Schulte
contact@andschulte.de





Berliner Experten- seminare

- Wissensvertiefung für Oracle-Anwender
- Mit ausgewählten Schulungspartnern
- Von Experten für Experten
- Umfangreiches Seminarangebot

Termine

18./19. April 2012:
Oracle Apex mit Plugins erweitern
Referent: Peter Raganitsch

24./25. April 2012:
Oracle Security
Referent: Pete Finnigan (Ersatztermin)

23./24. Mai 2012:
Oracle VM 3.0
Referent: Martin Bracher

30./31. Mai 2012:
Oracle Partitionierung
Referent: Klaus Reimers

5./6. Juni 2012:
„Solaris 11 Zonen-deep drive“
Referent: Heiko Stein



Deutsche ORACLE-Anwendergruppe e.V.
www.doag.org

Der Artikel zeigt, welche Layouts mit Apex möglich sind, welche Einschränkungen existieren und worauf man achten sollte. Er gibt Antwort darauf, wie eine funktionierende Vermischung von Design und Funktionalität aussehen kann und wann der Einsatz von jQuery sinnvoll ist.

Layout und dynamische Elemente für Apex-Anwendungen

Markus Dötsch, MuniQSoft GmbH

Apex kommt bei immer mehr Firmen in den Fachbereichen, aber auch bei der zentralen IT-Abteilung an. Die Frage nach den möglichen Web-Layouts, die in einer Apex-Umgebung machbar sind, lässt sich einfach und schnell beantworten: Alles ist möglich – was auch mit anderen webbasierten Systemen machbar ist. Die Frage nach den Einschränkungen wirft sofort eine Gegenfrage auf: Welche Vorkenntnisse bringt man mit und welche Schwerpunkte sind wichtig?

Apex aus Sicht der Oracle-Welt

Oft kommen die Anforderungen aus einer Oracle-Umgebung. Dort ist man Client-Server und eventuell Forms gewohnt. Hier lässt sich die Frage oft umformulieren: Welche Einschränkungen bestehen in einem browserbasierten System?

Die Beschreibungssprache HTML ist per se zwar keine Einschränkung, bringt aber aus Sicht einer lokal installierbaren Anwendung mit einem klassischen User-Interface ein paar Hürden mit. Die Anordnung von Elementen fällt am Anfang schwer, der Workflow einer Applikation ist unübersichtlich und es gibt zu viele Einschränkungen bei der Entwicklung – auch wenn man später feststellt, dass die Funktionalität oft nicht fehlt, sondern nur an anderer Stelle auffindbar ist.

Apex fühlt sich für den Anwendungsentwickler genauso gut oder schlecht an wie jedes andere auf Web-Technologie getrimmte System. Die Funktionen, die man im Web schon an anderer Stelle entdeckt hat, lassen sich auch mit Apex umsetzen – wenn

nicht mit Bordmitteln, dann sicher durch die Integration anderer Systeme oder Bibliotheken. Apex ist, genauso wie die darunterliegende Oracle-Datenbank, flexibel anpassbar und versteht sich mit vielen Schnittstellen und Web-Standards.

Apex aus der Web-Sicht

Aus der Sicht eines erfahrenen Web-Entwicklers ist Apex in der Regel zuerst relativ unübersichtlich. Man versteht jedoch sehr schnell, wie die Abläufe in der Darstellung von Webseiten funktionieren, da sie ja nicht anders erfolgen als in der gewohnten Umgebung. Nur die Darstellung der Entwicklungsoberfläche sieht unterschiedlich aus. Sobald man in das Geheimnis um Themes und Templates eingeweiht ist, sind alle Möglichkeiten klar strukturiert.

Was man zu schätzen lernt, ist der einfache Umgang mit den Daten. Report und Formular sind in Windeseile erstellt. Ein an die Corporate Identity angepasstes Template lässt sich immer wieder verwenden. Alles funktioniert modular und logisch. Erste Erfolge sind schnell erreicht.

Gedanken zum Thema „Layout“

Oft heißt es zu Beginn eines Projekts, dass eine neue Apex-Anwendung so aussehen soll wie das Intranet oder die Webseite. Aber auch wenn man mit einer Apex-Anwendung auf einer „grünen Wiese“ beginnen kann, stellt sich vielleicht die Frage: Können wir uns ein Webseiten-Layout von einem Profi erstellen lassen?

Rein theoretisch ist das kein Problem, aber in der Praxis gibt es ein paar kleine Stolpersteine.

Grundsätzlich muss dabei zwischen dem Rahmen einer Webseite, also dem Kopf-, Fuß- und Seitenbereich auf der einen Seite sowie dem Inhalt einer Webseite auf der anderen Seite unterschieden werden. Für den Anwender gehören diese beiden Welten zusammen. In der Kombination kann der Betrachter entscheiden, ob die Seite gefällt oder nicht. Das Gerüst einer Webseite kann in Apex vollkommen kopiert werden. Aber eine auf Layout getrimmte Webseite, die von Marketing-Profis erstellt wurde, sieht in der Regel auch im Content-Bereich gut aus. Viele Bilder, wenig Text, alles aufeinander abgestimmt.

Bei der Apex-Anwendung wirkt das nicht ganz so gut: Der eingebundene Report ist durch die vielen Spalten viel zu breit, das dahinterliegende Formular viel zu schmal und zu hoch.

Sind Layout und Funktionalität nun Gegensätze oder doch Verbündete? Die folgenden Beispiele zeigen auf, wie Apex trotz Layout noch benutzbar ist, beziehungsweise wie gut Apex auch mit Funktionalität aussehen kann.

Es verschmelzen hier sehr schnell die Ideen hinter Layout und dynamischen Elementen. Es soll gut aussehen und trotzdem benutzbar sein. Also verwenden wir doch die Möglichkeiten, die uns das Web bietet. Ein Baustein dieser Lösung ist das JavaScript-Framework jQuery. Ab der Version 4 von Apex wird diese Bibliothek bereits mitgeliefert. Einige der von Apex mitgelieferten Themes nutzen die Möglichkeiten aus dieser Sammlung.

Beispiel: Breiter Report

Eine designte Webseite hat in der Regel eine feste Breite. Der heutige Standard beträgt 1024 oder 1280 Pixel, je nach Anforderung und Benutzergruppe. Oft werden die Seiten auch zentriert dargestellt. Was passiert mit der Webseite, wenn man einen interaktiven Report einer Tabelle mit 30 Spalten oder mehr hinzufügt?

Auch bei einem Standard-Report, bei dem der Anwender nicht in der Lage ist, die Spalten zu verändern, kann es zu einem nicht gewünschten Aussehen kommen. Denn auch hier gilt: Der Inhalt bestimmt die Tabellendarstellung. Was passiert, wenn lange Inhalte – eventuell noch ohne Leerzeichen, was einen Zeilenumbruch verhindert – die gewünschte Darstellung zerstören? Entweder das Layout der Apex-Anwendung wird „zerrissen“, was zur Folge haben kann, dass nachfolgende Elemente, wie zum Beispiel der Footer, rechts daneben dargestellt werden oder das Logo seinen vorgesehenen Platz verlässt. Andernfalls bleibt das Layout weiterhin stabil, dann werden Teile des Reports nicht zu sehen sein oder sie sind überdeckt von anderen Bereichen wie zum Beispiel einer Sidebar (siehe Abbildung 1).

Eine Lösung kann darin bestehen, dem Anwender die Möglichkeit zu geben, die aktuelle Seite wahlweise in der vollen Breite darzustellen. Dies er-

```
$(document).ready(function() {
    $('#schalter').hide();
    $('#ausblenden').click(function() {
        $('#sidebar').toggle();
        $('#schalter').toggle();
        $('#main').css('width', '100%');
    });
    $('#schalter').click(function() {
        $('#sidebar').toggle();
        $('#schalter').toggle();
        $('#main').css('width', '.816px');
    });
});
```

Listing 1

reicht man durch einen immer erreichbaren Schalter, der zwischen voller Breite und designtem Layout mit fester Breite umschaltet. Zusätzlich könnte man eine eventuell störende Side- oder Navigations-Bar ein- und ausblenden. Hier übernimmt jQuery diese ansprechende Aufgabe für uns. Die Elemente müssen eindeutig über IDs bezeichnet sein. Über die Seite 0 oder eine externe JavaScript-Datei, die über ein Template geladen wird, sorgen wir global dafür, dass die notwendigen Code-Zeilen auf jeder Seite zur Verfügung stehen. Soll die Funktion nur auf einer Seite zur Verfügung gestellt werden, können wir den Code auch in der Seite unter „JavaScript -> Execute when Page Loads“ eintragen (siehe Listing 1).

In diesem Beispiel ist der Schalter in die Sidebar integriert. Sobald man darauf klickt, wird die Sidebar ausgeblen-

det und das Layout auf 100 Prozent ausgedehnt. Damit der Anwender die Sidebar wieder einblenden kann, wird stattdessen eine kleine Grafik als Schalter dargestellt. Die jQuery-Funktion „toggle“ blendet Sidebar und Schalter jeweils ein oder aus.

Eine weitere Idee besteht darin, die grundsätzliche Darstellung einer Seite bereits bei der Auslieferung der Applikation festzulegen. Ist der Inhalt zu breit, wird standardmäßig die Sidebar ausgeblendet, der Anwender hat aber über den Schalter immer die Möglichkeit, sie wieder anzuschalten. Es muss dann nur dafür gesorgt werden, dass dann die Sidebar komplett überlappt und nicht durchsichtig ist. Zu guter Letzt kann noch die Einstellung pro Seite im Benutzerkonto abgespeichert werden, damit der Anwender die Seite so wiederfindet, wie er sie beim letzten Mal verlassen hat.

Eine andere Möglichkeit für einen zu breiten Report

Eine weitere Lösung für einen zu breiten Report besteht in der Möglichkeit, zusätzliche Spalten nur auf Anforderung des Anwenders per Klick darzustellen. Mit einem sogenannten „Named Column“-Template für einen Report können wir eine beliebige, selbst definierte Anordnung der Spalten eines Select-Statements in einem Report wählen. Dieser Vorteil wird mit dem Nachteil erkauft, dass mit dieser Technik für jedes Select-Statement ein eigenes Template erstellt werden muss, da im Template jede Spalte mit Namen angesprochen wird.

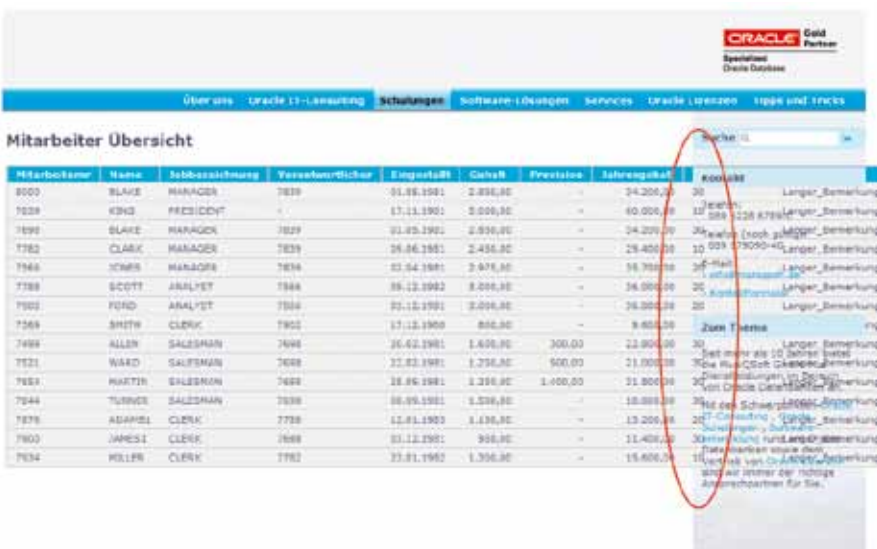


Abbildung 1: Da sich Report und Sidebar überlappen, ist der Text an dieser Stelle kaum mehr zu lesen



bi.doag.org

DOAG 2012 Business Intelligence München | 18. April 2012

Erfolgreicher durch BI!

Business Intelligence Technology
Big Data
Data Warehouse & ETL
Business Solutions & Management

Sponsoren



Kooperations-partner



DOAG
Deutsche ORACLE-Anwendergruppe e.V.




Mitarbeiter	Name	Abkürzung	Personalnummer	Eintrittsdatum	Lohn	Provision	Jahressumme	Abfertigung	Leistung
1000	BLAKE	MANAGER	7539	15.08.1981	8.000,00	-	36.000,00	30	
1001	KING	PRESIDENT	-	17.11.1981	9.000,00	-	40.000,00	10	
1002	BLAKE	MANAGER	7539	15.08.1981	8.000,00	-	36.000,00	30	
1003	CLARK	MANAGER	7539	09.09.1981	8.000,00	-	36.000,00	30	
1004	KING	MANAGER	7539	12.12.1981	8.000,00	-	36.000,00	30	
1005	SCOTT	ANALYST	7566	19.12.1982	8.000,00	-	36.000,00	30	
1006	FORD	ANALYST	7566	19.12.1981	8.000,00	-	36.000,00	30	
1007	JONES	MANAGER	7539	17.12.1980	8.000,00	-	36.000,00	30	
1008	WARD	SALESMAN	7549	12.12.1981	8.000,00	800,00	32.000,00	30	
1009	MARTIN	SALESMAN	7568	12.12.1981	8.000,00	1.400,00	32.000,00	30	
1010	TURNER	SALESMAN	7569	12.12.1981	8.000,00	-	32.000,00	30	
1011	ADAMS	CLERK	7567	12.12.1981	8.000,00	-	32.000,00	30	
1012	JAMES	CLERK	7550	12.12.1981	8.000,00	-	32.000,00	30	
1013	SLIVER	CLERK	7562	12.12.1981	8.000,00	-	32.000,00	30	

Abbildung 2: Die Darstellung in voller Breite zerstört zwar das Layout, man kann jedoch mehr Daten darstellen



EMPNO	ENAME	JOB	HIRE	COMM	DEPTNO
8000	BLAKE	MANAGER	7539	-	30
7829	KING	PRESIDENT	-	-	10
7539	BLAKE	MANAGER	7539	-	30

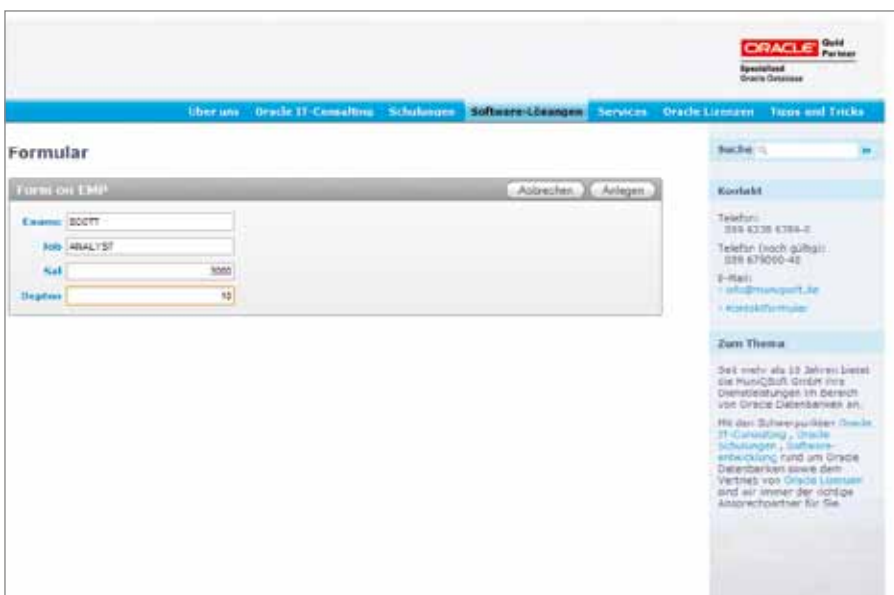
Einstellungsdatum: 01.05.1981 - Gehalt: 2.850,00 - Jahresgehalt: 34.200,00

EMPNO	ENAME	JOB	HIRE	COMM	DEPTNO
7782	CLARK	MANAGER	7539	-	30
7566	JONES	MANAGER	7539	-	30

Einstellungsdatum: 01.04.1981 - Gehalt: 2.975,00 - Jahresgehalt: 35.700,00

EMPNO	ENAME	JOB	HIRE	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	-	30
7902	FORD	ANALYST	7566	-	30
7369	SMITH	CLERK	7562	-	30
7499	ALLEN	SALESMAN	7549	300,00	30
7611	WARD	SALESMAN	7549	800,00	30
7654	MARTIN	SALESMAN	7568	1.400,00	30
7644	TURNER	SALESMAN	7569	-	30
7876	ADAMS	CLERK	7567	-	30
7800	JAMES	CLERK	7550	-	30
7934	SLIVER	CLERK	7562	-	30

Abbildung 3: Über den Klick auf eine Zeile werden weitere Informationen dargestellt



Formular

Form on EMP

Abbrechen Anlegen

Name: SCOTT

Job: ANALYST

Sal: 8000

Deptno: 10

Abbildung 4: Das Formular füllt die Seite nicht aus und sieht daher deplatziert aus

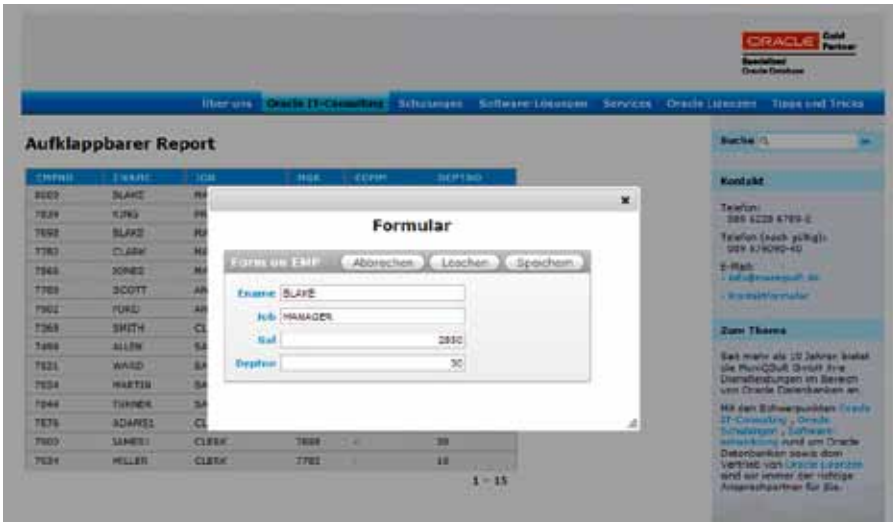


Abbildung 5: Das Formular wird über dem Report „schwebend“ dargestellt

Wir können dadurch jedoch mehrzeilige Ausgaben erreichen. Die Idee bietet sich vor allem in den Fällen an, in denen nicht alle Spalten notwendigerweise dargestellt werden müssen, um einen Datensatz zu identifizieren. Zusätzlich gibt es eventuell Spalten, die viel Text beinhalten. Also werden in der immer dargestellten Zeile die „kurzen“ Spalten ausgegeben, während der lange Text nur für den Datensatz angezeigt wird, der angeklickt wurde. Die unterhalb befindlichen Datensätze rutschen nach unten und machen Platz für den zusätzlich dargestellten Text.

Auch hier kommt jQuery zum Einsatz, das die Zeilen „anklickbar“ macht und für die Darstellung des zusätzlichen Inhalts sorgt (siehe Abbildung 3). Eine detaillierte Anleitung für die Erstellung einer solchen Lösung ist unter <http://www.muniqsoft.de/tipps/monatstipps/jahr-2011/februar-2011.htm> beschrieben.

Beispiel für ein kleines Formular

In unserer ausdesignten Apex-Anwendung kommen wir nun an den Punkt, an dem wir ein paar Daten über ein Formular eingeben möchten. Mit dem Apex-Wizard ist das Formular schnell erstellt, aber entweder benutzen die Formularfelder und die dazugehörigen Labels nur einen Teil des voll aufgezo-genen Fensters oder es klebt alles links oben im Content-Bereich. Der Region eine feste Breite zu verpassen, hilft hier zwar erst einmal, sieht allerdings trotzdem noch nicht gut aus (siehe Abbildung 4).

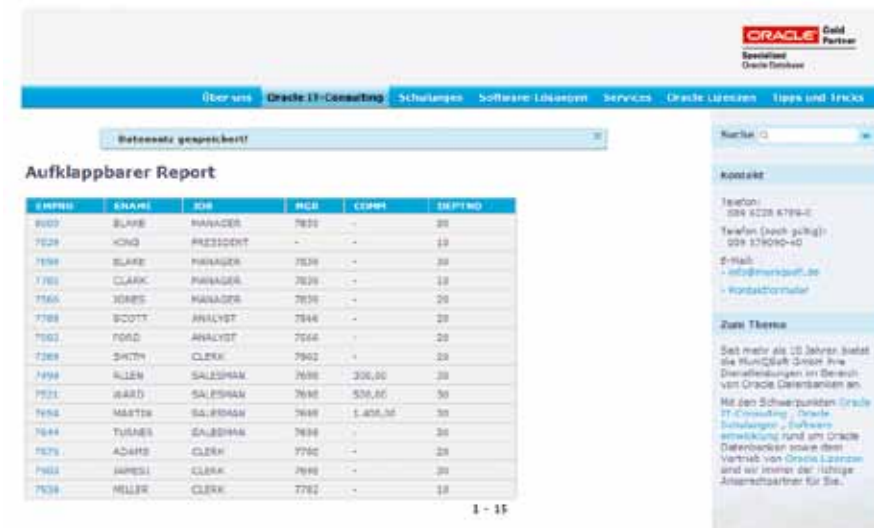


Abbildung 6: Der Report rutscht aufgrund der Message ein paar Pixel nach unten

Eine Lösung könnte sein, das Formular nicht auf einer eigenen Seite, sondern gewissermaßen über dem Report darzustellen. Erstellt man dieses Overlay mit der Ausprägung „modal“, lässt sich der darunter grau dargestellte Report erst wieder bearbeiten, wenn das Formular geschlossen wurde. Eine feste Breite und eine zentral ausgerich-tete Position lassen das kleine Formu-lar nun gut aussehen.

Aus technischer Sicht betrachtet, handelt es sich dabei eigentlich um eine Seite, die mit dem Formular-Wizard erstellt wurde. Zusätzlich wird auf der Seite der Report eingehängt. Mit

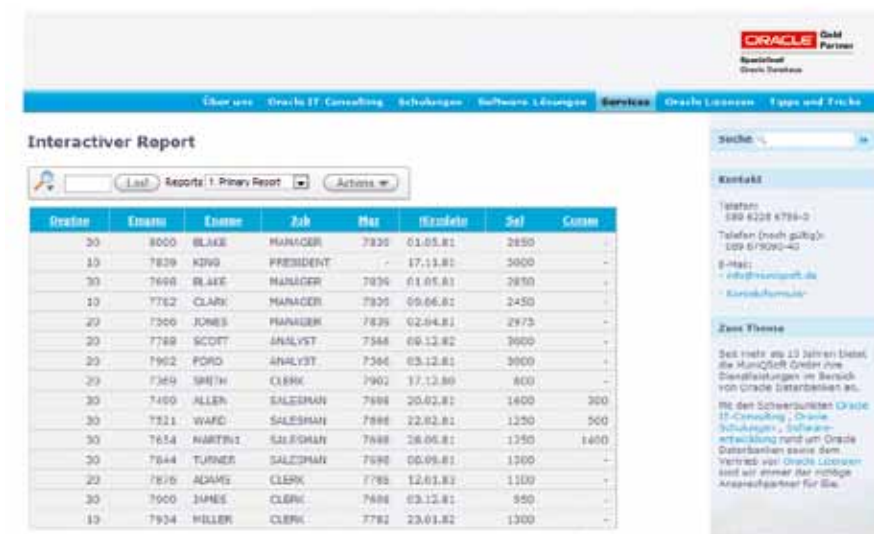


Abbildung 7: Eine kleinere Textbox und veränderte Button-Beschriftung in der Suchmaske des interaktiven Reports

ein paar „Conditions“ und dem Einsatz von jQuery wird nun dafür gesorgt, dass das Formular nur auf Klick im Report dargestellt und beim Speichern oder Abbrechen wieder ausgeblendet wird. Mit dem Einsatz der AJAX-Möglichkeiten lässt sich die Funktionalität sogar ohne eigentlichen „Submit“ der Seite durchführen. Das ist jedoch nicht in allen Fällen zwingend notwendig.

Mit jQuery wird die eindeutig benannte Formular-Region (Stichwort Static ID) mittels der Dialog-Funktion als manipulierbares Overlay dargestellt. Position, Größe und das Aussehen lassen sich dabei über Optionen steuern (siehe Abbildung 5).

Beispiel für die Message Areas

Nach so viel Speichern von Formularen stören die standardmäßig eingeblendeten Message Areas. Der gesamte restliche Content rutscht nach unten und verschiebt alles. Die Meldungen über erfolgreiches oder nicht erfolgreiches Speichern von Datensätzen

sind ja eigentlich ganz nützlich, aber dass die Oberfläche dabei immer verrutscht, ist nicht nur ein designtechnisches Problem, sondern auch eine Einschränkung der Ergonomie, da das Auge auf der Suche nach der Position derselben Formular-Elemente immer wandern muss.

Auch hier können jQuery und die Dialog-Funktion helfen. Zum einen wird die Nachricht über dem Report zentriert im oberen Bereich dargestellt, zum anderen verschwindet die Message-Box entweder automatisch nach ein paar Sekunden oder bei einem Klick außerhalb der Nachricht (siehe Abbildung 6 und Listing 2).

Damit der Dialog nur dargestellt wird, wenn gespeichert wurde, ist die Request-Variable beim Speichern belegt, was dann in JavaScript abgefragt werden kann. Die „Fade Out“-Funktion blendet innerhalb von drei Sekunden den Dialog sanft aus und schließt ihn danach.

Beispiel: Interaktiver Report ist Blackbox

Bei der Benutzung des interaktiven Reports merkt man als Entwickler sehr schnell, dass hier nicht alle Dinge verändert werden können, wie das bei einem normalen Report der Fall ist. Das liegt sicherlich daran, dass durch die

```
if (,&REQUEST.' == ,saved') {
    $('#messages').dialog({ position: 'top' });
    $('#messages').fadeOut(3000, function () {
        $('#messages').dialog('close')
    });
};
});
```

Listing 2



IT-Consulting

- › Performance Tuning
 - Oracle Datenbank Tuning
 - Oracle SQL + PL/SQL Tuning
- › Real Application Clusters
- › Data Guard + Fail Safe
- › Datenbank Management
 - Konfiguration
 - Backup & Recovery
 - Migration und Upgrade
- › OEM Grid Control
- › Oracle Security
- › Services
 - Remote DBA Services
 - Telefon-/Remotesupport

Nutzen Sie unsere Kompetenz für Ihre Oracle Datenbanken.

Schulungen

- › Oracle SQL
- › Oracle PL/SQL
- › Oracle DBA
- › Oracle APEX
- › Backup & Recovery
- › RMAN
- › Neuerungen 10g/11g
- › Datenbank Tuning
- › Datenbank Monitoring
- › Datenbank Security

Wir bieten Ihnen öffentliche Kurse sowie Inhouse-Schulungen.

Software-Lösungen

- › Individualsoftware
 - .NET und Visual Basic
 - Java
- › Oracle APEX
- › PL/SQL

Unser Ziel: Individuelle Softwareentwicklung mit Fokus auf Ihre Zufriedenheit.

Oracle Lizenzen

- › Oracle Datenbanken
 - Standard Edition One
 - Standard Edition
 - Enterprise Edition
 - Personal Edition

- › Oracle Produkte
 - Enterprise Manager
 - Oracle Tools

Optimale Lizenzierung durch individuelle Beratung.




```
$(,button.apexir-go-button > span').text(,Los!');
$(, #apexir_SEARCH').css(,width', ,50px');
```

Listing 3

Interaktivität viele Möglichkeiten festgesetzt sein müssen, um das Ganze nicht zu komplex zu machen.

Konkret bedeutet es zum Beispiel, dass wir keine Möglichkeit haben, das Template des Reports zu ändern. Die Änderung des Suchfelds oder die Beschriftung des Buttons ist nicht deklarativ möglich. Man könnte jetzt auf die Idee kommen, die darunter liegenden Quelldateien zu verändern. Schließlich besteht auch der interaktive Report nur aus HTML, CSS und ein bisschen JavaScript. Aber Änderungen an diesen

Stellen wirken sich in der Regel global aus, nicht nur in der Applikation, auch im Workspace oder sogar serverweit.

Auch hier bietet sich wieder einmal der Einsatz von jQuery an. Wir nutzen die Möglichkeit, Änderungen an der Seite durchzuführen, nachdem die Seite eigentlich fertig gerendert wurde, aber kurz bevor sie dem Anwender dargestellt wird. Über einen geeigneten Selektor zeigen wir auf das zu ändernde Element und verändern dann über jQuery-Funktionen die CSS-Eigenschaften oder die Texte. Aber Vor-

sicht, auch der interaktive Report selbst nutzt diese Technik und ändert vielleicht ebenfalls dynamisch nachträglich den Inhalt der Webseite. Und wer zuletzt kommt, gewinnt (siehe Abbildung 7 und Listing 3).

Über den Selektor „button.apexir-go-button > span“ finden wir einen Button, der die Klasse „apexir-go-button“ hat und der als Inhalt einen „“ besitzt. Den Text des Span-Elements ersetzen wir mit „Los!“.

Zusätzlich verpassen wir der Textbox, die mit der ID \$apexir_SEARCH\$ versehen ist, eine weitere CSS-Eigenschaft mit der Breite von 50 Pixeln.

Fazit

Ob eine Webseite ein gutes Layout hat oder nicht, ist oft Geschmackssache. Benutzerfreundlichkeit und Übersichtlichkeit sind eher messbar. Aber nur die harmonische Zusammenarbeit von Design und Funktionalität lässt eine inhaltlich gute Applikation auch zu einer akzeptierten Anwendung werden. Manchmal gilt das im Apex-Umfeld geflügelte Wort: „Akzeptanz durch Firlefanz!“

Markus Dötsch
MuniQSoft
markus.doetsch@muniqsoft.de



Kleines Beispiel jQuery

Die JavaScript-Bibliothek jQuery ist ein Open-Source-Framework, mit dem sich viele JavaScript-Funktionen auf einfache Weise nutzen lassen. In Apex ist jQuery seit der Version 4.0 in den Themes automatisch eingebunden. Grundsätzlich wird zuerst der Bereich auf der Webseite ausgewählt, der manipuliert werden soll, der sogenannte „Selektor“. Mit Punkt getrennt wird die Funktion angegeben, die auf das selektierte Element angewendet werden soll, zum Beispiel „\$(, #schalter').hide();“ (siehe Tabelle 1).

\$	Der Funktionsname von jQuery, alternativ kann auch „jQuery“ angegeben werden
#schalter	Bezeichnung des zu manipulierenden HTML-Elements „(<div id=“schalter“></div>“. Dabei steht die Raute (#) für IDs und der Punkt für Klassen (class). Es gibt noch viele weitere Möglichkeiten der Identifizierung von Elementen.
hide()	Funktion, um Element zu verstecken. In der Klammer können weitere Parameter angegeben werden.

Tabelle 1

Dazu ein weiteres Beispiel: „\$(, .content').css(,border', ,solid 3px red');“. Der Klasse „content“ wird dabei eine zusätzliche CSS-Eigenschaft mitgegeben, ein roter Rahmen mit drei Pixeln Breite. Bei den Regeln für die Syntax handelt es sich um die CSS-Schreibweise, mit dem einzigen Unterschied, dass Parametername und Wert kommasetrennt angegeben werden. Das Äquivalent zur oberen Zeile in einer CSS-Datei könnte so aussehen: „.content { border: solid 3px red; }“. Weiterführende Informationen und Tutorials für den Einsatz von jQuery sind unter <http://www.jquery.com> zu finden.

Newsticker

MySQL Cluster 7.2 ab sofort verfügbar

Die neue Version bietet laut Oracle bis zu sieben Mal mehr Leistung für komplexe Abfragen. Mit SQL- und NoSQL-Zugriff über eine neue Memcached API soll MySQL Cluster 7.2 das Beste aus „beiden Welten“ beinhalten: Es erlaubt Schlüssel/Wert-Operationen und komplexe SQL-Abfragen innerhalb derselben Datenbank.

Eine der Stärken von Apex – die einfache Anpassung des Layouts – ermöglicht schon heute das einfache Erstellen mobiler Anwendungen für Smartphones und Tablets. Dieser Artikel zeigt die Anforderungen mobiler Anwendungen und deren technische Umsetzung.

Apex für mobile Clients

Peter Raganitsch, click-click IT Solutions

Ein starker Trend der letzten Jahre – nicht zuletzt wegen des Erfolgs der iPhones und iPads – ist die immer stärker zunehmende Verbreitung von mobilen Endgeräten. Dies zieht sich vom „kleinsten“ Mitarbeiter bis hinauf in die Vorstandsetagen durch – nahezu jeder hat sein Smartphone oder Tablet. Auch Oracle ist sich der Tatsache bewusst geworden, dass in der heutigen Zeit mobile Clients immer mehr zunehmen, und wird – laut Statement of Direction (siehe Seite 14) – mit Apex Version 4.2 nativen Support für mobile Anwendungen liefern. Aufgrund der fast grenzenlosen Flexibilität von Apex ist es schon jetzt mit Version 3 und Version 4 möglich, Anwendungen zu erzeugen, die sich auch für mobile Endgeräte eignen beziehungsweise für diese optimiert sind.

Erwartungen an mobile Anwendungen

Dieser Abschnitt enthält nicht die umzusetzenden Punkte einer bestimmten Anwendung, sondern wir überlegen, was generell von einer mobilen Anwendung geleistet werden soll. Punkt eins ist die für die Bildschirmgröße optimierte Darstellung für die Bildschirmgröße: Der wohl größte Unterschied zwischen normalen Desktop-Anwendungen und mobilen Anwendungen ist die Größe des Bildschirms. Wird auf einem Smartphone-Browser eine normale Website geladen, dann sieht das aus wie in Abbildung 1. Eine optimierte Darstellung für mobile Endgeräte zeigt auf den ersten Blick die wesentlichen Informationen in der entsprechenden Größe, ohne dass dafür gezoomt werden muss (siehe Abbildung 2).

Hier ist also einerseits darauf zu achten, die Darstellung der Inhalte in der richtigen Größe auszugeben. Andererseits ist genau zu überlegen, welche Information wirklich übermittelt werden soll. Weniger ist an dieser Stelle sicher mehr. Durch das Weglassen unnötiger Information kann der Blick auf das Wesentliche gelenkt werden.

Ein weiterer Punkt ist die einfache Eingabe. Da man auf einem Smartphone nur über eine virtuelle Tastatur die Eingabe durchführen kann, ist es umso wichtiger, dem Benutzer die Eingabe so einfach als möglich zu machen. Mit der Definition der zu erwartenden Eingabe in einem Feld über den HTML-Input-Typ und einen entsprechenden Platzhalter gibt man dem Browser die Möglichkeit die Eingabe zu vereinfachen. Dies geschieht üblicherweise über eine



Abbildung 1: Darstellung einer normalen Webseite auf einem Smartphone-Browser



Abbildung 2: Darstellung einer optimierten Webseite



Abbildung 3: Eine Virtuelle Tastatur für Text

```

<!DOCTYPE html>
<html lang="&BROWSER_LANGUAGE." xmlns="http://www.w3.org/1999/xhtml" xmlns:html="http://html5.org/spec/html" xmlns:apex="http://apex.oracle.com">
<head>
  <title>#TITLE#</title>
  <link rel="icon" href="#IMAGE_PREFIX#favicon.ico" type="image/x-icon">
  <link rel="shortcut icon" href="#IMAGE_PREFIX#favicon.ico" type="image/x-icon">
  #HEAD#
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
  <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js"></script>
</head>
<body #ONLOAD#>
  <div id="Page&APP_PAGE_ID." data-role="page">
    #FORM_OPEN#

```

Listing 1

```

<div data-role="header" data-backbtn="false">
  <h1>#TITLE#</h1>
  #NAVIGATION_BAR#
</div><!-- /header -->
<div id="messages">#SUCCESS_MESSAGE##NOTIFICATION_
MESSAGE##GLOBAL_NOTIFICATION#</div>
<div data-role="content">
  #REGION_POSITION_01#
  #REGION_POSITION_02#
  #REGION_POSITION_03#
  #REGION_POSITION_04#
  #REGION_POSITION_05#
  #REGION_POSITION_06#
  #REGION_POSITION_07#
  #BOX_BODY#
</div><!-- /content -->

```

Listing 2

```

<div data-role="footer">
  <h4>#REGION_POSITION_08#</h4>
</div><!-- /footer -->

  #FORM_CLOSE#
</div>
</body>
</html>

```

Listing 3

spezielle virtuelle Tastatur für den jeweiligen Datentyp. So gibt es beispielsweise eigene Tastaturlayouts für Text, Zahlen, Telefonnummern, E-Mail-Adressen oder URLs (siehe Abbildung 3).

Ein besondere Augenmerk gilt der einfachen Auswahl. Aufgrund der begrenzten Bildschirmgröße ist es notwendig, Auswahlfelder, Links und

Schaltflächen so groß wie möglich zu gestalten, um deren Auswahl zu vereinfachen. Man denke nur an eine turbulente Zugfahrt, während der man versucht, auf einem Smartphone-Browser einen bestimmten Link auszuwählen. Mit einem großen Finger einen kleinen Link auf einem ebenso kleinen Bildschirm zu treffen, ist nicht einfach.

Apex wird mobil

Der Markt für Smartphones und Tablets ist riesig, das trifft nicht nur auf die Menge der Käufer zu, sondern auch auf die Anzahl der Hersteller, Modelle und Betriebssysteme. Jedes Gerät hat gewisse Vor- und Nachteile, vor allem aber unterschiedliche Features. Obwohl die meisten davon HTML5 und CSS3 schon umsetzen können, gibt es doch gewisse Feinheiten, die überall anders sind. Das beginnt bei Bildschirmgröße und Auflösung und endet mit der Interpretation von HTML5/CSS3-Standards.

Dass man sich da nicht selbst um die Feinheiten der Darstellung kümmern will, dürfte verständlich sein. Stattdessen bedienen wir uns zu diesem Zweck eines optimierten Frameworks für mobile Webseiten. Entschieden haben wir uns für „jQuery mobile“ (<http://jquerymobile.com/>), da dies perfekt zu dem bereits in Apex integrierten Basis-Framework jQuery passt, unter den verfügbaren mobilen Frameworks am reifsten ist und das breitetste Spektrum an Endgeräten abdeckt. Ganz nebenbei sei bemerkt, dass auch Oracle zu diesem Schluss gekommen ist und jQuery mobile als Basis für den mobilen nativen Support in Apex 4.2 verwenden wird.

Erstellen mobiler Templates in Apex

Der erste Schritt, um unsere Apex-Anwendung mobil zu machen, ist die Er-

stellung der entsprechenden Templates. Was wir genau in die Templates schreiben müssen, – wie die HTML-Struktur auszusehen hat – ist in der Dokumentation von jQuery mobile unter <http://jquerymobile.com/demos/1.0.1/docs/pages/page-anatomy.html> beschrieben. Wir werden daraus jetzt einen kleinen Teil umsetzen. Damit hat der Entwickler allerdings alle Informationen zur Verfügung, um weitere Templates nach der Vorgabe in der Dokumentation zu erzeugen. Dazu erstellen wir zunächst ein neues Page-Template in den Shared Components, nennen dieses „Mobile Page“ und fügen diese Code-Zeilen ein: Im Bereich „Header“ siehe Listing 1, im „Body“ siehe Listing 2 und im „Footer“ des Page-Template siehe Listing 3.

Ein Page-Template in Apex kann so kurz und übersichtlich aussehen. Damit haben wir den generellen Seitenaufbau definiert und jQuery mobile auf unserer Seite eingebunden. Für eine vollwertige Applikation fehlen noch Regions-, Listen- und Report-Templates, die wir sogleich anlegen. Listing 4 zeigt das Region-Template „Mobile Region“, Listing 5 das Listen-Template „Mobile Liste“ Before Rows und Listing 6 das „Current/Non-Current“-List-Element. Für „After Rows“ wird zum Schluss „“ eingefügt. Beim Anlegen des Report-Template „Formatted Text“ ist es wichtig, den Typ „Named Column (row template)“ zu wählen (siehe Listing 7).

Der Grund, aus dem wir für Reports dieses spezielle Template erstellen, hängt mit den beengten Platzverhältnissen auf Smartphones zusammen. Die von Desktop-Browsern gewohnte Report-Darstellung in tabellarischer Form scheidet aufgrund des Platzmangels aus. Stattdessen verwenden wir eine datensatzorientierte Darstellung, die im jQuery-mobile-Jargon „ListView“ heißt. Dabei wird nicht ein generisches Template für Spalten definiert, sondern ein Template für den gesamten Datensatz. Das gibt uns wiederum volle Kontrolle über das Layout. Einzige Bedingung für „Named Column“-Report-Templates ist, dass das Report-Select-Statement genau jene Spalten liefert, die laut den Tem-

```
<div id="#REGION_STATIC_ID#" #REGION_ATTRIBUTES#>
  <h2>#TITLE#</h2>
  <div data-role="controlgroup" data-type="horizontal">#CLOSE##PREVIOUS##NEXT##DELETE##EDIT##CHANGE##CREATE##CREATE2##EXPAND##COPY##HELP#</div>
  <div>
    #BODY#
  </div>
</div>
```

Listing 4

```
<ul data-role="listview" data-inset="true">
```

Listing 5

```
<li #IMAGE_ATTR#><a href="#LINK#" rel="external"
#A01#>#TEXT#</a></li>
```

Listing 6

```
<li>
  <a href="#LINK#" rel="external">
    <h3>#TITLE#</h3>
    <p><strong>#BOLD_TEXT#</strong></p>
    <p>#PLAIN_TEXT#</p>
  </a>
</li>
```

Listing 7

plate-Platzhaltern definiert wurden. Wer jetzt verwirrt ist, wird weiter unten beim Report-Beispiel hoffentlich Klarheit bekommen.

Mit der Erstellung dieser Basis-Templates haben wir die Möglichkeit, unsere mobile Apex-Anwendung mit genau den gleichen Mitteln zu erstellen, wie wir dies auch für eine normale Desktop-Apex-Anwendung tun würden.

Erstellen der ersten mobilen Apex-Anwendung

Zug um Zug entsteht nun die erste mobile Apex-Anwendung, die sich aus einer Startseite mit Hauptmenü, einer Report-Seite und einer Formular-Seite zusammensetzt. Für die Startseite (Page 1) mit dem Hauptmenü erstellen wir in den Shared Components eine Apex-Liste, die zumindest einen einzigen Ein-

trag „Customer“ enthält, der auf Page 10 verweist. Selbstverständlich verwenden wir für die erstellten Elemente (Seite, Region, Liste) die zuvor erstellten Templates. Abbildung 4 zeigt das Ergebnis der soeben erstellten Seite.

Weiter geht es mit Seite 10. Diese wird erstellt mit einem klassischen Apex-Report (nicht als interaktiver, sondern als normaler Bericht) unter Verwendung des zuvor erstellten Named-Column-Template. Das Select-Statement des Berichts basiert auf dem Datenmodell der Apex Sample Application und liefert genau jene Spalten, die im Report-Template erwartet werden (siehe Listing 8).

Will man an dieser Stelle noch mehr oder andere Spalten in den Bericht aufnehmen, muss das Template entsprechend erweitert werden. Die Namen der Platzhalter kann man dabei frei ver-


```

SELECT ,f?p=&APP_ID.:11:&APP_SESSION.::::P11_CUSTOMER_ID: '||CUSTOMER_ID AS LINK
      , cust_last_name || , , || cust_first_name AS TITLE
      , CUST_STREET_ADDRESS1 || decode(CUST_STREET_ADDRESS2, null, null, , , || CUST_STREET_ADDRESS2) AS
BOLD_TEXT
      , cust_city || ', , || cust_state || ', , || cust_postal_code AS PLAIN_TEXT
FROM DEMO_CUSTOMERS
ORDER BY CUST_LAST_NAME
      , CUST_FIRST_NAME

```

Listing 8

geben, die einzige Bedingung ist das Übereinstimmen von Spaltennamen und Platzhalter (siehe Abbildung 5).

Die erste Spalte des Berichts liefert eine URL zurück. Diese wird verwendet, um vom Übersichts-Bericht auf das Detail-Formular zur Bearbeitung des Datensatzes zu verzweigen. Entsprechend der Link-Definition im Bericht erstellen wir nun eine neue Seite 11 mit einem Formular auf Tabelle „DEMO_CUSTOMERS“. Da wir die nötigen Definitionen zum Aussehen der Seite bereits in den Templates erledigt haben, sehen wir sofort das Ergebnis (siehe Abbildung 6).

Beim Durchgehen der einzelnen Eingabefelder sehen wir, dass in allen Feldern dieselbe virtuelle Tastatur eingeblendet wird. Wollen wir spezielle Tastaturen (wie oben beschrieben) verwenden, müssen wir uns vorerst eines Plug-ins bedienen. Unter http://www.apex-plugin.com/oracle-apex-plugins/item-plugin/html5-input-item_107.html kann das freie Plug-in „HTML5 Input Item“ heruntergeladen werden. Dies unterscheidet sich vom regulären Text-Feld durch zwei zusätzliche Einstellungen. Einerseits kann der HTML5-Typ des Eingabefelds (Text, E-Mail, Telefon, URL etc.) gesetzt werden, andererseits auch ein Platzhalter. Mit dem HTML5-Typ wird die virtuelle Tastatur gesteuert, der Platzhalter zeigt einen Werte-Vorschlag an, falls das Eingabefeld noch leer ist. Diese beiden Einstellungen erleichtern dem Benutzer die Eingabe und zeigen, in welcher Form man den Eingabewert erwartet.

Navigation Bar

Bisher wurde noch nicht erklärt, wie die Buttons in die Kopfzeile kommen. Dort steht links neben dem Seitentitel

ein Button „Back“ und rechts neben dem Titel ein Button „Logout“. Beides sind typische Elemente einer Apex Navigation Bar und diese wollen wir dafür auch verwenden. Dazu fügen wir in unserem Page-Template im Bereich „Navigation Bar“ folgenden Code ein:

```

<a href="#LINK#"
rel="external" #ALT#>#TEXT#</
a>.

```

Bei der Definition der Navigation Bar Entries können wir nun im Feld „Icon Image Alt“ Zusatz-Attribute für unsere Buttons definieren. Beim Button „Back“ steht dort: „data-icon=“arrow-left““. Der Button „Logout“ wird ergänzt durch „data-icon=“delete“ class=“ui-btn-right““.

Durch Setzen dieser Zusatzattribute im HTML-Code steuern wir jQuery mobile. Mit „data-icon“ geben wir an, welches Icon bei dem Button angezeigt werden soll (eine vollständige Liste finden sie in der Dokumentation von jQuery mobile), mit dem Setzen der CSS-Klasse „ui-btn-right“ stellen wir sicher, dass der Button rechtsbündig angezeigt wird. Auch dies ist in der Dokumentation entsprechend beschrieben.

Im Grunde können wir mit dieser Vorgangsweise alle Features von jQuery mobile entsprechend in unserer Apex-Applikation abbilden. Einfach die in der Dokumentation beschriebene HTML-Struktur im Template abbilden, schon haben wir das gewünschte Ergebnis.



Abbildung 4: Die Startseite mit dem Hauptmenü



Abbildung 5: Apex-Bericht in mobiler Ansicht



Abbildung 6: Apex-Formular in mobiler Ansicht



Abbildung 7: Das Hauptmenü mit den Icons

Layout-Möglichkeiten

Durch den Einsatz von jQuery mobile haben wir natürlich weiterhin die Möglichkeit, das Aussehen unserer Applikation zu beeinflussen. So kann durch Definition eines sogenannten „Swatch“ eines der mitgelieferten Farb-Schemata von jQuery mobile aktiviert werden. Dazu einfach das „data-theme“ im HTML-Code hinzufügen und auf den Wert a,b oder c setzen. Damit können die 3 mitgelieferten Farbschemata aktiviert werden, Details dazu in der Dokumentation unter <http://jquerymobile.com/demos/1.0.1/docs/pages/pages-themes.html>. Aber auch abseits von jQuery mobile ist eine Gestaltung des Layouts

möglich. Allein durch Verwendung eigener Listen-Templates können sehr ansehnliche Darstellungen erzeugt werden (siehe Abbildung 7). Dazu verwenden wir eine normale Apex-Liste mit Icon-Definition und ein eigenes Listen-Template. Eine genauere Beschreibung davon sprengt an dieser Stelle allerdings den Rahmen des Artikels.

Aussicht auf Oracle Apex 4.2

Oracle selbst darf natürlich noch nicht verraten, was genau in dem kommenden Release enthalten sein wird und wann dieses erscheint (siehe Seite 14), deswegen an dieser Stelle ein paar Mutmaßungen des Autors. Oracle Apex

4.2 enthält native Unterstützung für die Erstellung mobiler Applikationen. Dies wird über jQuery mobile realisiert und in einer Mischung aus Plug-ins (für Items und für Listendarstellungen) und Templates zur Verfügung gestellt. Das Ziel an dieser Stelle wird sicherlich die einfache Anwendung mit größtmöglicher Flexibilität sein. Eine Demo-Anwendung und ein eigenes mobiles Theme werden an dieser Stelle ebenso erwartet. Im Bereich „Charts für mobile Anwendungen“ wird vermutlich die neueste Version von Any-Chart integriert, die anstelle von Flash mit HTML5 arbeitet und somit auf allen Plattformen funktioniert.

Fazit

Die kommende Version von Oracle Apex wird einen großen Schwerpunkt auf das Thema „Mobile“ legen. Allerdings müssen wir nicht darauf warten und können schon jetzt mobile Anwendungen erstellen. Da wir dazu dieselben Mittel verwenden, wie Apex das in Zukunft auch machen wird, ist diese Investition auch zukunftssicher.

Peter Raganitsch
click-click IT Solutions
peter.raganitsch@click-click.at



Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

Annegret Schlenker	Harald Unterschütz	Marion Erlebach	Ralf-Peter Schaum
Albin Hollenstein	Ralph Mösch	Wolfram Ditzer	Claus-Dieter Seidel
Sascha Schmorde	Hans Frötsch	Detlef Pollkläsner	Piotr Giemza
Jerome Witt	Denis Heinzmann	Michael Podewils	Olaf Czekay
Gregory Steulet	Andreas Risch	Andreas Wismann	Ulrich Küsters
Marek Adar	Matthias Laudt	John Bintz	Volker Eckert
David Hüber	Fredi Schweizer	Sergey Semenov	Kurt Stadlmair

Firmenmitglieder

Jörg Biesewig, megatel I.-u. K.systeme
Martin Wunderli, Trivadis
Thilo Rottach, TeamBank
Jan Gorkow, SD&C
Thomas Eifert, RWTH Aachen

Tipps und Tricks aus Gerds Fundgrube

Heute: Toolbar, Menüs und Hotkeys

Gerd Volberg, OPITZ CONSULTING GmbH

Funktionen wie zum Beispiel Einfügen, Löschen oder Speichern werden dem Anwender in einer guten Applikation immer mehrfach zur Verfügung gestellt – als Hotkey, über das Menü und auch als Toolbar-Button (siehe Abbildung 1). Wichtig ist hierbei, dass der dahinterliegende Source Code nicht mehrfach geschrieben wird.



Abbildung 1: Applikation mit Menü und Toolbar

Alle Menü-Items sowie die Buttons der Toolbar sollten mit dem Befehl „Do_Key“ entsprechende Hotkeys auf Form-Ebene starten (siehe Abbildung 2).

Der zentrale Dreh- und Angelpunkt ist nun der KEY-Trigger. In diesem steht der Source Code, der vom Hotkey, vom Menü und vom Toolbar-Button

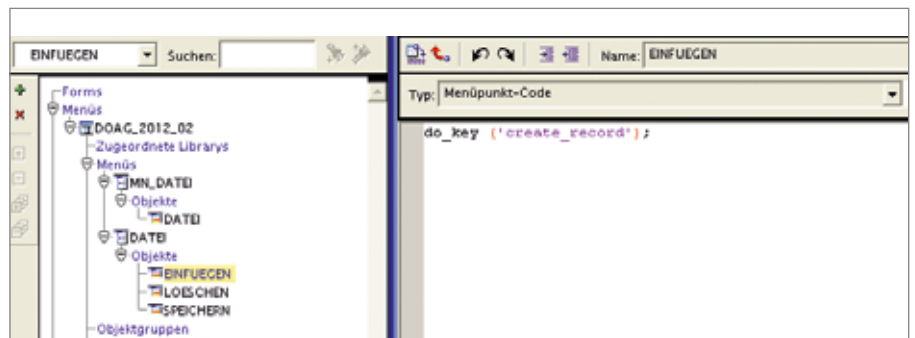


Abbildung 2: Einsatz der DO_KEY-Technik

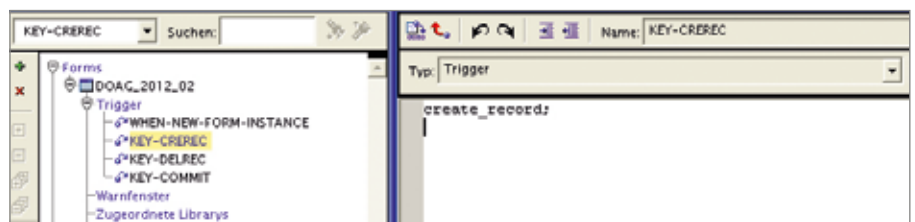


Abbildung 3: Zentraler Source Code für „Einfügen“

aus erreicht werden kann (siehe Abbildung 3). Sollte später einmal die Funktionalität des „Einfügens“ erweitert werden müssen, kann dies an zentraler Stelle im KEY-Trigger geschehen.

Gerd Volberg

gerd.volberg@opitz-consulting.com
talk2gerd.blogspot.com



Impressum

Herausgeber:
DOAG Deutsche ORACLE-Anwendergruppe e.V.
Tempelhofer Weg 64, 12347 Berlin
Tel.: 0700 11 36 24 38
www.doag.org

Verlag:
DOAG Dienstleistungen GmbH
Fried Saacke, Geschäftsführer
info@doag-dienstleistungen.de

Chefredakteur (ViSdP):
Wolfgang Taschner
redaktion@doag.org

Chefin von Dienst (CvD):
Carmen Al-Youssef
office@doag.org

Titel, Gestaltung und Satz:
Katja Borgis
DOAG Dienstleistungen GmbH

Titelfoto: Deutsche Bahn

Anzeigen:
CrossMarketteam Ralf Rutkat, Doris Budwill
www.crossmarketteam.de
Mediadaten und Preise finden Sie unter:
www.doag.org/go/mediadaten

Druck:
adame Advertising and Media
GmbH Berlin, www.adame.de



Dr. Dietmar Neugebauer,
Vorstandsvorsitzender der DOAG

Rekordbeteiligung bei der Beiratssitzung

Mehr als 60 DOAG-Aktive trafen sich Anfang Februar in der Nähe von Potsdam zur jährlichen Beiratssitzung. Der Beirat, bestehend aus den Repräsentanten der Querschnittsgruppen und den Community-Leitern, unterstützt die DOAG-Leitung beratend in allen Angelegenheiten des Vereins durch

sein Fachwissen und seine Position. Schwerpunkt der diesjährigen Beiratssitzung war die Arbeit in den Communities und deren Zusammenarbeit mit den Regionalgruppen.

Als Einstieg erhielten die Teilnehmer am ersten Abend eine Zusammenstellung der Zahlen und Fakten zu den durchgeführten Veranstaltungen und zur Mitgliederstruktur im Jahr 2011. Diese Unterlagen waren Voraussetzung für die Arbeit an den folgenden Tagen. Highlight des Donnerstagabends war der Auftritt des Gastreferenten Urs Meier, ehemaliger FIFA-Schiedsrichter und Management-Trainer, mit seinem Vortrag „Zwischen den Fronten – Entscheidungen unter Druck“. Er konnte anhand von Fußballszenen sehr gut vermitteln, wie wichtig es ist, notwendige Entscheidungen schnell zu fällen, auch wenn einem nicht alle Informationen immer zu 100 Prozent zur Verfügung stehen.

Mit dieser Motivation starteten am Freitagmorgen dann die vier Communities „Datenbanken“, „Development“, „Infrastruktur/Middleware“ und „Business Solutions“ in ihren Arbeitskreisen. Dort wurde sehr intensiv an den Themen Kommunikation, Events, DOAG Konferenz, Medien/Newsletter, DOAG-Online sowie Presse und Öffentlichkeitsarbeit diskutiert und die nächsten Maßnahmen dazu entschieden. Gleichzeitig beschäftigte sich eine fünfte Arbeitsgruppe mit der Thematik des Mitglieder-Profiles, den Aufgaben des regionalen Repräsentanten sowie der Zusammenarbeit mit den Communities. In allen Arbeitsgruppen waren auch Mitarbeiter von Oracle mit eingebunden, um eine enge Abstimmung der Inhalte auf beiden Seiten zu erzielen.

Am Samstagvormittag stellten die Arbeitsgruppen ihre Ergebnisse im Plenum vor und es wurde hier auch nochmal deutlich, dass die DOAG mit der Neustrukturierung auf dem richtigen Weg ist, um die Inhalte zu den Themen besser an die Mitglieder und Interessenten zu vermitteln. Anschließend wurde noch der aktuelle Stand des Projekts „Satzungsänderung“ präsentiert. Den Abschluss der Veranstaltung bildete schon traditionell der Besuch von

Günther Stürner, Vice President Server Technologies und Sales Consulting bei Oracle. Er gab einen Überblick zu den aktuellen Technologie-Schwerpunkten bei Oracle. Noch wichtiger war, dass er sich wie immer sehr offen den Fragen der Beiräte stellte und keine Frage unbeantwortet ließ.

Nochmals vielen Dank an alle, die an den drei Tagen mitgewirkt und so intensiv an den DOAG-Themen 2012 gearbeitet haben.



Christian Trieb
Leiter der Datenbank Community

Neues aus der Datenbank Community

„Select MAX(KnowHow) from DOAG group by Database;“ – das Community-Team hat im Rahmen der Beiratssitzung dieses Motto für die Datenbank Community entwickelt. Es verdeutlicht das zentrale Anliegen: Wissen und Erfahrungen über die Oracle-Datenbank einschließlich nahestehender Produkte an die DOAG-Mitglieder weiterzugeben und den Austausch untereinander zu fördern. Um dieses Anliegen umzusetzen, wurden folgende Möglichkeiten identifiziert:

- Online mithilfe des Internets
- Persönliche Treffen im Rahmen von Veranstaltungen der DOAG

Bei der Online-Präsenz spielen die Webseiten DOAG sowie die neuen Webinare die wichtigste Rolle. Auf www.doag.org finden Mitglieder sehr viele Präsentationen und Dokumente zu den unterschiedlichen Datenbank-Themen. Die Webinare, die jeden

DIE DOAG TRAUERT UM



Patrick Schwanke

† 5. Februar 2012

Patrick Schwanke war der DOAG als Referent auf der Konferenz und auf den deutschlandweiten Regionalveranstaltungen bis zuletzt stark verbunden. Die Teilnehmer waren angetan von seiner Fachkompetenz und seiner sympathischen Art, die Themen verständlich vorzutragen. In einer Vielzahl von Artikeln in den DOAG News hat er sein Fachwissen veröffentlicht und durch mehrere Buchpublikationen war er als Experte bekannt.

zweiten Freitag im Monat um 11 Uhr stattfinden, dienen dazu, ein bestimmtes Thema kurz den Mitgliedern zu vermitteln. Sie wurden bisher sehr gut angenommen.

Bei den Veranstaltungen spielen die Regionaltreffen, die SIGs und die jährliche DOAG Konferenz die größte Rolle. Die Datenbank Community achtet darauf, dass dabei die unterschiedlichen Datenbank-Themen angemessen repräsentiert sind, und entwickelt die Veranstaltungen nach den Wünschen der Mitglieder weiter. Hierzu werden Anregungen an christian.trieb@doag.org gern entgegengenommen. Für die DOAG 2012 Konferenz + Ausstellung wurden die Datenbank-Streams festgelegt und die Streams-Leiter sowie die Bewerber der eingereichten Vorträge benannt.

Im Rahmen der Beiratssitzung konnte die Datenbank Community zwei neue Mitglieder begrüßen: Matthias Mann, System Specialist bei der UniCredit, und Thomas Tretter, DOAG-Repräsentant in der Region Rhein/Main. Wer in der Datenbank Community mitarbeiten möchte, kann sich gerne beim Community-Leiter Christian Trieb (christian.trieb@doag.org) melden.



Stefan Kinnen
Leiter der Development Community

Die Development Community nimmt Fahrt auf

Die Definition des Namens „Development Community“ klingt einfach – sie war allerdings nicht selbstredend. Die Community-Leitung, im Wesentlichen bestehend aus den SIG-Leitern und den assoziierten Oracle-Ansprechpartnern,

nutzte jedoch das positive Klima der Beiratssitzung 2012, um sich einheitlich unter diesem Titel zu gruppieren.

Aktuell sind die beiden Fachkonferenzen DOAG 2012 BI (18. April 2012 in München) und DOAG 2012 Development (14. Juni in Bonn) die herausragenden Ereignisse.

Fachlich gibt es derzeit viele Themen. Die Diskussion mit Oracle über den Produktstatus von Oracle Reports läuft und wird gesondert publiziert. Dass Apex kein Nischen-Thema mehr ist, zeigen die Artikel dieser Ausgabe, aber auch das Expertenseminar mit Peter Raganitsch (18./19. April in Berlin). Die weiteren Highlights aus der Development Community (BI, Spatial, BPM, Java) stehen auf der Website der DOAG. Selbstverständlich freuen wir uns sehr über Ihren Kommentar zu den einzelnen Meldungen.



Christian Schwitalla
Leiter der SIG Development

Mobile Computing im Oracle-Umfeld

Mobile Computing ist in aller Munde. Die SIG Development vermittelte bei ihrer ersten Veranstaltung im Jahre 2012 einen guten Einblick über das Oracle-Umfeld. Aufgrund der zahlreichen Anmeldungen musste der ursprünglich geplante Tagungsort gewechselt werden, um allen 45 Teilnehmern reichlich Platz bieten zu können – ein wahres „Luxus-Problem“.

Stefan Kinnen, Leiter der DOAG Development Community, stellte zunächst die neu gegründete Development Community vor. Sie setzt sich aus den bisher bekannten SIGs zu-

sammen, die sich an die Zielgruppe der Software-Entwickler und -Designer richten. Diese SIGs werden ihre gewohnten Event-Reihen fortsetzen. Darüber hinaus wird es auch gemeinsame Veranstaltungen geben. Stefan Kinnen kündigte auch gleich die erste an: Die „DOAG 2012 Development – Erfahrungsaustausch für Software-Entwickler“ wird am 14. Juni 2012 in Bonn stattfinden.

Im ersten Vortrag „Überblick mobile Lösungen“ von Jörg Bredlau, Opitz Consulting, erhielten die Teilnehmer ein Wissensfundament rund um Mobile Computing. Eine wahre Flut an Informationen rollte auf die Teilnehmer zu – jedoch immer gut strukturiert, fundiert begründet und leicht verständlich dargestellt. Der Referent berichtete aus dem Berufsalltag über die stetig steigende Nachfrage nach Beratung zum Thema „Mobile Computing“. Nach dem Vortrag waren auch Mobile-Computing-Anfänger bereit für weiterführende Beiträge.

Jürgen Menge und Volker Linz, beide von Oracle, präsentierten in ihrem Vortrag „Oracle ADF Mobile“ eine noch nicht am Markt befindliche mobile Erweiterung des bekannten Oracle ADF Frameworks. In Kürze werden erste Versionen an Early-Adopter-Partner ausgeliefert. Mit ADF Mobile können in gewohnter Art und Weise ADF-Anwendungen entwickelt und auf mobile Plattformen verteilt werden. Im SDK-Umfang steht ein simuliertes mobiles Gerät zur Verfügung, um erste Tests zu fahren. Darüber hinaus sollten aber weitere Tests auf realen mobilen Geräten durchgeführt werden. Als großen Vorteil der ADF-Mobile-Entwicklungsplattform hob Jürgen Menge die Tatsache hervor, dass Entwickler in einer für sie gewohnten Umgebung (JDeveloper) jetzt auch mobile Anwendungen erstellen können, ohne neue Tools oder Programmiersprachen erlernen zu müssen. Im Anschluss fand eine Live-Demo von Volker Linz statt. Die Teilnehmer haben diese Lösung als sehr vielversprechend wahrgenommen.

Für viele Teilnehmer war die Tatsache unbekannt, dass sich auch mit Apex mobile Lösungen entwickeln lassen. Umso gespannter verfolgten sie

den Vortrag „Oracle Apex Mobile“ von Peter Raganitsch, click-click IT Solutions. Der Autor hat die weite Anreise aus Wien nicht gescheut, um den Teilnehmern Beispiele und Herausforderungen von mobilen Apex-Anwendungen zu präsentieren. Er ist in der DOAG Development Community als ein begeisterter Apex-Entwickler bekannt. Das war auch die Message seiner Präsentation – mit einigen Tipps und Tricks lassen sich heute schon mobile Anwendungen mit Apex erstellen – und das mit der bekannt hohen Produktivität. Das nächste Apex-Release wird voraussichtlich weitere Features für mobile Anwendungen liefern. Wir sind sehr gespannt darauf.

Zum Abschluss präsentierte Karin Patenge von Oracle in ihrem Vortrag „Unternehmensanwendungen mit lokaler Datenhaltung auf mobilen Endgeräten und deren Synchronisation mit zentralen Anwendungen: Berkeley DB, Mobile Server & Co. im Blick“ eine wichtige Seite vieler mobiler Anwendungen – das Daten-Management. Insbesondere zu diesem Vortrag hat sich eine lebhaft Diskussions entwickelt, da viele Teilnehmer zurzeit in ihren Projekten genau diese Aufgabe beschäftigt. In Zusammenhang mit dem Wegfall von Oracle Lite hat die Dozentin mögliche Migrationspfade zu Nachfolgeprodukten wie Berkeley DB beleuchtet. Dabei spielen eine SQL-API für die Berkeley DB und das Zusammenspiel mit dem Mobile Server eine zentrale Rolle. Last but not least thematisierte Karin Patenge das Pricing der neuen mobilen Produkte. Die Dozentin bot den Teilnehmern Unterstützung bei der Vorbereitung von Migrationsprojekten und eventuellen Problemen in diesen an.

Die DOAG bedankt sich für die zahlreiche und lebhaft Teilnahme bei allen. Die Vorträge zur Veranstaltung stehen unter <http://www.doag.org/termine/termine.php?tid=433877> zum Download bereit.

Mit der neuen Kommentar-Funktion können alle interessierten DOAG-Mitglieder an dieser Stelle ihre Erfahrungen und Informationen austauschen.

Weitere Veranstaltungen zum Thema „Mobile Computing“ sind in Planung. Die aktuellen Termine stehen

unter <http://www.doag.org/de/events/sigs/sig-developmenttools.html>.

Das nächste Treffen der SIG Development findet im Rahmen der „DOAG 2012 Development – Erfahrungsaustausch für Software-Entwickler“ am 14. Juni 2012 in Bonn statt. Themenvorschläge für weitere Treffen der SIG Development sind jederzeit willkommen.



*Dr. Frank Schönthaler
Leiter der Business Solutions Community*

Neuigkeiten aus der Business Solutions Community

Am 10. und 11. Februar 2012 fand am schönen Schwielowsee bei Potsdam die Jahressitzung des DOAG-Beirats statt. Ein wichtiges Ziel ist die Planung und Abstimmung aller DOAG-Aktivitäten. Für die Business Solutions Community stand am Freitag eine Leitungssitzung auf dem Programm. Bevor über die wichtigsten Ergebnisse berichtet wird, eines vorweg: Die Ergebnisse können sich sehen lassen. Sie spiegeln das hohe Engagement der Community-Aktiven wider, die neben den Interessen ihrer Community-Mitglieder stets auch übergeordnete strategische Überlegungen im Auge behalten haben:

- Kasi Färcher-Haag, Leiter der JD Edwards Community, wurde mit sofortiger Wirkung zum stellvertretenden Leiter der Business Solutions Community gewählt. Herzlichen Glückwunsch.
- Des Weiteren wurde die aus der bisherigen Agile PLM Community neu formierte Enterprise PLM Commu-

nity durch den Community-Leiter Charly Wachtel vorgestellt. In der Community finden sich die Anwender von Oracle PLM-Applikationen entlang der gesamten Produktwertschöpfungskette zusammen. Die Community hat auf der DOAG 2012 Applications viel vor: neben übergreifenden Themen werden auch produktspezifische Themen für e6- und A9-Anwender behandelt (siehe <http://plm.doag.org>).

- Die Vorbereitungen für die DOAG 2012 Applications (8. – 10. Mai 2012 in Berlin) sind in vollem Gange. Im Rahmen der Leitungssitzung wurde der Begutachtungsprozess auf Basis des Calls for Presentation abgeschlossen. Entstanden ist ein absolutes Top-Programm – prall gefüllt mit den neuesten Themen rund um das ganze Spektrum von Oracle Business-Applikationen. Über drei Tage hinweg werden Konferenzbeiträge sowie Praxis-Workshops angeboten. Um sich weiter zu informieren, gibt es parallel eine Ausstellung. Auf dieser können zusätzlich Informationen ausgetauscht und Networking betrieben werden.
- Neben der DOAG 2012 Applications wurde auch über weitere Veranstaltungen gesprochen. Die Partnerveranstaltung der Integrata Stiftung ist eng mit der DOAG 2012 Applications in Berlin verzahnt. Der Integrata-Kongress beginnt unmittelbar im Anschluss an die Applications, am 10. und 11. Mai 2012 (siehe <http://integrata-kongress.doag.org>).
- Auch die DOAG 2012 Logistik + SCM stand im Fokus der Beiratssitzung. Sie findet am 5. Juni 2012 in der Hamburger Speicherstadt statt. Weitere Informationen zu der Veranstaltung stehen unter <http://logistik-und-scm.doag.org>.
- Ein wichtiger Punkt auf der Tagesordnung war die Strategie für zukünftige BSC-Veranstaltungen. Generell zeigte sich, dass die einzelnen BSC Communities aktuell zu unterschiedlich sind, um zu einer einheitlichen Veranstaltungsstrategie zu kommen. Die BSC Communities planen ihre Veranstaltungen deshalb weiterhin eigenständig.



16.04.2012
Regionaltreffen München/Südbayern
 Franz Hüll, Andreas Ströbel
 regio-muenchen@doag.org

17.04.2012
Regionaltreffen Rhein-Main
 Thomas Tretter, Kathleen Hock
 regio-rhein-main@doag.org

18.04.2012
DOAG 2012 Business Intelligence
 Christian Weinberger
 sig-dwh@doag.org

18.04.2012
SIG Oracle Spatial
 Kushtrim Krasniqi, Paul Hurys
 sig-spatial@doag.org
 18.04.2012 – 19.04.2012

18.04.2012 – 19.04.2012
Berliner Expertenseminar
 Oracle APEX mit Plugins erweitern
 mit Peter Raganitsch
 Cornel Albert
 expertenseminare@doag.org

19.04.2012
Regionaltreffen Nürnberg/Franken
 Daniel Saraci, André Sept
 regio-franken@doag.org

24.04.2012
Regionaltreffen Freiburg
 Volker Deringer
 regio-freiburg@doag.org

24.04.2012
Regionaltreffen Hamburg/Nord
 Stefan Thielebein
 regio-nord@doag.org

25.04.2012
SIG Oracle & SAP
 Jörg Hildebrandt
 sig-sap@doag.org

24.04.2012 – 25.04.2012
Berliner Expertenseminare
 Oracle Security mit Pete Finnigan (Ersatztermin)
 Cornel Albert
 expertenseminare@doag.org



08.05.2012
Regionaltreffen Bremen
 Ralf Kölling
 regio-bremen@doag.org

08.05.2012-10.05.2012
DOAG 2012 Applications
 DOAG Geschäftsstelle
 office@doag.org

08.05.2012-09.05.2012
Community Day Agile A9
 Karl Wachtel
 bsc-agile@doag.org

08.05.2012
Regionaltreffen NRW
 Stefan Kinnen
 regio-nrw@doag.org

11.05.2012
Webinar: Cost based Optimizer
 mit Randolf Geist
 DOAG Geschäftsstelle
 office@doag.org

14.05.2012
Regionaltreffen Würzburg
 Oliver Pyka
 regio-wuerzburg@doag.org

15.05.2012
SIG Middleware: Plan & Build
 Andreas Risch, Jan-Peter Timmermann,
 Björn Bröhl
 sig-middleware@doag.org

15.05.2012
Regionaltreffen Hannover
 Andreas Ellerhoff
 regio-hannover@doag.org

21.05.2012
**Regionaltreffen Osnabrück/
 Bielefeld/Münster**
 Andreas Kother, Klaus Günther
 regio-osnabrueck@doag.org

22.05.2012
Regionaltreffen München/Südbayern
 Franz Hüll, Andreas Ströbel
 regio-muenchen@doag.org

23.05.2012 – 24.05.2012
Berliner Expertenseminar
 Oracle VM 3.0 mit Martin Bracher
 Cornel Albert
 expertenseminare@doag.org

23.05.2012
**Regionaltreffen Trier/Saarland/
 Luxemburg**
 Bernd Tuba, Holger Fuchs
 regio-trier@doag.org

24.05.2012
Regionaltreffen Stuttgart
 Jens-Uwe Petersen
 regio-stuttgart@doag.org

30.05.2012 – 31.05.2012
Berliner Expertenseminar
 Oracle Partitionierung mit Klaus Reimers
 Cornel Albert
 expertenseminare@doag.org

Vorschau auf die nächste Ausgabe

Die Ausgabe 03/2012 hat das
 Schwerpunktthema:

„Virtualisierung“

Sie erscheint am 1. Juni 2012.

Aktuelle Termine und
 weitere Informationen finden Sie unter
www.doag.org/termine/calendar.php