

Forms fits4all

Stephan La Rocca
TEAM GmbH
Paderborn

Schlüsselworte

Oracle Forms Modernisierung, Customizing, Entwicklungskonzepte und Werkzeuge

Einleitung

Wer liebt sie nicht? Die hochauflösende, brillante Darstellung, die einem die längst gängigen HD-Monitore bieten? Doch wer eine Oracle Forms-Anwendung betreibt, hat leider nicht viel von den höheren Auflösungen. Wird das Forms-Fenster vergrößert, sieht der Anwender meist grau - die Farbe des Hintergrundes.

Oracle Forms-Applikationen werden vom Entwickler für eine feste Auflösung optimiert erstellt und erlauben keine Anpassungen des Anwenders, an die Auflösung des eigenen Bildschirms.

Diese Maxime zu brechen ist TEAM angetreten und hat den FormsResizer entwickelt.

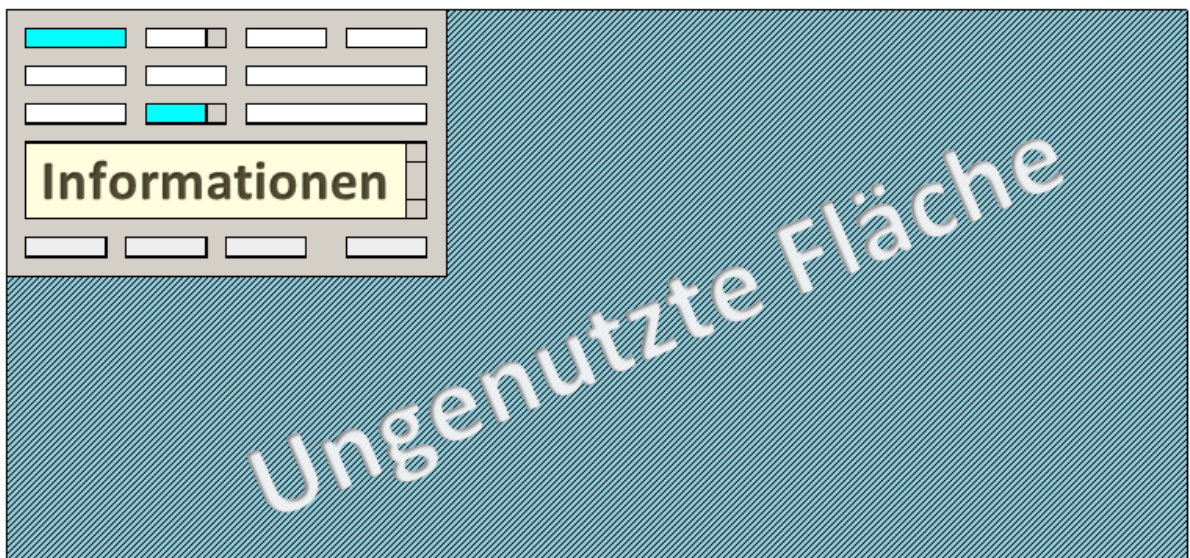


Abb. 1: Forms auf höheren Auflösungen

Aufgabenstellung

Für die Forms-Applikationen haben wir schon länger die Möglichkeit geschaffen, das Layout der Dialoge zur Laufzeit aus der Datenbank zu lesen und somit recht komfortable und einfach benutzerspezifische Darstellungen der Dialoge zu unterstützen. Im WHEN-NEW-FORM-INSTANCE-Trigger eines jeden Dialoges werden alle sichtbaren Objekte auf der Leinwand neu gemalt. Die Übernahme der Layoutinformationen aus einem hierarchischen Datenmodell ist so optimiert, dass die Dialogwechselzeiten nur weniger als eine Sekunde mehr in Anspruch nehmen. Somit kann der Anwender die Breite und Reihenfolge der Spalten in einer Tabelle selbst bestimmen und sogar ganze Felder aus- oder wieder einblenden. Ein getrennter Dialog, der über eine

Objektbibliothek in jede FMB vererbt wird, hilft dem Anwender bei der Gestaltung seiner Layoutanpassungen zur Laufzeit.

Es wäre alles perfekt und völlig ausreichen, wenn, ja wenn nicht eine einzige Einschränkung dem Entwickler das Leben erschweren würde – die Anzahl der Datensätze in einem Block können zur Laufzeit nicht verändert werden, was eine vertikale Skalierung von Tabellen unmöglich macht. Es fehlt ein einziges `SET_BLOCK_PROPERTY`, dann könnten Sie sich die folgenden drei Seiten einfach sparen.

Lösung

Somit muss für die Aufgabenstellung eine Entscheidung der zu unterstützenden Auflösung bereits in der FMB berücksichtigt werden. Unterschiedliche Auflösungen führen somit zu einer Vervielfachung der FMBs und das würde aber bedeuten, dass der Entwickler mehrfach den gleichen Dialog warten müsste, nur mit jeweils unterschiedlichen Layoutkomponenten.

Damit dieses Verfahren wartbar bleibt, haben wir auf Basis des Forms API-Masters der Firma ORCL-Toolbox ein Skript geschrieben, welches aus einer Basis FMB verschiedene Ziel-FMBs erstellen kann. Alle identisch in der Logik und nur unterschiedlich im Layout. Der Entwickler verändert weiterhin nur eine FMB und im Deployment-Prozess verändert dieser Schritt die Dateien.

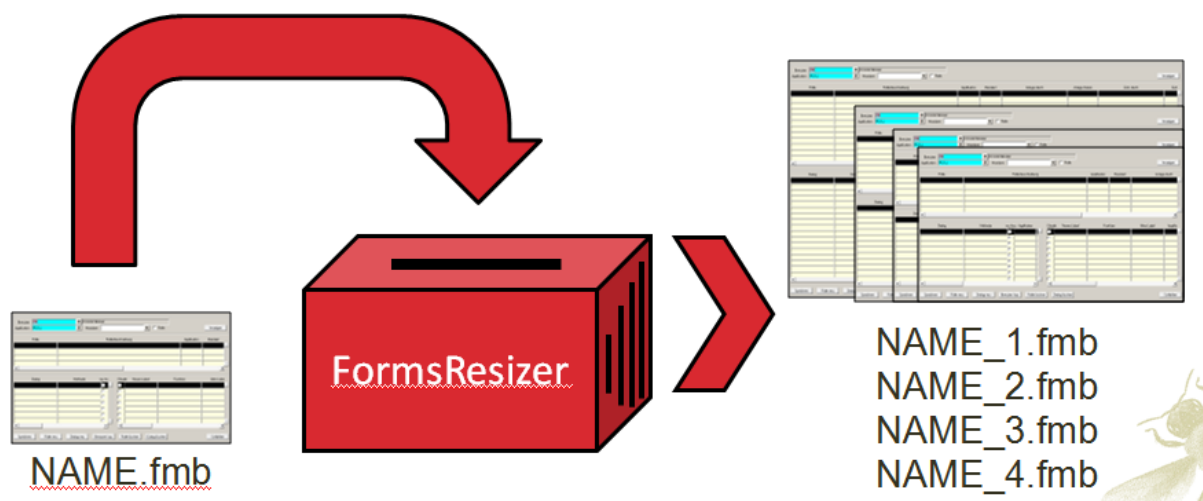


Abb.2: Das Vorgehen

Der Dialog wird dabei nicht nur einfach skaliert, sondern zunächst nach einem mathematischen Modell in verschiedene Layoutsektoren unterteilt. Diese Sektoren werden durch ein Parsen des Dialoges erkannt und im When-New-Form-Instance-Trigger als Kommentar hinterlassen. Zwei Tabellen nebeneinander, zwei Tabellen untereinander, eine Tabelle und darunter ein Single-Record-Block, usw. alles Muster, die der Parser fehlerfrei erkennen kann. Durch die automatische Dokumentation des erkannten Layout-Musters als Kommentar im Trigger hat der Entwickler die Information gleich parat, wie der FormsResizer die Skalierung für andere Auflösungen in dem jeweiligen Canvas plant.

Darüber hinaus können in den Kommentaren zusätzliche Optionen festhalten werden, mit dem der FormsResizer später entscheiden kann, nach welchem Verfahren der vorliegende Dialog skaliert

werden soll. Zu diesen Optionen gehört zum Beispiel, dass komplette Canvases von der Skalierung ausgespart werden.

Der FormsResizer kann als Executable unter Windows genutzt werden und in den Compile-Vorgang der Forms-Module oder in ein Check-in/Check-out-Skript der Versionsverwaltung integriert werden. Neben der Berechnung einzelner Module bieten wir durch ein geeignetes Skript die Möglichkeit auch auf einen Schwung alle Dialoge in einem Verzeichnis zu skalieren. Dabei werden nur die Dialoge angefasst, die sich auch tatsächlich nach der letzten Skalierung geändert haben.

Um manuelle Einschränkungen zu ermöglichen bieten wir für dieses Verfahren dann auch eine Blacklist an, in der komplette Dialoge ausgespart werden können.

Neben der Bereitstellung der Dialoge in den unterschiedlichen Auflösungen muss noch sichergestellt werden, dass innerhalb der Anwendung bei einem Dialogwechsel auch die passende FMX geöffnet wird. Dazu haben wir die Funktionen OPEN_FORM, CALL_FORM, NEW_FORM und GO_FORM erweitert und konform der Nameskonvention den Suffix der Auflösung an den Dateinamen gehängt. Ähnlich dem Verfahren mit Einführung der Webutil-Funktionen CLIENT_HOST, CLIENT_TEXT_IO, die durch Suchen&Ersetzen beim Kompilieren/Migrieren der Forms-Dialoge die Funktionen HOST, TEXT_IO ersetzt haben, nutzen wir TEAM_OPEN_FORM, TEAM_CALL_FORM, etc, die zunächst die gleiche Signatur besitzen, aber in ihrer Implementierung entsprechend der Namenskonvention den Namen der FMX-Datei, die aufzurufen ist, neu berechnen.

Mit dem Api-Master war ein einfaches „Suchen&Ersetzen“ dieser Aufrufe in den Dialogen möglich, wenn vermutlich auch schon der hauseigene Compiler mit der Option UPGRADE=YES und dem Pflegen der Search-Replace-Property-Datei ausgereicht hätte.

Nutzung in der Applikation

Für den Anwender kann in die Forms-Applikation ein Dialog integriert werden, mit dem er seine persönliche Vorliebe für die Skalierung festhalten kann. Beim ersten Start der Applikation wird über die WebUtils die aktuell eingestellte Auflösung des Bildschirms ausgelesen und ein vordefinierter Vorschlag gestartet.

In dem Dialog aber kann der Anwender aus beliebig vielen Skalierungen eine bestimmen, die dann für ihn gespeichert und beim nächsten Start wieder genutzt wird. Eine Mischung der Dialoggrößen innerhalb der Applikation ist derzeit nicht vorgesehen.

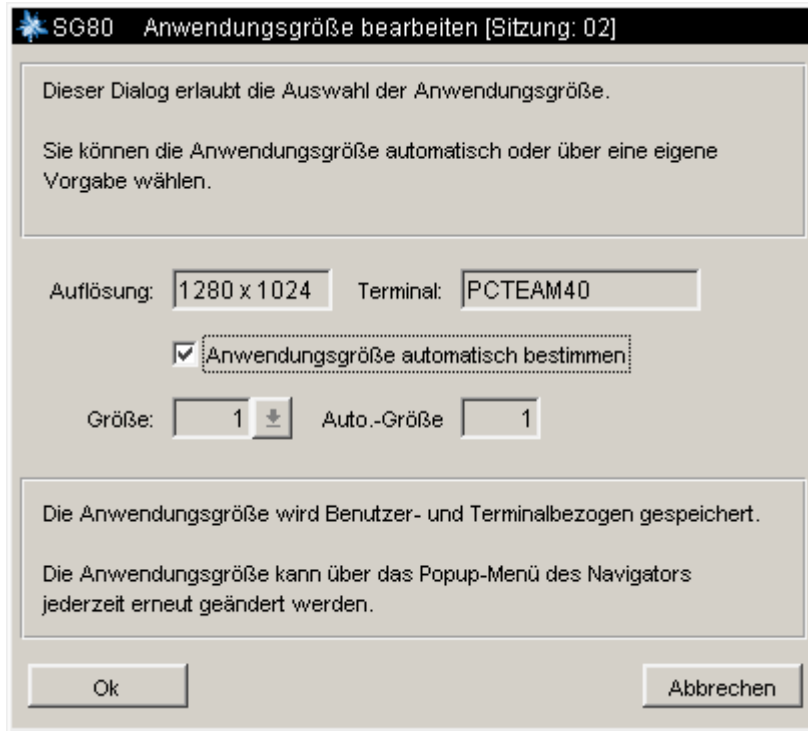


Abb.3: Aus Sicht des Users

Für unsere Applikation ProStore[®] ist dieses Verfahren etabliert und bietet mit den oben genannten Styleguide-Funktionalitäten für den Anwender eine wunderbare Möglichkeit, sich zur Laufzeit die Dialoge so anzupassen, dass Sie seinen individuellen Bedürfnissen gerecht werden.

Zur Zeit wird an einer reinen Java-basierten Lösung gearbeitet, die die Verwendung des Forms-API-Masters hinfällig macht. Darüber hinaus sind noch Verbesserungen im Erkennen von verschachtelten Layouts geplant.

Kontaktadresse:

Stephan La Rocca
TEAM GmbH
Hermann-Löns-Str. 88
D-33104 Paderborn

Telefon: +49 (0) 5254-80080
Fax: +49 (0) 5254-800819
E-Mail: sr@team-pb.de
Internet: www.team-pb.de