

Versteckte Juwelen: Nützliche APEX-Features - auch aus älteren Releases

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

Einleitung

Seit dem ersten Release der HTML DB zusammen mit Oracle10g in 2003 ist eine Menge passiert. HTML DB heißt seit einiger Zeit APEX; die Version 4.2 ist verfügbar und es gibt eine Fülle neuer Features und Möglichkeiten.

Wie bei jedem Softwareprodukt gibt es inzwischen auch in APEX Features, die zwar sehr nützlich sind, jedoch in Artikeln, Blogs oder Howto-Dokumenten kaum erwähnt werden. Der Vortrag auf der DOAG2012 schafft Abhilfe und stellt einige selten genutzte, aber sehr nützliche APEX-Features vor.

Formatmasken für Berichte

Eine numerische oder Datumsspalte in einem APEX-Bericht kann man formatieren - das ist bekannt. Dabei werden die aus SQL bekannten Formatmasken wie **9G999D99** oder **DD-MM-YYYY** verwendet. APEX bietet jedoch zusätzliche Formatmasken an, welche die Lesbarkeit eines Berichtes 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 APEX 4.0 verbessert, steht es doch als Schnellauswahl direkt unterhalb des Eingabefeldes für die Formatmaske bereit (Abbildung 1). Es ist geeignet für Zahlen zwischen 0 und 100. Durch Doppelpunkte getrennt, wird zunächst die Vordergrundfarbe, danach die Hintergrundfarbe und schließlich die Breite des Balkens in Pixel angegeben. Der Balken wird "von rechts" gefüllt - möchte man es "von links" her haben, sollte 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.

Column Definition ↑

Column Name: **PERCENT_MASK**
 Column Type: **NUMBER**
 Group: - Select Column Group -
 Display Type: **Standard Report Column**
 * Column Heading: Percent Mask Use Same Text for Single Row View
 * Single Row View Label: Percent Mask
 Number / Date Format: **PCT_GRAPH.ff7777:ccccc:150**
Numeric format mask: 999G999G999G999G990
 BLOB Download Format Mask
 Graphical formatting for percentages, whole numbers between 0 and 100
 Heading Alignment: center Column Alignment: right
 Allow Users To:
 Hide Sort Filter Highlight Control Break Aggregate Compute Chart Group By

Abbildung 1: Formatmaske PCT_GRAPH

Die Formatmasken 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 - es 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 (Abbildung 2).

Formatmasken

Percent	Percent Mask	Datum	Datum Mask	Filesize	Filesize Mask
15	<div style="width: 15%; height: 10px; background-color: #ff4500; border: 1px solid #ccc;"></div>	27.01.81	vor 30,8 Jahren	1762376761762	2TB
85	<div style="width: 85%; height: 10px; background-color: #ff4500; border: 1px solid #ccc;"></div>	06.11.11	vor 2 Wochen	176237	172KB

Abbildung 2: Formatmasken in Aktion

Die Vorteile der eingebauten Formatmasken 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 Formatmasken 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**

Schnellauswahlen nutzen

Schnellauswahlen sind Links unterhalb von Eingabefeldern in Formularen - bei Klick darauf wird das Eingabefeld automatisch gefüllt. Im Application Builder werden die Schnellauswahlen sehr intensiv genutzt, zum Beispiel bei den Formatmasken (siehe auch Abbildung 1). APEX erlaubt Schnellauswahlen auch in eigenen Anwendungen, und zwar für alle Formularelemente. Navigiert man zu den Einstellungen eines Elementes, so gibt es den Abschnitt **Schnellauswahl (Quick Picks)**; Abbildung 3 zeigt das anhand eines Beispiels:

Label	Value
1 [CLERK]	1 CLERK
2 [SALESMAN]	2 SALESMAN
3 [ANALYST]	3 ANALYST
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Abbildung 3: Einrichten einer Schnellauswahl

Das Formular ist sofort wesentlich leichter zu bedienen.

* Empno

Ename

Job
[CLERK] [SALESMAN] [ANALYST]

Mgr

Hiredate

Sal

Comm

Deptno

Abbildung 4: Schnellauswahlen in Aktion

APEX-Komponenten-Management

Viele der weniger bekannten Features beschäftigen sich mit dem Management von APEX-Komponenten. Und Features, die nicht direkt zur Benutzeroberfläche einer Anwendung beitragen, gehören nahezu immer zu den "kaum bekannten" Funktionen.

Erstelloptionen

Erstelloptionen (Build Options) erlauben es, mehrere Komponenten einer APEX-Anwendung auf einmal zu aktivieren oder zu deaktivieren - ohne dass dazu eigene Bedingungen definiert werden müssen. Dazu muss zuerst eine neue Erstelloption erzeugt werden - das geschieht in den Gemeinsamen Komponenten und dort bei Erstelloptionen. Das Beispiel in Abbildung 5 erzeugt eine neue Erstelloption mit dem Status **Exclude**.



Abbildung 5: Eine neue Erstelloption mit Status EXCLUDE

Mit dieser Erstelloption können nun alle Anwendungskomponenten versehen werden (Abbildung 6). Solange der Status auf Exclude steht, werden sie zur Laufzeit nicht ausgeführt. Erstelloptionen werden auch beim Export berücksichtigt; es ist also möglich, eine Anwendung mit oder ohne Komponenten einer bestimmten Erstelloption zu exportieren.

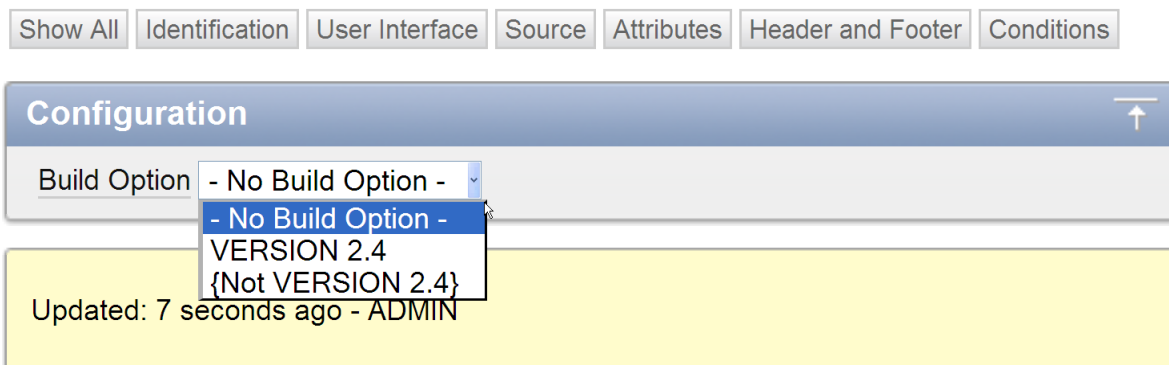


Abbildung 6: Erstelloption bei einer APEX-Komponente einstellen.

Wenn in die Anwendung neu eingebaute Funktionalität aus vielen Komponenten auf mehreren Seiten besteht, sind Erstelloptionen sehr zu empfehlen - das Aktivieren oder Deaktivieren wird mit einfachem Setzen der Erstelloption auf Include oder Exclude realisiert. Gleichzeitig werden Anwendungskomponenten dokumentiert - man sieht sofort, zu welchem Zweck und ggfs. zu welcher Zeit eine Komponente eingeführt wurde.

Kommandozeilen-Export und APEX_APPLICATION_INSTALL

Geht es darum, Anwendungskomponenten auf einen anderen Server zu übertragen oder in ein Versionskontrollsystem, wie CVS, Subversion oder andere, 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 Kommandozeilenwerkzeug mitliefert, welches nicht nur den Export einzelner Anwendungen, sondern auch den Export aller Anwendungen eines Workspace oder gar der APEX-Instanz erlaubt. Das Aufsetzen eines Jobs, der täglich alle Anwendungen exportiert und ins Versionskontrollsystem eincheckt, ist dann nur noch eine Kleinigkeit. Das Kommandozeilenwerkzeug muss allerdings erst eingerichtet werden - wie das geht, ist in einem Tipp der deutschsprachigen APEX-Community (<http://tinyurl.com/apxcmdexp>) beschrieben.

```
$ 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

Ein weiteres Werkzeug ist der *APEX Export Splitter*, welcher eine Exportdatei in ihre Komponenten zerlegt - es entsteht eine Verzeichnisstruktur, in der eine eigene Exportdatei für jede APEX-Komponente enthalten ist. (Abbildung 7). Auch dieser ist im Tipp der deutschsprachigen APEX-Community näher erklärt.

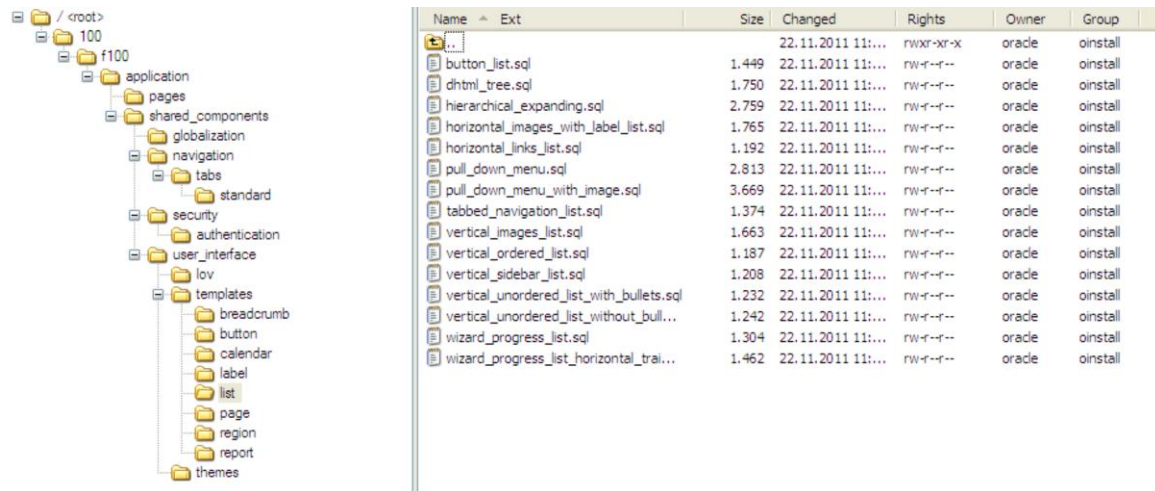


Abbildung 7: Der APEX Export Splitter zerlegt eine Exportdatei

Kombiniert man dies mit dem erwähnten täglichen Exportieren und Einchecken ins Versionsmanagement, 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** erwähnenswert. Wie die meisten Entwickler wissen, ist eine APEX-Exportdatei ein SQL-Skript, welches 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. Hat man nur eine einzelne Seite oder Komponente exportiert, so ist der Import in eine andere Anwendung noch nicht einmal über die APEX-Entwicklungsumgebung Oberfläche möglich.

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 APEX 4.0 oder höher erstellt wurde. Ist sie älter, so sollte sie in APEX 4.0 oder höher importiert und daraus wieder exportiert werden.

```

SQL> begin
  2 apex_application_install.set_workspace_id(3138423103500405);
  3 apex_application_install.set_application_id(999999);
  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*

Dies und das: Verschiedenes

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 aber 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 Inhaltes "Name: APEX" für das Element **P1_PRODUKT** per URL 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

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

Der URL-Parameter **p_trace** kann auf **YES** gestellt werden; APEX wird dann für das Seiten-Rendering eine SQL-Tracedatei erzeugen. Diese Datei wird in der *user_dump_destination* des Datenbankservers 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 Performanceproblemen eingesetzt.

Ab APEX 4.0 kann man die Spracheinstellung mit dem URL-Parameter **p_lang** einstellen - das ist wesentlich einfacher als das umständliche Einstellen der Language Preference im Browser. Mit

p_lang können verschiedene URLs zur Anwendung direkt so konstruiert werden, dass fremdsprachige Nutzer gleich in der richtigen Sprache landen.

Wertelisten

Generell sollten Wertelisten, wo möglich, dynamisch definiert werden, denn statische Wertelisten existieren nur in APEX und können nur im Application Builder angepasst werden. Ein Fremdschlüssel, welcher 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 noch mehr Anredeformen.

Bei der Definition einer statischen Werteliste werden die einzelnen Einträge durch Komma und innerhalb eines Eintrags werden Anzeige- und Rückgabewert durch ein Semikolon getrennt. Verwendet man das Schlüsselwort **STATIC**, so sortiert APEX die Einträge alphabetisch, bei **STATIC2** werden die Einträge in der definierten Reihenfolge dargestellt.

Die einzelnen Trennzeichen können aber selbst definiert werden: Sollen Komma oder Semikolon in den Einträgen verwendet werden, 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%N
```

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

Listing 4: Syntax für statische Wertelisten

Der Application Builder

Mit APEX 4.0 wurde eine wertvolle Navigationshilfe in den Application Builder eingeführt. Oben rechts befindet sich das Eingabefeld für die "Powersuche". 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 die APEX-Syntax *[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.

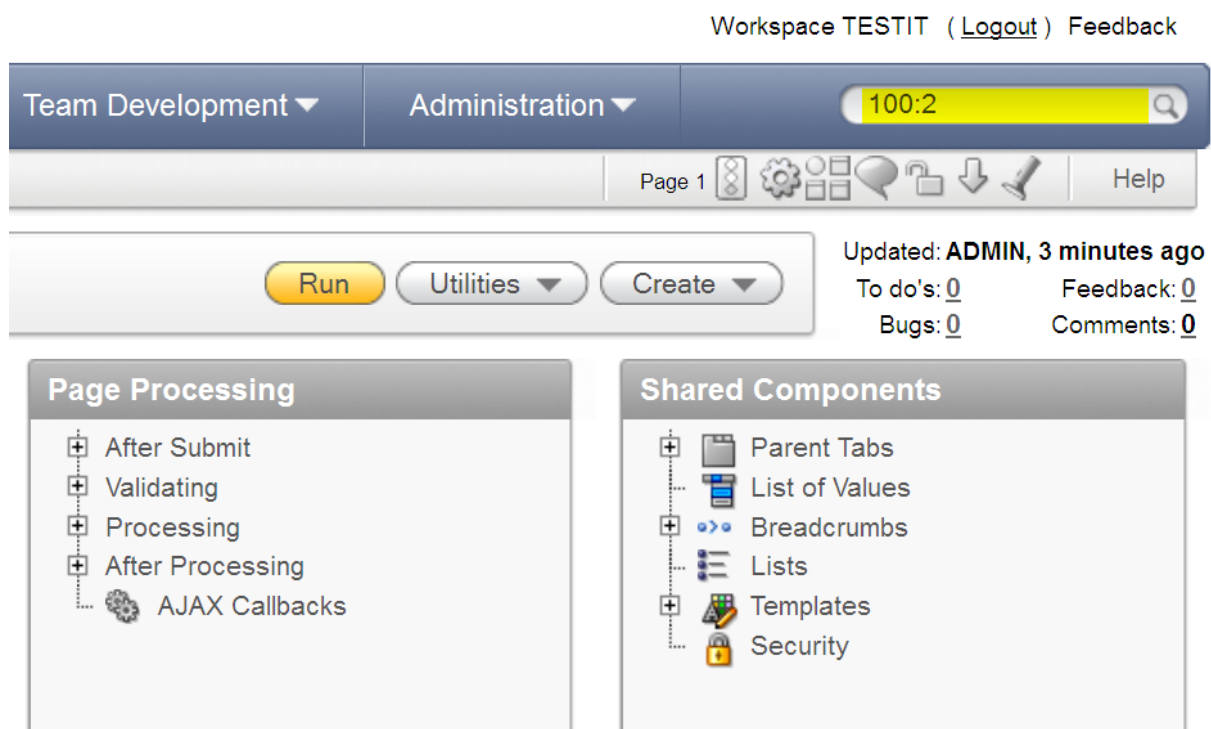


Abbildung 8: APEX Powersuche

Vielfach steht man in der Situation, dass man das Aussehen eines Template oder einer Region ändern möchte - und da ist es üblich, dass man sich die gemachten Änderungen immer wieder ansieht. Typischerweise hat man die Anwendung hierfür in einem zweiten Browserfenster geöffnet.

Das in Abbildung 9 dargestellte **Return to Page** ist hier extrem nützlich - ist es angeklickt, so 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.

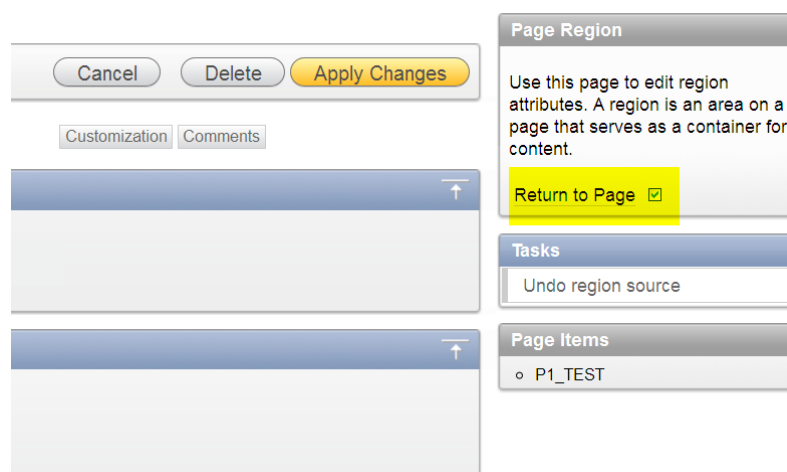


Abbildung 9: Return to Page: Beim Speichern der Änderungen landet man wieder auf der gleichen Seite

Der Blick in die Vergangenheit.

APEX nutzt Flashback Query sehr intensiv. So kann man eine Anwendung so exportieren, wie sie vor bspw. fünf Minuten aussah (Abbildung 10). Wie weit man in die Vergangenheit zurückgehen kann, hängt von der Konfiguration der Datenbank (hier: des UNDO-Tablespace) ab, Zeiträume bis ca. 10 Minuten sollten aber auf nahezu allen Systemen möglich sein. Und mit einem solchen Export lassen sich versehentlich gelöschte Komponenten einfach wiederherstellen - Voraussetzung ist allerdings, dass man den Fehler schnell genug bemerkt.

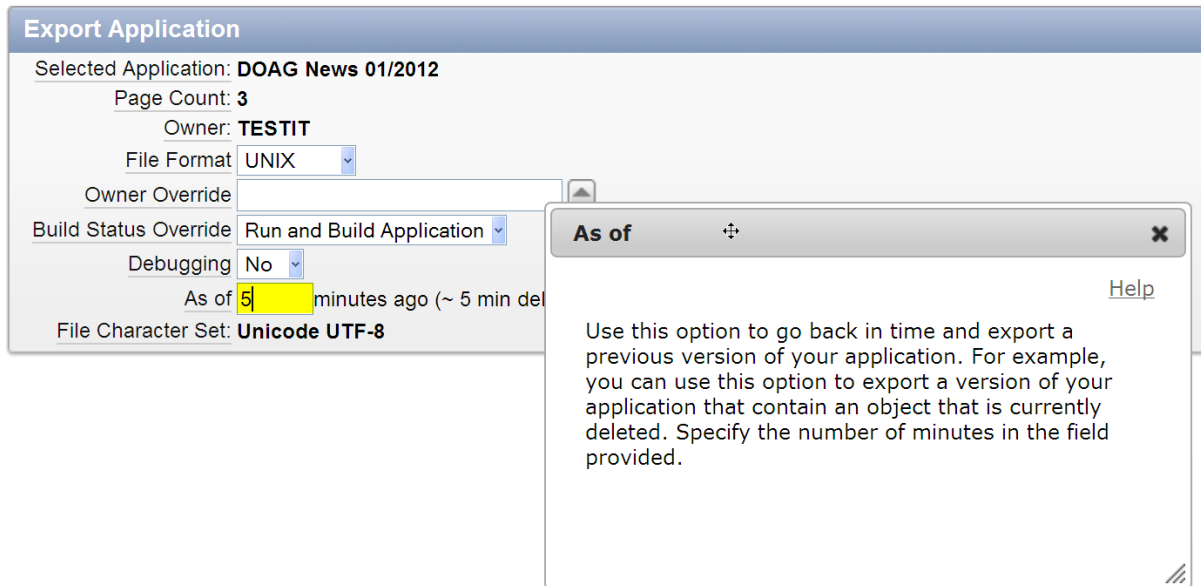


Abbildung 10: Exportieren "aus der Vergangenheit"

Undo-Funktionen mit Hilfe von Flashback sind auch in den Dialogen zur Regionbearbeitung 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 zeigt sich der in Abbildung 11 dargestellte Dialog.

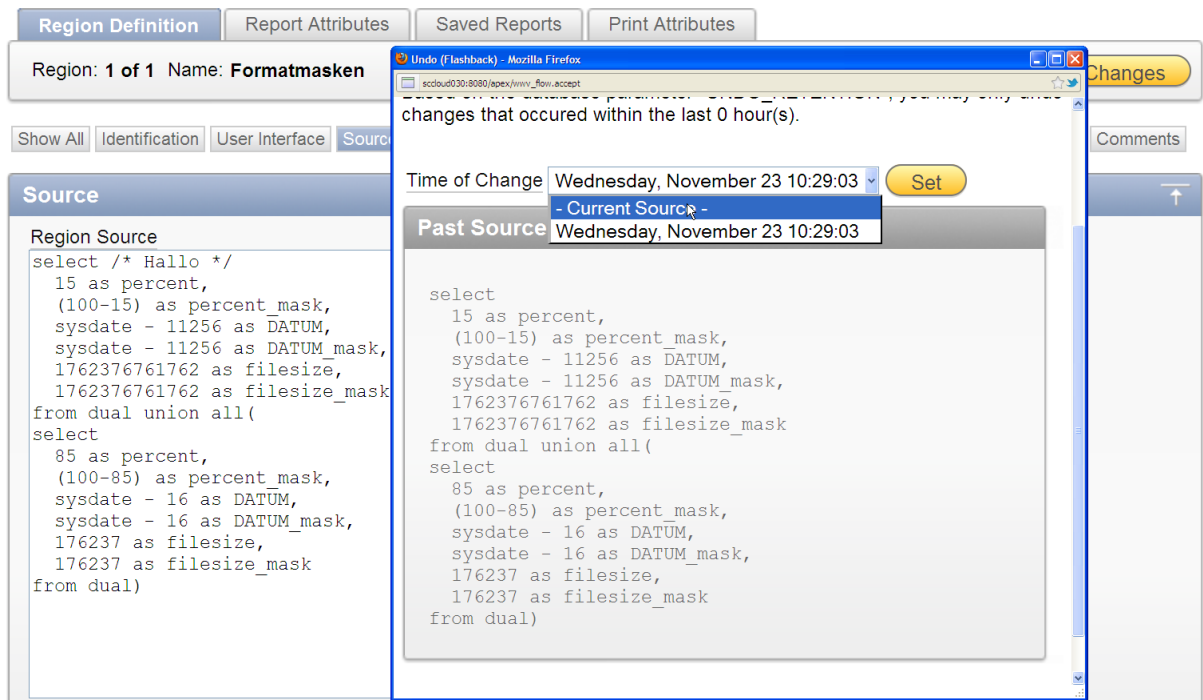


Abbildung 11: Änderungen an der Regionsquelle rückgängig machen

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 Material auch in deutscher Sprache vorhanden:

[1] Deutschsprachige APEX Community

<http://tinyurl.com/apexcommunity>

[2] Oracle APEX im OTN

<http://otn.oracle.com/apex>

[3] Oracle Datenbank auf Twitter

<http://twitter.com/OracleBUDB>

Kontaktadresse:

Carsten Czarski
ORACLE Deutschland B.V. & Co KG
Riesstr. 25
80992 München

Telefon: +49 (0) 89 1430 2116
E-Mail carsten.czarski@oracle.com
Internet: www.oracle.de

Blog des Autors <http://sql-plsql-de.blogspot.com>
Twitter [@cczarski](https://twitter.com/cczarski)