

Kurzanleitung zum Oracle9iDS Forms Developer

Ralf Tautenhahn

26. Februar 2004

Inhaltsverzeichnis

1	Einleitung	3
1.1	Aufbau und Bestandteile von Formularen	3
1.2	Starten des Forms Builders	3
2	Entwicklung eines einfachen Eingabe/Abfrageformulars	4
2.1	Einleitung	4
2.2	Datenblock erstellen	4
2.3	Leinwand(Canvas) erstellen	7
2.4	Anzeigen des Formulars	10
3	Erstellung eines Master-Detail-Formulars	11
3.1	Einleitung	11
3.2	Detail-Rahmen einfügen und verknüpfen	11
3.3	Ein zweiter Detail-Rahmen	15
4	Wertelisten	17
4.1	Erstellen einer Werteliste für ein Formularfeld	17
5	Trigger	20
5.1	Einleitung	20
5.2	1. Beispiel : Abfrage ausführen	21
5.3	2. Beispiel : Werte aus anderer Tabelle abfragen	22
5.4	3. Beispiel : Ein Sort-by Menü	23
6	Einige Fehlermeldungen	25

1 Einleitung

1.1 Aufbau und Bestandteile von Formularen

Der Forms Developer als Teil der Internet Developer Suite dient zum Erstellen von maskenbasierten Anwendungen (Formularen) zur Dateneingabe und -abfrage von Oracle Datenbanken. Ein Formular wird durch eine Zusammenstellung von Objekten definiert, z.B.

- Datenblöcke (aus Tabellen oder Views)
- Steuerblöcke (z.B. Buttons)
- Leinwände (Canvas)
- Datensatzgruppen (Record group)
- Wertelisten (LOV's)
- Trigger
- Fenster
- ...

1.2 Starten des Forms Builders

Vor dem eigentlichen Start des *Forms Builders* muss *Start OC4J* (Oracle Components For Java) ausgeführt werden. Damit steht ein Java-Servlet bereit, das nötig ist um die erstellten und compilierten Forms (=*.fmx Dateien) im Browser auszuführen. Nach dem Start des *Forms Builders* sollte man sich zunächst über Datei|Anmelden einloggen. Erfolgt keine Fehlermeldung kann man davon ausgehen, daß man eingeloggt ist und mit der Arbeit beginnen. (Evtl. muss beim ersten Start noch über Bearbeiten|Voreinstellungen|Laufzeit der Web-Browser-Speicherort eines java-fähigen Browsers bestimmt werden.)

2 Entwicklung eines einfachen Eingabe-/Abfrageformulars

2.1 Einleitung

Benutzt wird hier das bei Oracle mitgelieferte Beispielschema Sales History *SH* (CUSTOMERS,COUNTRIES,PRODUCTS,SALES..). Es können natürlich aber auch andere tables verwendet werden. Zunächst soll als Beispiel ein einfaches Eingabe-/Abfrage-Formular erstellt werden. Dank des mitgelieferten Datenblock- bzw. Layout-Assistenten ist ein schneller Erfolg dabei nur schwer zu verhindern.

2.2 Datenblock erstellen

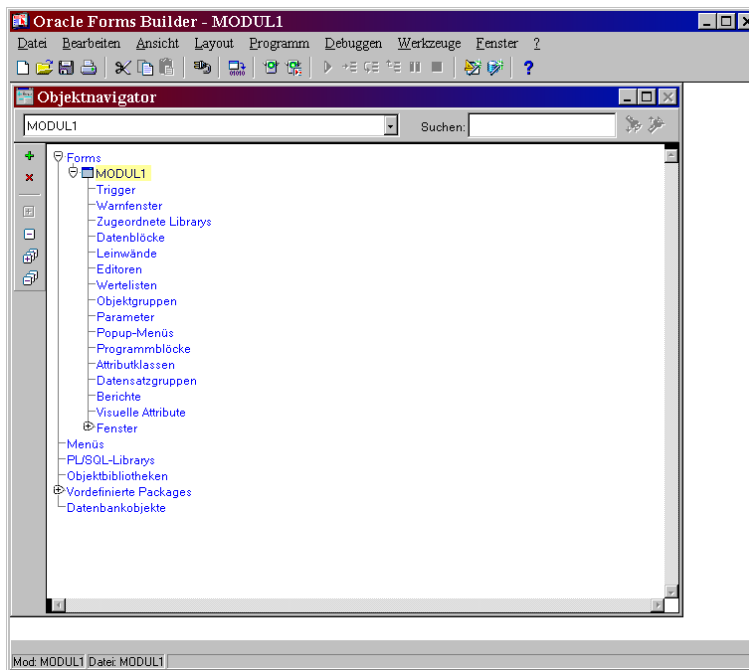


Abbildung 1: Der Objektnavigator.

Es begrüßt uns zunächst der Objektnavigator mit einem neuen, noch leeren Formular(Abbildung 1).

Um einen neuen Block zu erstellen, den Assistenten über Werkzeuge|Datenblock-Assistent starten. Da eine Tabelle bzw. ein View als Datenquelle dienen soll, den ersten Punkt auswählen (Abbildung 2).

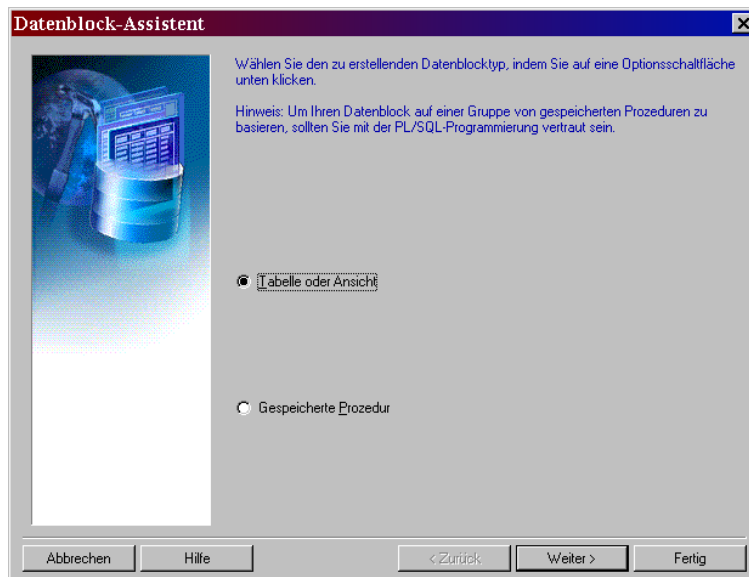


Abbildung 2: Datenblock-Assistent

Im nächsten Dialogfenster lassen sich nun Spalten aus einer Tabelle auswählen, die einen Datenblock bilden werden und Grundlage für die Darstellung im Formular sind (Abbildung 3). Nachdem man dem Block noch einen Namen

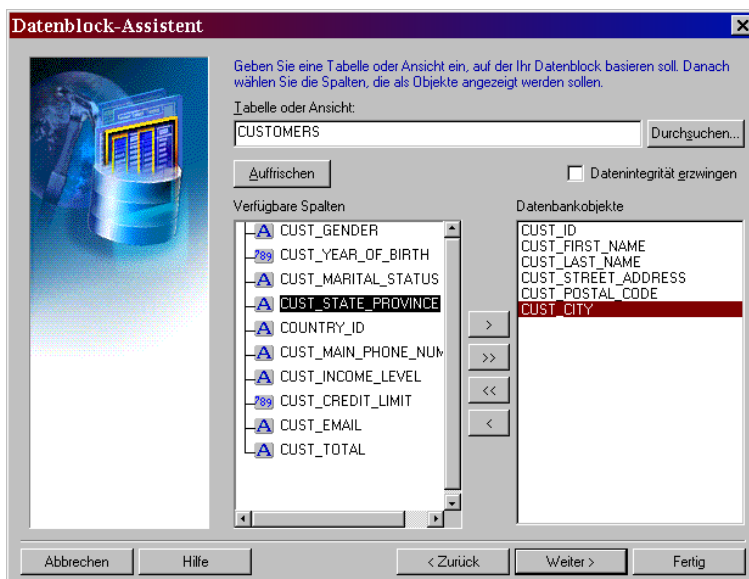


Abbildung 3: Datenblock-Assistent

verpaßt hat, ist dieser schon vollständig beschrieben und der Forms Builder bietet an, gleich noch den Layout-Assistenten zu starten.

2.3 Leinwand(Canvas) erstellen

Wenn dies getan wird, fragt der Assistent zunächst nach dem Typ der zu erstellenden Leinwand. Wir belassen es bei den Voreinstellungen (Abbildung 4).

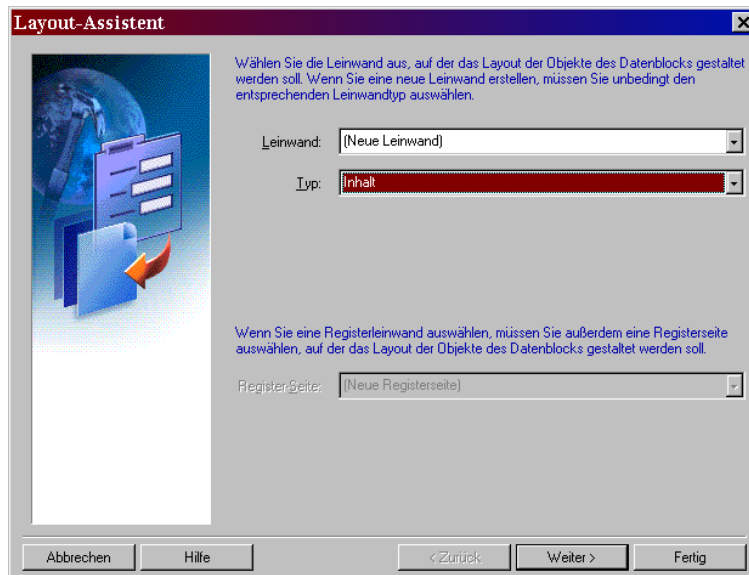


Abbildung 4: Layout-Assistent

Im nächsten Dialogfenster lässt sich nun auswählen, welche Elemente des soeben erstellten Datenblocks auf der Leinwand dargestellt werden sollen (Abbildung 5).

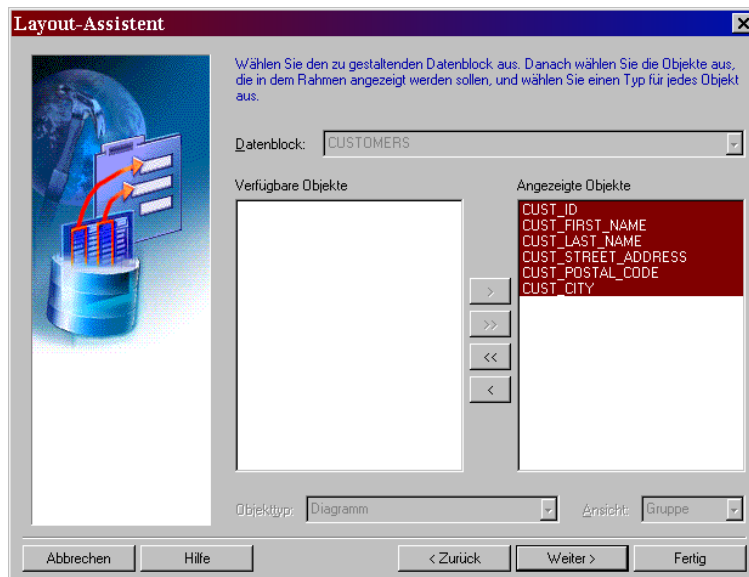


Abbildung 5: Layout-Assistent

Im darauf folgenden Fenster hat man die Möglichkeit Bezeichnung und Größe der einzelnen Formularfelder zu ändern. Dies läßt sich aber viel anschaulicher später im Editor erledigen, also beläßt man es auch hier am besten bei den vorgeschlagenen Werten. Danach als Layout-Stil Form auswählen. Im

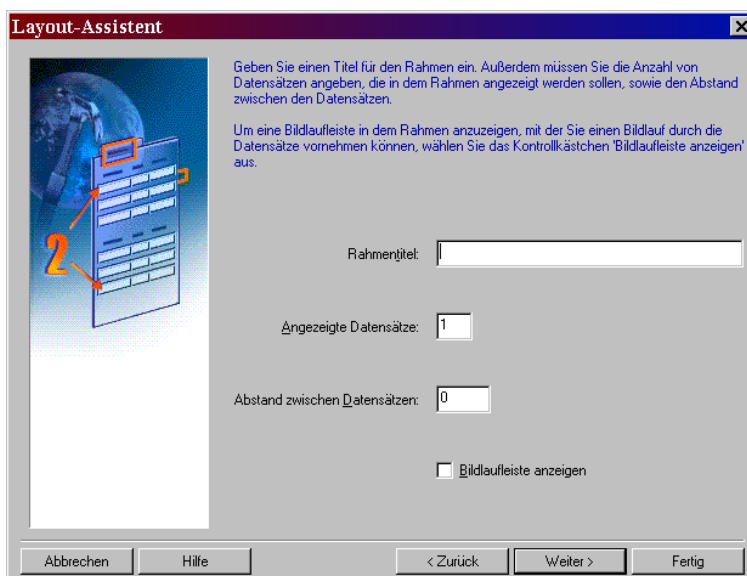


Abbildung 6: Layout-Assistent

letzten Dialog-Fenster (Abbildung 6) kann dem erzeugten Rahmen ein Titel zugewiesen werden. Die Anzahl der angezeigten Datensätze kann man bei 1 belassen, ansonst werden die Formularfelder vervielfacht. Eine Bildlaufleiste innerhalb des Formularrahmens ist im einfachen Fall ebenfalls nicht erforderlich.

Nach Beendigung des Assistenten wird das Ergebnis im Layouteditor präsentiert. Jetzt könnte man noch ein wenig am Layout feilen, oder erst mal das Formular 'starten'.

2.4 Anzeigen des Formulars

Man wähle Programm|Form ausführen bzw CTRL + R. Nun sollte das Formular im Browser auftauchen(Abbildung 7). Man befindet sich nun standardmäßig im Eingabemodus, könnte also neue Datensätze eingeben. Um in den Abfragemodus zu wechseln, Abfrage|Eingabe wählen bzw. F11. Nun könnte man eine Abfrage eingeben, beispielsweise alle Kunden, die mit 'K' beginnen. Nach dem Eintrag von 'K%' in *Cust Last Name* startet man die Abfrage mit STRG+F11 oder Abfrage|Ausführen. Durch die Treffermenge navigiert man mit Cursortasten Oben/Unten oder den Vorheriger/Nächster-Datensatz-Icons. Mit STRG+K kann man sich die übrigens die aktuelle Tastenbelegung im Formular ansehen.

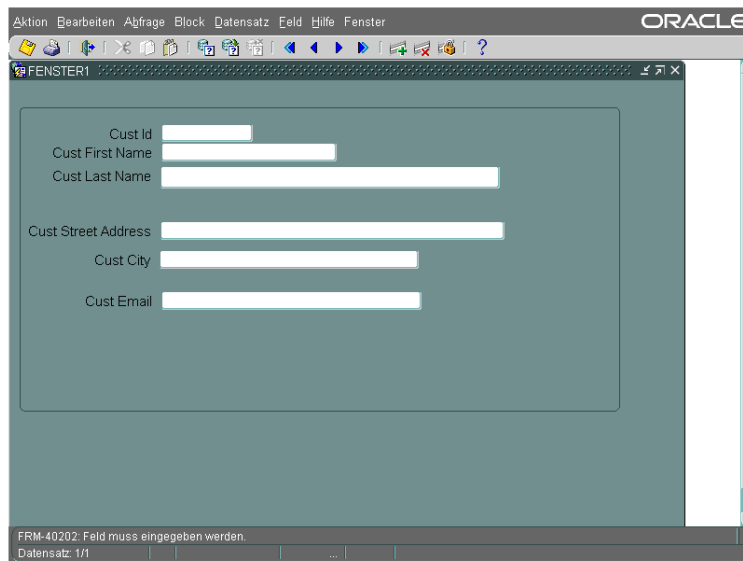


Abbildung 7: Formular im Browser-Fenster

Es können natürlich auch mehrere Felder für die Abfrage ausgefüllt und logische Operatoren wie $<$, $<=$, $>$, $>=$ und $!=$ benutzt werden.

3 Erstellung eines Master-Detail-Formulars

3.1 Einleitung

Das eben erstellte Formular soll nun noch um zwei Rahmen erweitert werden, beispielsweise sollen zum ausgewählten Kunden noch die an ihn verkauften Artikel und später außerdem Artikeldetails dargestellt werden. Der größte Teil der Arbeit läßt sich auch hier wieder an die Assistenten abschieben.

3.2 Detail-Rahmen einfügen und verknüpfen

Ohne daß einer der vorhandenen Datenblöcke markiert ist, startet man den Datenblock-Assistenten (so wird ein neuer Block angelegt, ansonsten nur der markierte verändert). In den nächsten Dialogen wie gewohnt Tabelle und dann die für den zweiten Datenblock nötigen Spalten (im Beispiel aus SALES) übernehmen. Der nächste Dialog bietet an, die Verknüpfung automatisch zu erstellen, was aber nur im eindeutigen Fall funktioniert (Abbildung 8).

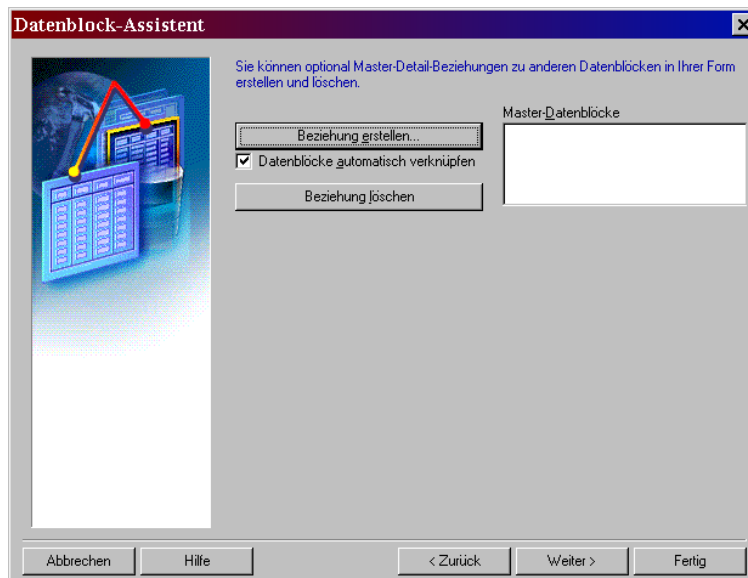


Abbildung 8: Datenblock-Assistent

Um das zu überprüfen, 'Beziehung erstellen' wählen. Jetzt sollte ein Vorschlag für die Verknüpfung auftauchen (Abbildung 9). Wenn sich der As-

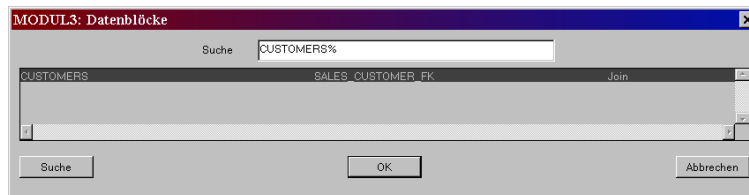


Abbildung 9: Datenblock-Assistent

sistent weigert ('Keine Master-Datenblöcke verfügbar.') so hat man wahrscheinlich eine Tabelle ausgewählt die keine direkte Beziehung zum 'Master'-Datenblock bietet oder es existieren schon mehrere Datenblöcke, dazu später mehr. Ansonsten kann man nach einem OK die JOIN-Bedingung überprüfen (Abbildung 10) und dann den Layout-Assistenten wie gewohnt benutzen. Nun ist es aller-

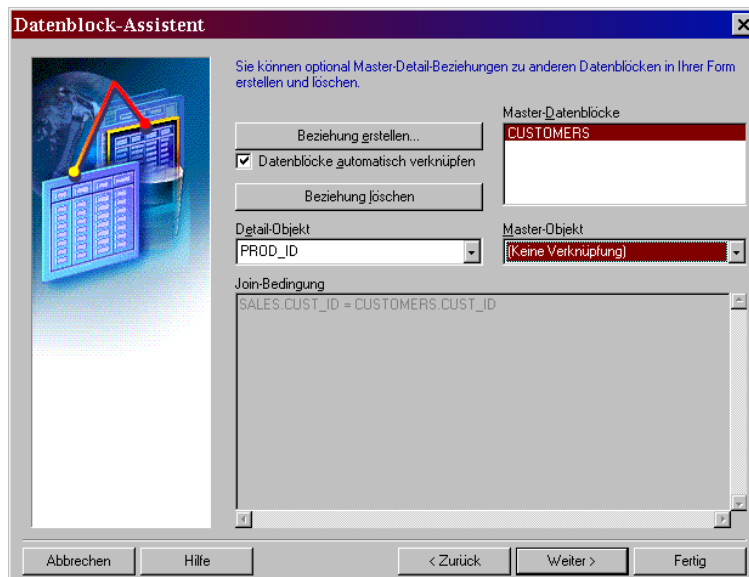


Abbildung 10: Datenblock-Assistent

dings günstig (da es sich im Beispiel um eine 1:N Beziehung handelt) für den Detail-Rahmen als Layout Tabellenform zu wählen und auch die Anzahl der angezeigten Datensätze auf z.B. 5 oder 10 zu setzen. Eine Bildlaufleiste wäre

nun auch recht praktisch. Das fertige Formular zeigt an, welche Produkte an den ausgewählten Kunden verkauft wurden (Abbildung 11). Wahrscheinlich

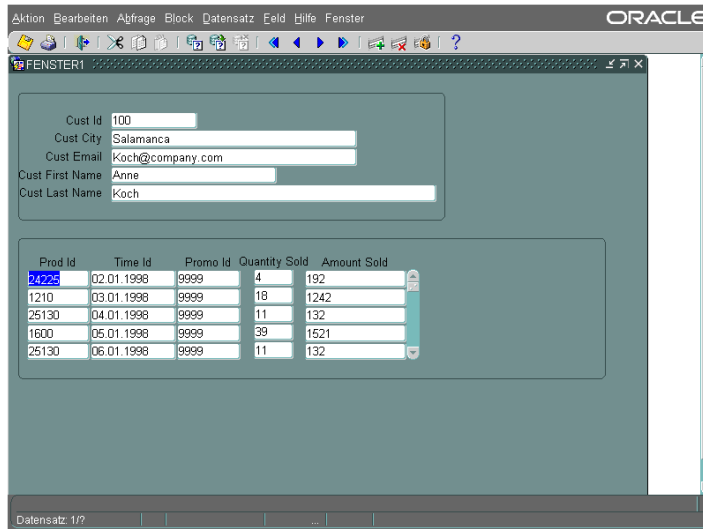


Abbildung 11: Master-Detail-Formular im Browser-Fenster

wäre es sinnvoll, die Ergebnisse im Detail-Rahmen nach einem Attribut sortiert darzustellen. Dazu im Objektnavigator die Attributpalette des zugehörigen Datenblocks anzeigen lassen und bei Datenbank | ORDER-BY-Klausel z.B. 'TIME_ID DESC' eingeben, um die aktuellsten Umsätze zuerst auszugeben (Abbildung 12).

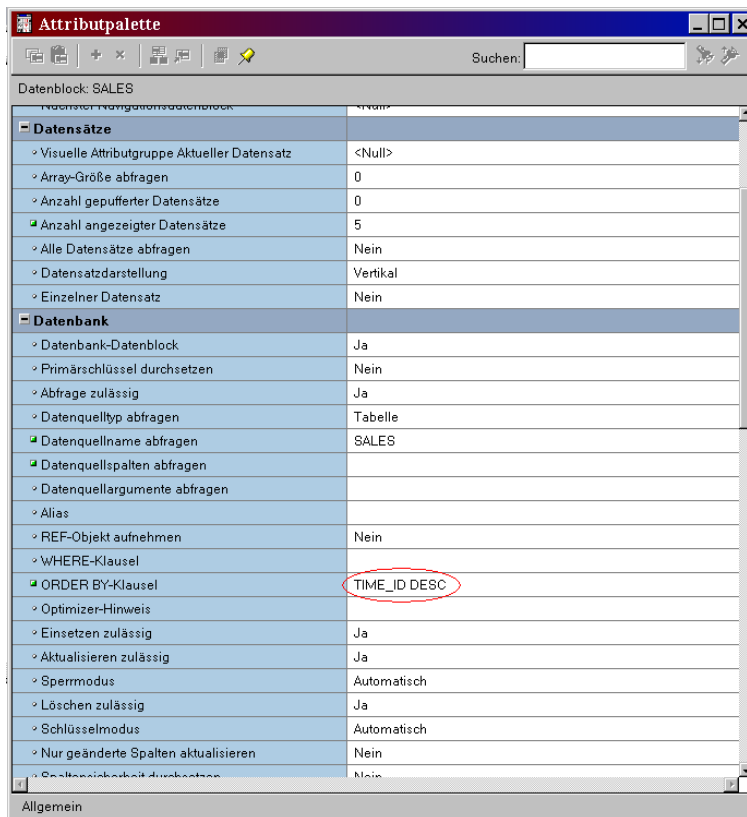


Abbildung 12: Attributpalette des Datenblocks

3.3 Ein zweiter Detail-Rahmen

Da die `PROD_ID` alleine nicht interessant genug ist, könnte man noch eine Relation bemühen, um ein paar Eigenschaften des Produktes darzustellen, z.B. dessen Bezeichnung und Kategorie. Dazu ist ein weiterer Datenblock nötig, der wiederum über den Assistenten neu erstellt wird. Man wähle `PROD_ID` und dazu einige Attribute für den neuen Datenblock. Die Master-Detail-Verknüpfung mit der `SALES`-Tabelle muß jetzt 'von Hand' erfolgen. Also Häkchen aus 'Datenblöcke automatisch verknüpfen' entfernen und 'Beziehung erstellen' wählen. Die neue Beziehung soll auf einer JOIN-Bedingung basieren, dessen JOIN-Partner dann ausgewählt werden kann. Nach Auswahl von `SALES` kann die Beziehung dann festgelegt werden (Abbildung 13).

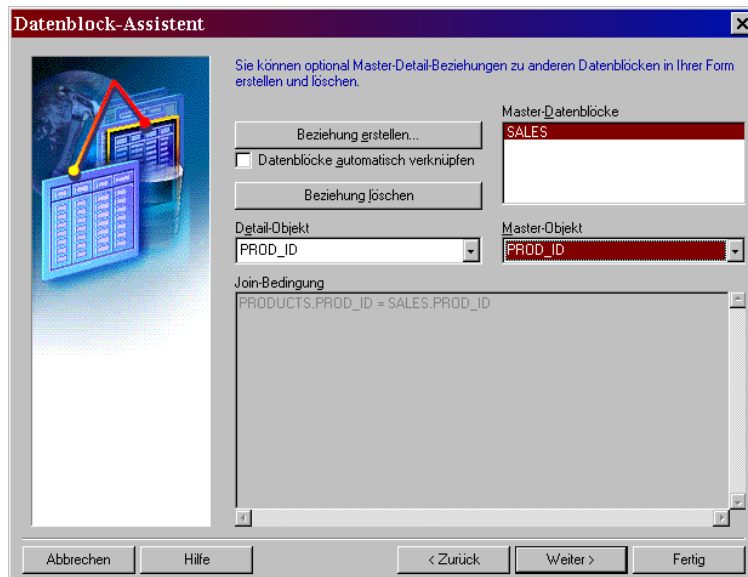


Abbildung 13: Master-Detail-Beziehung

Das Ergebnis leistet dann das Gewünschte (Abbildung 14).

Prod Id	Time Id	Promo Id	Quantity Sold	Amount Sold
4210	29.11.2000	9999	8	72
4310	28.11.2000	9999	39	2574
1210	27.11.2000	9999	7	483
335	26.11.2000	9999	37	3034
325	25.11.2000	9999	3	48

Product

Prod Name: Lucky Baggy Fit Jean
Prod Desc: this is the famous Lucky Baggy Fit Jean in color blue of size XXXL
Prod Subcategory: Jeans - Men
Prod Category: Men

Abbildung 14: Master-Detail-Formular im Browser-Fenster

4 Wertelisten

4.1 Erstellen einer Werteliste für ein Formularfeld

Eine Werteliste (LOV, List of Values) dient zur Vereinfachung der Eingabe/Abfrage für Felder mit einer diskreten Anzahl von Werten. Als Beispiel soll ein Formular für die Tabelle PRODUCTS stehen, in dem die Produktkategorie über eine Werteliste ausgewählt werden kann. Nach dem Erstellen eines Formulars mit den gewünschten Attributen, den Wertelisten-Assistenten aus dem Wekrzeuge-Menü starten (oder aus dem Objektnavigator unter Werteliste). Dann kann zum Generieren der Liste der Query-Builder benutzt werden, oder man gibt das SQL-Statement direkt ein, z.B. 'SELECT DISTINCT PRODUCTS.PROD_CATEGORY FROM PRODUCTS ORDER BY 1' (Abbildung 15).

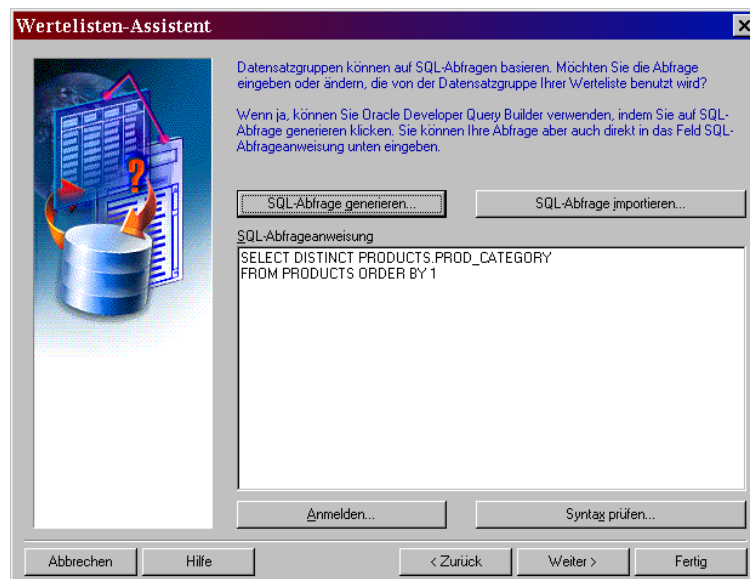


Abbildung 15: Wertelisten-Assistent

Die daraus erstellte Datensatzgruppe kann im nächsten Fenster in die Werteliste übernommen werden. Nach Beendigung des Assistenten muss die erstellte LOV noch mit dem entsprechenden Formularfeld verknüpft werden. Dies geschieht in der Attributpalette des Objekts, im Beispiel von PROD_CATEGORY. Dort einfach im Eintrag Werteliste die Bezeichnung der erstellten LOV auswählen, z.B. WERTELISTE6 (Abbildung 16). Wenn das Formular jetzt gestartet wird, ist

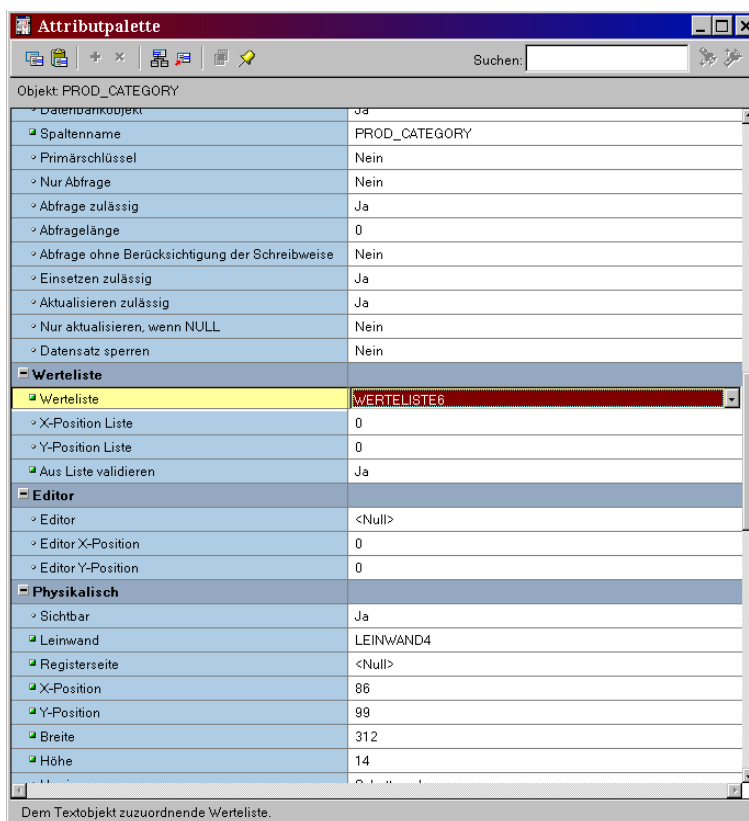


Abbildung 16: Attributpalette des Formularfelds

zwar im entsprechenden Feld eine Werteliste verfügbar (Erkennt man daran, dass in der unteren Statuszeile jetzt die Bemerkung 'Werteliste' auftaucht) aber sonst passiert zunächst nichts. Auch nach dem Aufruf der Liste mit STRG+L wird die Auswahl nicht ins Feld übernommen. Deshalb in der Attributpalette der Werteliste unter Funktional|Attribute Spaltenzuordnung die Verknüpfung vornehmen, indem das betreffende Formularfeld als Rückgabeobjekt gewählt wird (Abbildung 17). Bei der Gelegenheit gleich noch

'Automatische Auswahl' auf 'Ja' setzen.

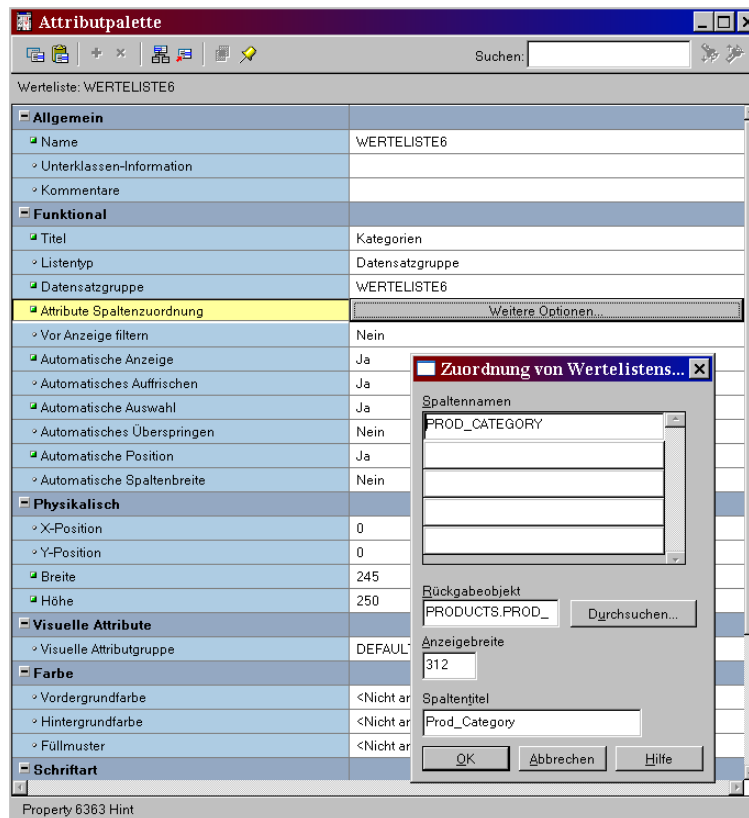


Abbildung 17: Attributpalette der Werteliste

Nun sollte die Werteliste im Feld automatisch auftauchen und der Wert dann auch übernommen werden (Abbildung 18).

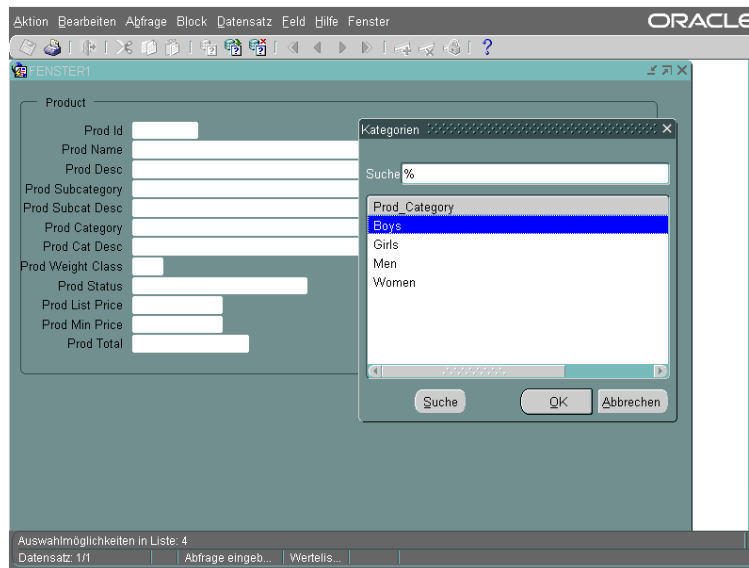


Abbildung 18: Formular mit Werteliste

5 Trigger

5.1 Einleitung

Ein Trigger ist eine Folge von PL/SQL-Anweisungen, die ausgeführt wird, wenn ein Ereignis eintritt. Das Ereignis kann direkt mit einer vom Benutzer durchgeführten Aktion zusammenhängen, wie das Betätigen einer Schaltfläche. Oder das Ereignis ist indirekt mit einer vom Benutzer durchgeführten Aktion verbunden, wie der Zeitpunkt vor der Ausführung einer Abfrage. Trigger können auf drei Ebenen definiert werden:

- Auf Formularebene
- Auf Blockebene
- Auf Feldebene

Häufig enthält ein Formular auf jeder Ebene einen oder mehrere Trigger. [1]

5.2 1. Beispiel : Abfrage ausführen

Für manche Anwendungen könnte es praktisch sein, wenn gleich beim Öffnen des Formulars Datensätze angezeigt werden. Dazu reicht ein sehr einfacher Trigger. Im Objektnavigator einen neuen Trigger vom Typ WHEN-NEW-FORM-INSTANCE im 'Wurzelverzeichnis' des Formulars anlegen. Im PL/SQL-Editor einfach `execute_query;` eintragen.(Abbildung 19).

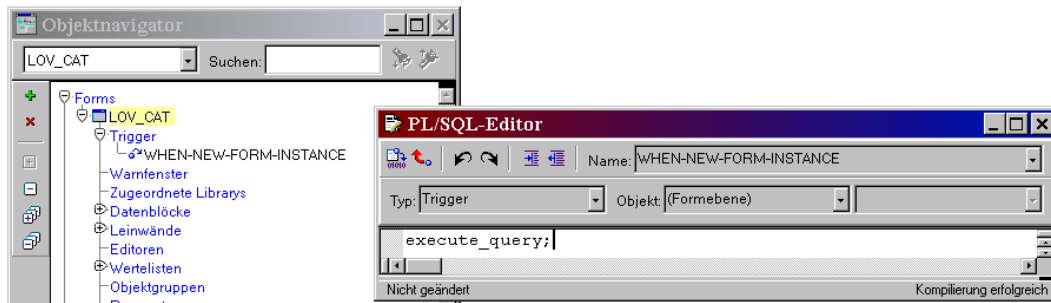


Abbildung 19: Trigger anlegen

5.3 2. Beispiel : Werte aus anderer Tabelle abfragen

Bei der Anzeige der Kundenadresse wird aus CUSTOMERS als Information über das Land zunächst nur die COUNTRY_ID bereitgestellt. Ein neues Textfeld soll den dazugehörigen Ländernamen aus der Tabelle COUNTRIES anzeigen. Dazu zunächst im Layouteditor ein neues Textobjekt erstellen (Abbildung 20). In dessen Eigenschaften (Attributpalette) den Namen des Ob-

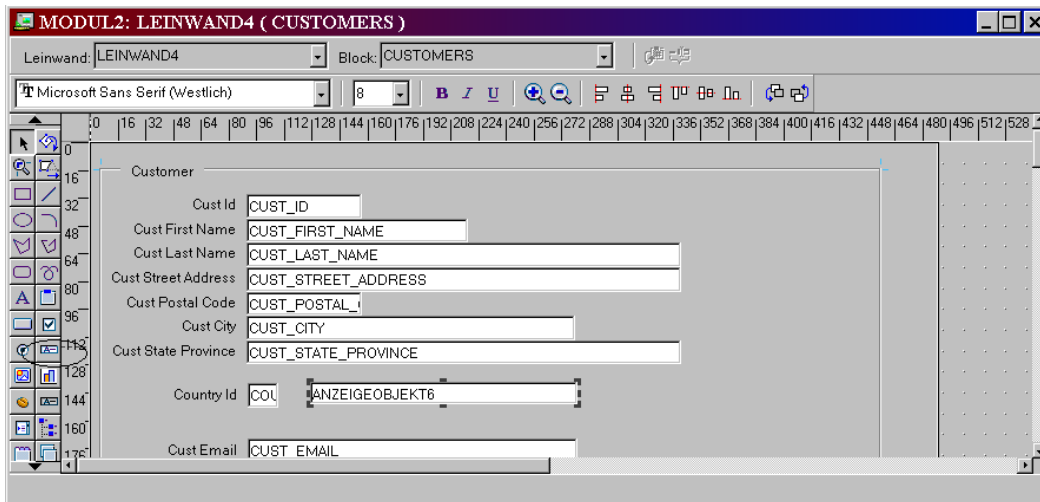


Abbildung 20: Neues Textobjekt

jekts in z.B. COUNTRY ändern und die Eigenschaft Datenbank|Datenbankobjekt auf Nein setzen.

Im Datenblock einen neuen Trigger vom Typ POST-QUERY anlegen, dessen PL/SQL Code könnte folgendermaßen lauten:

```
SELECT country_name  
INTO :customers.country  
FROM countries  
WHERE country_id = :customers.country_id;
```

(Abbildung 21).

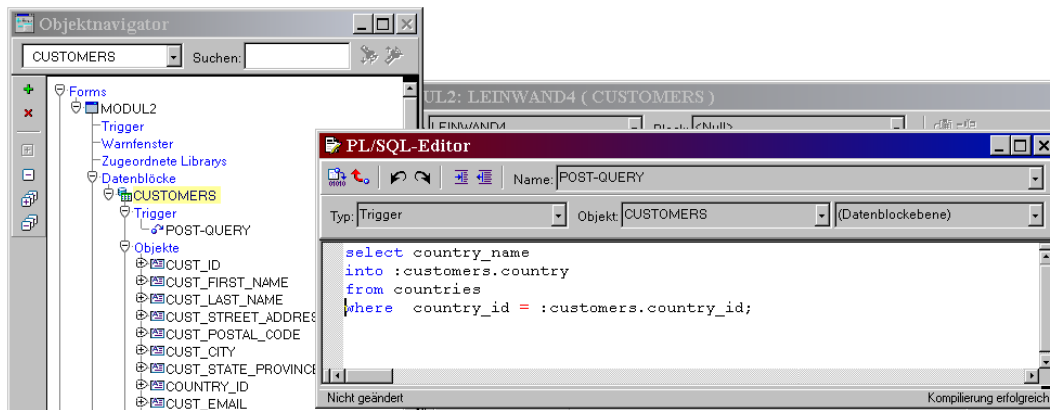


Abbildung 21: Trigger anlegen

5.4 3. Beispiel : Ein Sort-by Menü

Für eine tabellarische Darstellung soll ein List-Item erstellt werden, der die Sortierung ändert (Abbildung 22). Die Idee 'Dynamic Sorting' ist entnommen aus [2].

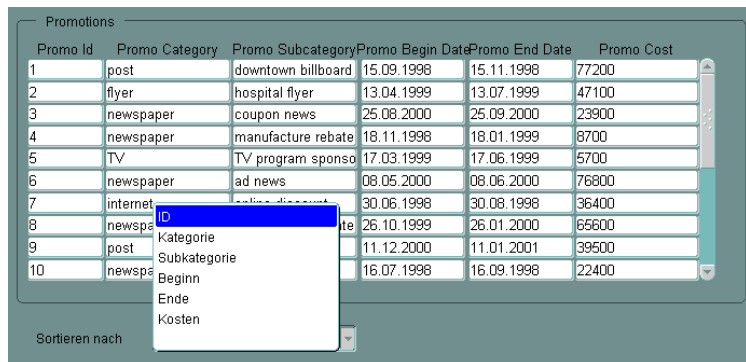


Abbildung 22: Sort-by Menü

Zunächst muss natürlich ein Listenobjekt erstellt werden, dies geschieht am einfachsten im Layout-Designer. Wenn das Formular als Tabelle erstellt wurde, wird dann automatisch ein Listenobjekt erzeugt, daß nicht nur ein, sondern mehrere Listen darstellt. Dies läßt sich über die Objekteigenschaften (Datensätze|Anzahl angezeigter Objekte = 1) wieder beheben. Über Funktional|Elemente in Liste können dann die Elemente der Liste festgelegt

werden. Die Bezeichnung ist frei wählbar, nur der eingetragene Elementwert muß, damit der Trigger später funktioniert, genau dem jeweiligen Spaltennamen des zu sortierenden Datenobjekts entsprechen (Abbildung 23).

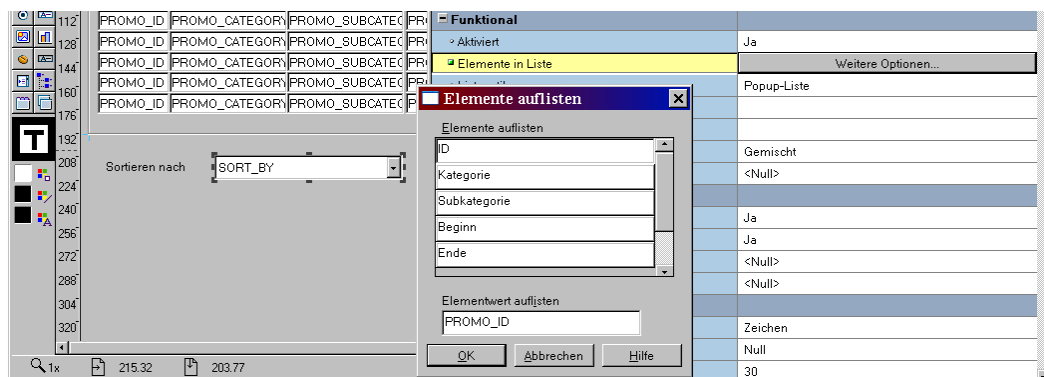


Abbildung 23: Elemente der Liste festlegen

Dem Listenobjekt kann jetzt im Objektnavigator ein SmartTrigger zugeordnet werden, vom Typ WHEN-LIST-CHANGED:

```
set_block_property('PROMOTIONS',ORDER_BY, :sort_by);
go_block('PROMOTIONS');
execute_query;
```

Den Code am besten sofort probeweise compilieren (Button links oben im PL/SQL-Editor). Damit das Formular mit einer Abfrage startet, kann auch der Trigger aus Bsp.1 ins Formular aufgenommen werden. In den Eigenschaften der Liste muß unbedingt noch Datenbank|Datenbankobjekt auf 'Nein' gesetzt werden, dann sollte die Sache funktionieren.

6 Einige Fehlermeldungen

Fehler Nr.	Fehlermeldung	Abhilfe
FRM-10757	Keine Master-Datenblöcke verfügbar.	Häkchen aus 'Datenblöcke automatisch verknüpfen' entfernen. 'Beziehung erstellen..' wählen und selbst verknüpfen.
FRM-30087	Formdatei (Name) kann nicht erstellt werden.	Noch offenes Formular im Browser über Aktion/Beenden schließen, Browser ebenfalls beenden.
FRM-30087	Formdatei (Name) kann nicht erstellt werden.	Hat nicht geholfen? Dann Task <i>ifweb90</i> beenden.
FRM-90928	Positionsparameter nach Schlüsselwort in der Befehlszeile	Formular in einem Pfad abspeichern, der keine Leerzeichen enthält

Literatur

- [1] David Lockman, Norbert Debes. *Oracle 9i Datenbankentwicklung in 21 Tagen . Design und Implementierung*. Markt und Technik, 2002.
- [2] Peter Koletzke, Dr. Paul Dorsey. *Oracle Developer Advanced Forms & Reports*. Osborne/McGraw-Hill, Berkeley, California, 2000.