

SUBSCRIBE! - Flexibel entwickeln mit APEX Master Applications

**Andreas Wismann
WHEN OTHERS
D-41564 Kaarst**

Schlüsselworte

APEX, Subscription, Shortcuts

Einleitung

Eines der frühesten Features von Application Express ist, dass man eine Vielzahl von Komponenten über mehrere APEX-Anwendungen hinweg "subscriben" kann (in der deutschsprachigen APEX-Version: "abonnieren").

Komplett voneinander unabhängige APEX-Anwendungen im Unternehmen sind eher die Seltenheit. Komplett miteinander verwobene ebenfalls. Wie erreicht man eine möglichst flexible und den eigenen Ansprüchen genügende Kohärenz zwischen Applikationen? Durch Kopplung und Wiederverwendung. Mit "Subscription" steht dem Entwicklerteam ein mächtiges Werkzeug für diese Aufgabe zur Verfügung.

Ähnlich, aber nicht identisch mit dem "Subclassing" in Oracle Forms, ermöglicht Subscription die Weitergabe von Themes, Templates, Wertelisten, Shortcuts, Plugins etc., die in einer (oder mehreren) Master-Applikationen zentral gewartet werden.

Die Methode der Subscription wird meiner Erfahrung nach in APEX-Projekten oftmals gar nicht oder nicht konsequent genug eingesetzt, um sein volles Potenzial wirklich auszuschöpfen, denn mittels Subscription können einerseits die Kosten der Softwareerstellung und -wartung deutlich gesenkt werden, zum anderen ermöglicht sie erstaunliche Flexibilität bei der Anwendungsentwicklung.

Und das alles fast ohne Coding, sondern deklarativ.

Abgrenzung

Der Begriff Subscription taucht bei Oracle Application Express seit jeher auch im Zusammenhang mit Interactive Reports auf, wo der Benutzer bei Änderungen der Daten einen aktualisierten Bericht anfordern kann – diesem völlig anderen Thema widmet sich der Vortrag nicht.

Voraussetzungen für Subscription

Zunächst eine Übersicht, was man überhaupt subscriben kann (alphabetisch sortiert):

- Authentifizierungsschemas
- Autorisierungsschemas
- Lists Of Values
- Navigationsleisteneinträge
- Plugins
- Shortcuts
- Templates

- Themes

Wie funktioniert Subscription grundsätzlich?

Anhand eines einfachen Beispiels:

Sie verwenden in Anwendung A eine statische Werteliste (List of Values) namens „EINS-ZWEI-DREI“, welche Sie in mehreren anderen Anwendungen (B, C, ...) genau so übernehmen möchten. Voraussetzung: Die Anwendungen befinden sich alle im selben APEX-Workspace.

In der Anwendung B erstellen Sie dafür nun unter den Shared Components eine neue Werteliste:

Shared Components > List of Values > Create > As a Copy of an Existing List of Values

Im folgenden Auswahldialog navigieren Sie zu der Werteliste in Anwendung A und wählen als Kopiermethode „Copy and Subscribe“ aus.

Create List of Values ×

✔ ✔ ● **New Lists of Values**

Name the Lists of Values to be copied and select the copy option. Copy options include:

- **Yes.** Enter new List of Values name.
- **No.** Do not copy this List of Value.
- **Copy and Subscribe.** Copy and subscribe to this List of Values.

Application	From Name	To Name	Copy?
27245	EINS-ZWEI-DREI	Copy of EINS-ZWEI-DREI	No
27245	LOGIN_REMEMBER_USERNAME	Copy of LOGIN_REMEMBER_USERNAME	No

Dadurch wird eine Werteliste namens „Copy of EINS-ZWEI-DREI“ erstellt, die Sie alternativ (auch später noch) beliebig benennen können. Die neue Liste in Anwendung B ist nun durch Subscription mit der ursprünglichen Liste in Anwendung A verbunden.

Soweit, so gut. APEX hat sich zwar intern die Verbindung der beiden Listen gemerkt. Ansonsten verhalten sich beide Wertelisten aber scheinbar völlig unabhängig voneinander, denn:

- Sie können die Werteliste in Anwendung A ändern, ohne dass sich dies auf B auswirkt
- Sie können sogar die Werteliste in Anwendung B ändern, denn: Interessanterweise ist die neue Werteliste, obwohl sie als „abonnierte“ Kopie von A erstellt wurde, ein komplett selbständig bearbeitbares Anwendungselement. Nicht spricht dagegen, in B Änderungen vorzunehmen und sich damit beliebig weit vom Ursprung A zu entfernen.

Um den Subscription-Mechanismus zu nutzen, gibt es nun zwei Möglichkeiten:

- Sie gehen in Anwendung B in den Shared Components zur Bearbeitung der Werteliste „Copy of EINS-ZWEI-DREI“ und klicken unter „Subscription“ auf die Schaltfläche „Refresh LOV“: Achtung: Ohne jegliche Rückfrage wird nun die Werteliste quasi auf Werkseinstellungen zurückgesetzt, genau genommen auf den derzeitigen Zustand der Liste in A. Hierbei sollte man also sicherstellen, nicht versehentlich diese Schaltfläche anzuklicken, denn es erscheint tatsächlich keinerlei „Sind Sie sicher?“-Abfrage. Im Anschluss an die Rücksetzung sind die zuvor vorgenommen Änderungen an der Werteliste B unwiederbringlich verloren.
- oder: Sie Öffnen die Original-Werteliste in Anwendung A. Dort erhalten Sie den Hinweis, dass es andere Wertelisten gibt, die von A subscriben:

This is the "master" copy of this List of Values.

Lists of Values subscribing to this List of Values:

28121 Test2 - Copy of EINS-ZWEI-DREI					
Source: Static Values					
<div style="display: flex; justify-content: space-between; align-items: center;"> <div> <input type="text" value="Search: All Text Columns"/> <input type="button" value="Go"/> </div> <div> <input type="button" value="Actions"/> </div> <div> <input type="button" value="Edit"/> <input type="button" value="Save"/> <input type="button" value="Add Row"/> <input type="button" value="Reset"/> </div> </div>					
<input type="checkbox"/>	<input type="checkbox"/>		Sequence ↑	Display	Return
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	eins	1
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2	zwei	2
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3	drei	3
1 rows selected					

Nun der Clou: Durch Klick auf die Schaltfläche „Publish“ wird die Liste in B (und beliebig viele andere, auf dieselbe Art entstandene Listen) mit den aktuellen Werten aus A aktualisiert.



Publish this List of Values?

You have requested to publish **EINS-ZWEI-DREI**. This will refresh the following Lists of Values:

- 28121 - Copy of EINS-ZWEI-DREI

Zwischenfazit:

Die Verbindung zwischen dem Original und dem „abhängigen“ Element, das durch Subscription kopiert wurde, ist eine „weiche“ Abhängigkeit. Man kann sowohl das Original als auch die Kopie unabhängig in ihren jeweiligen Anwendungen verwenden und bearbeiten. Erst, wenn entweder der Subscriber (B, individuell) oder der Publisher (A, generell für alle Subscriber) die Aktualisierung des Elements auslöst, wird das anhängige Element aktualisiert.

Einschränkungen und Besonderheiten

- Subscription funktioniert nur zwischen den Anwendungen im selben APEX-Workspace.
- Die Aktualisierung des Subscribers muss manuell ausgelöst werden; dafür ist dem Autor keine scriptbare Variante bekannt.
- Die Verbindung zwischen Publisher und Subscriber lässt sich nur lösen, indem der Subscriber gelöscht und neu erstellt wird
- Da der Subscriber eine „flache“ Kopie des Publishers darstellt (und lediglich die Information abspeichert, vom welchem Element er abstammt), wirkt sich Subscription in keinsten Weise negativ auf die Performance einer Anwendung zur Laufzeit aus.

Master-Applikation

Um die Arbeit von Anfang an möglichst effizient zu strukturieren, ist es empfehlenswert, im Workspace eine (einzige) Master-Applikation zu installieren. Nur ausgehend von dieser werden Subscriptions erlaubt. Diese Master-Applikation sollte auch möglichst nur solche Komponenten enthalten, die in anderen Anwendungen wiederverwendet werden.

APEX speichert jede Subscription als Beziehung von Komponenten, die sich in Anwendungen mit bestimmten Applikations-IDs befinden. Ist eine Subscription einmal definiert, ist es anschließend völlig legitim, die Master-Applikation zu löschen und danach durch eine andere zu ersetzen. Auf diese Art und Weise lassen sich, über mehrere APEX-Workspaces hinweg, Master-Applikationen verteilen – es muss also lediglich darauf geachtet werden, dass die aktualisierte Anwendung unter derselben App-ID installiert wird.

Aus demselben Grund ist es auch grundsätzlich nicht einmal erforderlich, die Master-Applikation in den Produktions-Workspace zu deployen. Es sei denn, man möchte genau das, um für vergleichsweise unkritische Änderungen flexibel zu bleiben (Beispiel: textuelle Anpassungen).

Sollte die Master-Applikation nicht mehr zur Verfügung stehen (was selbst bei professioneller Softwareverwaltung nicht ganz undenkbar ist), ließe sich eine „neue“ durch einfaches Zurückkopieren bestehender Komponenten-Instanzen wiederherstellen.

Anwendungsfälle

Warum sollten Sie überhaupt Subscription nutzen?

Erstens, weil Sie damit größere Kontrolle über zusammengehörige Anwendungen erlangen. Dort, wo Sie Objekte subscriben, geben Sie deklarativ die Hoheit an eine zentrale Stelle ab, von der aus die spätere Wartung aller abonnierten Elemente erleichtert wird. Oder anders ausgedrückt: Wenn Sie auf Subscription verzichten, werden sich Ihre Entwickler mehr Freiheiten herausnehmen (müssen), um eigentlich wiederkehrende Elemente jeweils individuell zu implementieren – und dabei verliert das Team schnell die Übersicht, wo sich solche Artefakte in den Anwendungen befinden bzw. „verstecken“, denn ein kopiertes (und nicht abonniertes) Element führt ja ab seiner Geburt ein völlig eigenständiges Leben.

Zweitens, um den berühmten „Wildwuchs“ einzudämmen, der gerade im Bereich scheinbarer Kleinigkeiten ansonsten überhandnehmen könnte.

Drittens, um sich nicht ständig wiederholen zu müssen, denn letztendlich macht das bequeme „Copy and Subscribe“ an vielen Stellen auch fehlerträchtige Tipparbeit überflüssig und kann dadurch wertvolle Entwicklungszeit einsparen helfen.

Anhand eines zentralen Beispiels fast aller Anwendungen soll aufgezeigt werden, dass Subscription sogar zu Dokumentationszwecken taugt: Die „Ja/Nein“-Werteliste, sozusagen der Klassiker ihrer Art. Hier müssen Sie einmal zentral festlegen, welches Werte-Paar für die Return-Werte Sie in der Datenbank grundsätzlich verwenden (1/0, Y/N, J/N, TRUE/FALSE etc.) und ob NULL-Werte ebenfalls unterstützt werden. Oftmals existieren Legacy-Anwendungen, für die andere Werte gelten – und auch diese können an zentraler Stelle durch die Aufnahme in die Master-Applikation dokumentiert werden. Außerdem hilft das enorm, das sogenannte „Wording“ und dessen „Look&Feel“ in Ihren Anwendungen zu vereinheitlichen: Bevorzugen Sie „Ja/Nein“ oder „ja/nein“? Nicht zuletzt erfährt der Workspace-Owner, an wie vielen Stellen die Entwickler die Werteliste bereits subscribed haben – oder stattdessen frei selbst erstellt ...

Ausblick auf den Vortrag

Die Präsentation zeigt anhand praxisnaher Live-Demos mit steigender Komplexität, wie schnell die Subscription in APEX-Applikationen zu Erfolgen führt und wo sie am sinnvollsten einsetzbar ist. Gerade im Zusammenspiel mit APEX Shortcuts (ein ähnlich unterrepräsentiertes Feature in den meisten APEX-Anwendungen) spielt Subscription ihre Stärken aus, was es zu demonstrieren gilt.

Kontaktadresse:

Andreas Wismann
WHEN OTHERS Inh. Andreas Wismann
Hirschstr. 10
D-41564 Kaarst

Telefon: +49 (0) 2131 - 314 9966
mobil: +49 (0) 176 - 7800 3109
E-Mail: wismann@when-others.com
Internet: <https://when-others.com>