

Etwas mehr als ein Jahr nach Erscheinen von Application Express (Apex) 4.0 mit seiner Fülle an neuen Features ist nun Apex 4.1 herausgekommen. Die Versionsnummer legt es schon nahe: Apex 4.1 vervollständigt im Wesentlichen die bereits in Apex 4.0 eingeführten Features und rundet diese ab – so gibt es erhebliche Verbesserungen bei tabellarischen Formularen. Aber es gibt auch das eine oder andere völlig neue Feature wie ein komplett überarbeitetes Error-Handling.



Neu: Application Express 4.1

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

Wie schon bei der Version 4.0 ging dem Erscheinen eine mehrmonatige, öffentliche „Early-Adopter-Phase“ auf dem Server „tryapexnow.com“ voraus.

Im Bereich der Formulare und tabellarischen Formulare hat das Apex-Entwicklerteam einige hilfreiche Verbesserungen eingebaut. So kann man nun anstelle des Primärschlüssels die Oracle-Row-Id als „eindeutige Spalte“ für ein Formular auf eine Tabelle festlegen (siehe Abbildung 1).

Das ist insbesondere hilfreich, wenn die zugrundeliegende Tabelle einen zusammengesetzten Primärschlüssel mit mehr als zwei Spalten hat: In der Vergangenheit musste man das Formular für solche Tabellen komplett manuell erstellen, da die Apex-Assistenten maximal zwei Primärschlüssel-Spalten zulassen. Apex 4.1 bietet nun auch für solche Tabellen sogenannte „Standardformulare“ an: Das Arbeiten wird nochmals schneller und produktiver.

Im Bereich der „tabellarischen Formulare“ enthält Apex 4.1 von Entwicklern lang erwartete Funktionen. So ist es endlich möglich, Validierungen zu hinterlegen, die beim Absenden des Formulars für jede Zeile einzeln geprüft werden. Abbildung 2 zeigt eine Syntax zum Ansprechen der Spalten des tabellarischen Formulars in PL/SQL-Code. Wie bei normalen Apex-Elementen wird mit der „Doppelpunktsyntax“ gearbeitet. „:SAL“ meint also die Spalte „SAL“ des

tabellarischen Formulars. Diese neue Syntax kann übrigens auch verwendet werden, wenn ein Entwickler die Verarbeitungsprozesse des tabellarischen

Formulars selbst in PL/SQL implementieren möchte. War er in der Vergangenheit gezwungen, die unhandlichen Arrays „Apex_APPLICATION.G_F01“



Abbildung 1: Die Row-Id ist nun eine eindeutige Spalte des Apex-Formulars

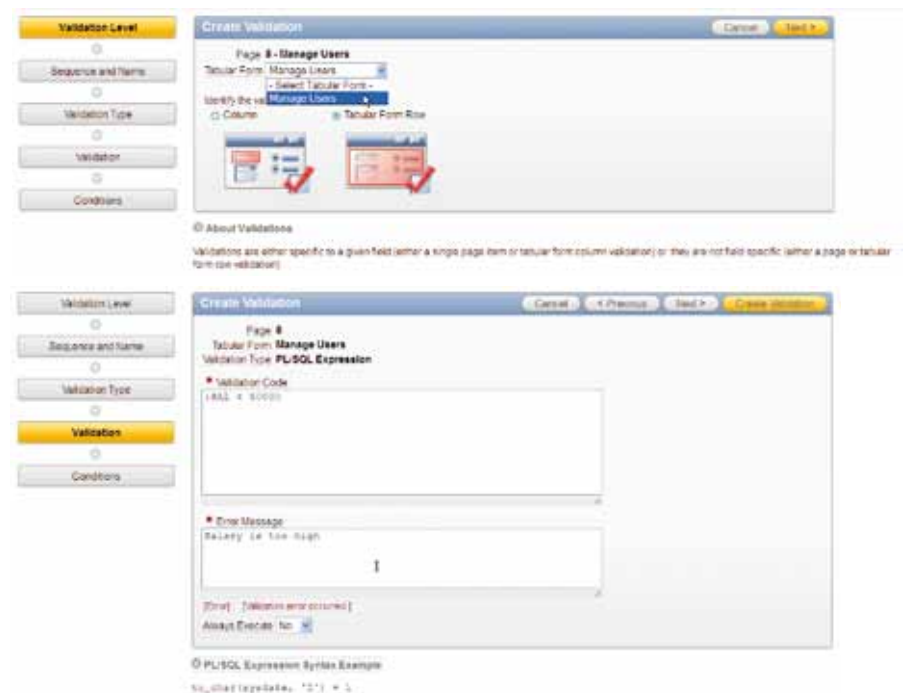


Abbildung 2: Validierung für jede einzelne Zeile eines tabellarischen Formulars

bis „Apex_APPLICATION.G_F50“ einzusetzen, so lässt sich die Formularspalte nun elegant per Doppelpunkt-Syntax referenzieren. Der Code eines solchen PL/SQL-Prozesses ist damit besser les- und wartbar. Man kann nun auch deklarativ festlegen, ob der Prozesscode nur einmal für das gesamte Formular oder einmal für jede Zeile ausgeführt werden soll. So reicht zum Einfügen aller Zeilen des Formulars in eine neue Tabelle „EMP1“ folgender PL/SQL-Code aus, wenn eingestellt wird, dass er pro Zeile ausgeführt werden soll (siehe Listing 1).

```

Begin
  insert into emp1 (
    empno, ename, mgr, job, sal,
    hiredate, comm, deptno
  ) values (
    :EMPNO, :ENAME, :MGR, :JOB,
    :SAL, :HIREDATE, :COMM, :DEPTNO
  );
end;

```

Listing 1: PL/SQL-Prozesscode zum Verarbeiten der Eingaben eines tabellarischen Formulars

Plug-ins

Plug-ins, in Apex 4.0 eingeführt, sind hervorragend geeignet, Funktionalitäten zu kapseln und wiederverwendbar zu gestalten. Ist ein Plug-in von breitem Interesse, so bietet es sich an, es auf Community-Webseiten wie „http://www.apex-plugin.com“ zu veröffentlichen. Aber nicht jedes Plug-in muss gleich für die Öffentlichkeit bestimmt sein – auch unternehmensintern ist es sehr sinnvoll, Funktionen als Plug-in zu kapseln und in einer Art Bibliothek vorzuhalten – spätestens bei der zweiten Verwendung rechnet sich der Mehraufwand für die Erstellung des Plug-ins.

Apex 4.1 führt zwei neue Plug-in-Typen ein: So können nun Plug-ins für Authentifizierungs- und Autorisierungsschemata erstellt werden. Insbesondere die Authentifizierungs-Plug-ins sind für Apex-Entwickler in größeren Unternehmen hochinteressant. Denn typischerweise müssen im Unternehmen alle Anwendungen das gleiche Login-Verfahren verwenden – es bietet sich also an, dieses einmalig

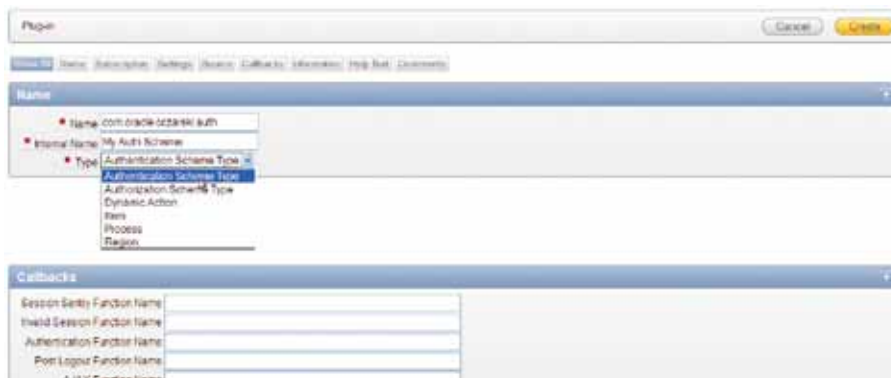


Abbildung 3: Apex 4.1 bietet unter anderem Plug-ins für Authentifizierungsschemata



Abbildung 4: Einstellung der Anzeigeposition für Fehlermeldungen

als Plug-in zu kapseln und danach wiederzuverwenden, denn das Einspielen eines Plug-ins ist bei weitem nicht so aufwändig und fehlerträchtig wie das Nacharbeiten einer Anleitung und das Copy & Paste von Code-Fragmenten. Darüber hinaus gibt es auch hier Verbesserungen im Detail: So kann ein Plug-in nun 15 anstatt 10 Parameter besitzen und es sind neue Parameter-Typen dazugekommen.

Neues Error-Handling in Apex-Anwendungen

Auf das verbesserte Error-Handling bei DML-Prozessen in Apex-Formularen haben Entwickler lange gewartet. Bislang leitete Apex bei einem Fehler im Prozess auf eine Fehlerseite um – das Layout dieser Seite lässt sich nur schwer beeinflussen. Apex 4.1 schafft Abhilfe und bietet für Prozesse ein neues Attribut „Error Message Display Location“ an (siehe Abbildung 4).

Allein dies vereinfacht den Umgang mit Fehlern schon massiv, denn gerade die separate Fehlerseite war in den meisten Anwendungen ein Problem.

Doch damit nicht genug: Apex 4.1 führt zusätzlich die Möglichkeit ein, eine eigene Error-Handling-Funktion zu hinterlegen. Diese muss, wie immer bei einer Schnittstelle, eine bestimmte Parameter-Signatur aufweisen. Sie wird von Apex aufgerufen, wann immer ein Fehler auftritt. Die Implementierung der Funktion selbst ist frei. Denkbar sind also zusätzliches Logging, automatische Nachrichten an den Entwickler oder aber auch das Ändern der dargestellten Fehlermeldung.

Das Einrichten der eigenen Error-Handling-Funktion ist einfach: Zunächst erstellt man die PL/SQL-Funktion selbst. Der folgende Code stellt beispielsweise fest, welcher SQL-Fehler vorliegt. Wenn es ein „ORA-014382“ ist, dann wird die Standard-Fehlermeldung durch einen eigenen Text ersetzt (siehe Listing 2). Diese Funktion muss nur noch in den Seiten-Attributen als „Error Handling Function“ eingetragen sein und schon ist man fertig (siehe Abbildung 5). Die Anwendungsseite zeigt nun im Fehlerfall den vom Entwickler festgelegten Text an.



Abbildung 5: Die selbstgeschriebene Error-Handling-Funktion wird für die Apex-Seite hinterlegt

```

create or replace function my_
error_handler (
  p_error in apex_error.t_error
) return apex_error.t_error_re-
sult is
  v_er apex_error.t_error_re-
sult;
begin
  v_er := apex_error.init_er-
ror_result ( p_error => p_error
);
  if p_error.ora_sqlcode =
-1438 then
    v_er.message := ,Es wurde
ein zu langer Wert eingegeben -
Bitte
überprüfen Sie Ihre Angaben.‘;
  else
    v_er.message := p_error.
message;
  end if;
  return v_er;
end;
    
```

Listing 2: Beispielcode für eine eigene Error-Handling-Funktion



Abbildung 6: Neuer Apex-Assistent „Data Loading“

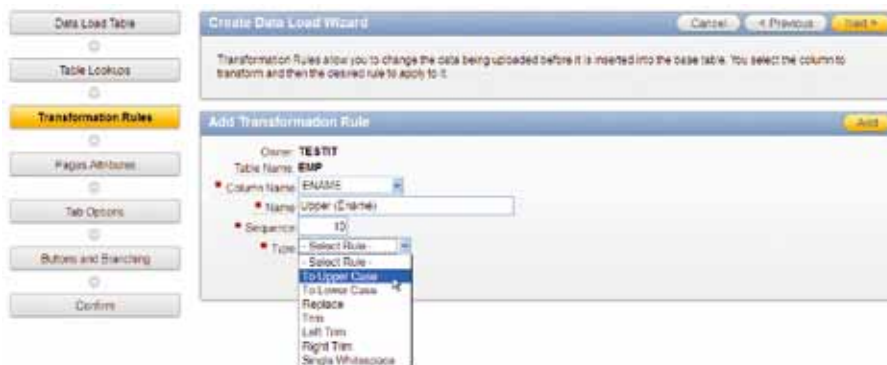


Abbildung 7: Hinterlegen einfacher Daten-Transformationen

Excel-Uploads für den Endanwender

In vielen Unternehmensanwendungen kommt es vor, dass Endanwender mehrere Datensätze auf einmal eintragen oder ändern müssen. Häufig liegen diese bereits in Spreadsheet- oder kommaseparierten Dateien vor. Apex bietet dem Entwickler schon seit dem ersten Release die Möglichkeit, solche Dateien in Tabellen hochzuladen – was jedoch fehlte, war ein einfacher Dialog

für den Endanwender. Apex 4.1 bietet hierfür den neuen Assistenten „Data Loading“, um eine Apex-Anwendung mit Dialogen zum Excel-Upload zu versehen (siehe Abbildung 6).

Dieser Assistent fügt der Apex-Anwendung gleich mehrere Seiten hinzu: Es wird ein vom Endanwender nutzbarer Workflow zum Hochladen kommaseparierter Dateien erstellt. Neben der reinen Upload-Funktionalität ist es möglich, zusätzliche Validierungen

Fazit

Apex 4.1 ist etwas mehr als nur eine Abrundung und Vervollständigung von Apex 4.0. Es sind auch einige wirklich neue Dinge dabei und über das hier Vorgestellte hinaus eine Fülle an Verbesserungen und Kleinigkeiten im Detail enthalten. Auch die zahlreichen Neuerungen im Bereich der Apex-Worksheets müssen einem eigenen Artikel vorbehalten bleiben; an

dieser Stelle sei auf die Online-Liste der neuen Features unter <http://apex.oracle.com/pls/apex/f?p=52663:1> oder unter <http://otn.oracle.com/apex> verwiesen.

Weitere Informationen

- [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>

Carsten Czarski
 carsten.czarski@oracle.com
<http://twitter.com/cczarski>
<http://sql-plsql-de.blogspot.com>



Vorschau auf die nächste Ausgabe

Die Ausgabe 06/2011 hat das Schwerpunktthema:

„Enterprise Manager“

Folgende Themen sind im Fokus:

- Erfahrungen im Umgang mit dem jetzigen Enterprise Manager
- Die neue Version des Enterprise Managers mit folgenden Artikeln:
 - Alle Neuheiten im Überblick
 - Configuration Management
 - Administration der Exadata Database Machine
 - Cloud Management
 - New Features for Database Management

Darüber hinaus bringen wir zahlreiche weitere Artikel mit Best Practices im Umgang mit den Oracle-Produkten.

Die Ausgabe 06/2011 erscheint am 2. Dezember 2011

Das nächste Schwerpunktthema der DOAG News:
 01/2012: Tuning, Performance



Die Oracle Experten

Machen Sie doch was Sie wollen!

Mit unseren Remote Datenbank und System Administratoren entlasten Sie Ihr IT-Team, um sich wieder den eigentlichen Aufgaben Ihres Unternehmens widmen zu können.

Schon ab € **584,-** exkl. MwSt. pro Monat unterstützen wir Sie mit professioneller Fernwartung von Oracle Datenbanken und Oracle Application Servern.
Auf Wunsch auch rund um die Uhr.

Automatische Systemüberwachung und periodische Health Checks sowie Backup/Recovery-Tests geben Ihnen genaue Auskunft über den aktuellen Status Ihrer Datenbanken und garantieren die Sicherheit und Verfügbarkeit Ihrer Unternehmensdaten.



Specialized
 Oracle Database
 Oracle Database Performance Tuning
 Oracle Linux

www.dbconcepts.at

Tel.: +43 1 890 89 99 0

office@dbconcepts.at



bit.ly/remotedba