

# Mehrsprachigkeit in APEX-Anwendungen - Pfingstfest für APEX

Stefan Röb  
BASF Business Services GmbH  
Ludwigshafen

## Schlüsselworte

Apex, APEX\_LANG API, Mehrsprachigkeit.

## Einleitung

Es kommt der Tag da steht man mit seiner APEX-Anwendung vor der Herausforderung die Applikation in mehr als einer Sprache zu betreiben. Diese Anforderung ist freilich nicht nur auf große internationale Unternehmen beschränkt, auch in kleinen und mittelständischen Firmen können Mitarbeiter unterschiedlicher Muttersprachen vorhanden sein. Dieses Thema kann man auf verschiedene Arten für APEX-Anwendungen angehen. Dabei ist jeder Lösungsansatz mit einem mehr oder weniger großen Aufwand verbunden. Der hier vorgestellte Weg zeigt eine Lösung (Best Practice), die erst seit der **APEX-Version 4.2.3** möglich ist. Mehrsprachigkeit ist in APEX seither einfach zu haben und stellt deshalb keine nennenswerte Hürde mehr dar. Die BASF APEX Standard Anwendung (BAsTA) lagert das Übersetzen als eine Funktionalität in die Anwendung aus, d.h. wir übergeben die Verantwortung an die Personen, welche die Fremdsprachen beherrschen. Dabei müssen die Personen noch nicht einmal fachliche Anwender der Applikation sein. Somit sind die Zeiten von schwerverständlichen XLIFF-Dateien, die einen speziellen Editor benötigen, endgültig vorbei. Mit Hilfe der APEX API APEX\_LANG ist es uns gelungen eine einfache, rollenbasierte Methodik zu entwickeln, die als Muster für jede APEX-Anwendung zur Mehrsprachigkeit dienen kann.

## Was muss übersetzt werden ?

Jede neue Apex Anwendung innerhalb der BASF wird mit dem BAsTA Framework erstellt. Dabei ist ein nachträgliches implementieren einer oder mehrerer Sprachen für die neue Applikation eine Baseeigenschaft. Realisiert wurde das Mehrsprachigkeitsmodul mit den in APEX zur Verfügung stehenden APIs von APEX\_LANG.

Übersetzt werden müssen alle Seiten des „**User Interface**“ (Abbildung 1) und die „**Translatable Messages**“ (Abbildung 2). Tabellendaten welche die Fachlichkeit einer Anwendung betreffen sind hierbei **nicht** zu berücksichtigen.

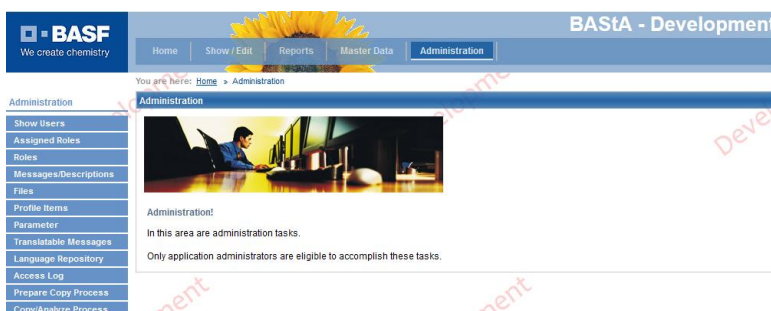


Abbildung 1: Beispiel eines User Interfaces (UI)



Abbildung 2: „Translatable Messages, left a success message, right a failure message“

Sollen allerdings auch diese Daten, also der Dateninhalt aus den Tabellen übersetzt werden, so muss dies ein fachliches Datenmodell in Form einer Übersetzungstabelle bereitstellen. Diese Anforderung ist unabhängig von Apex und ist in jeder Anwendung, gleich welcher Technologie bereitzustellen.

### Prozesse des Übersetzens

Das User Interface wird in BASTa mit dem integrierten Mehrsprachigkeitsmodul von Apex übersetzt. Dabei wird eine Art Schattenapplikation für jede Sprache erzeugt.

Die folgenden vier Schritte müssen für jede Sprache durchgeführt werden und sind in der BASTa Anwendung nur für speziell berechnigte Endanwender vorhanden.

- „Create“
  - o Anlegen der Übersetzungssprache.
- „Seed“
  - o „Aussäen“ der UI Wörter/Texte
- „Translate“
  - o Übersetzen der Wörter/Texte
- „Publish“
  - o Veröffentlichen/Bekanntgeben der übersetzten Wörter/Texte

### Create Prozess

Mit dem „Create“ Prozess wird das Mapping der „Primary Application Language“ zur „Translated Application Language“ festgelegt. Dies geschieht über einen Eintrag in die Tabelle `wwv_flow_language_map` des APEX Schemas. Ausgeführt über:

```
apex_lang.create_language_mapping (p_application_id => p_app_id
                                   ,p_language       => p_lang
                                   ,p_translation_application_id => p_trans_app_id)
```

### Seed Prozess

Nun wird die **Tabelle** `wwv_flow_translatable_text$` über die **Prozedur** `wwv_flow_lang.seed_translations` mit allen übersetzbaren Werten befüllt.

Dies wird unter Zuhilfenahme der Metatabelle `wwv_flow_translatable_cols$` durchgeführt. Ausgeführt über:

```
apex_lang.seed_translations (p_application_id => p_app_id
                             ,p_language     => lv_trans_lang);
```

### Translate Prozess

Hierbei werden die Werte der **Spalte** `translate_to_text` aus der **Tabelle** `wwv_flow_translatable_text$` geändert. Ausgeführt über:

```
apex_lang.update_translated_string (p_id      => p_id
                                    ,p_language => lv_trans_lang
                                    ,p_string  => p_to_string);
```

Der „Translate Prozess“ kann in BASTa auf zwei Arten durchgeführt werden. Entweder über eine Einzeldatensatzbearbeitung oder über einen Excel Download zur Massbearbeitung und anschließendem Upload. Beide Vorgehensweisen werden im Vortrag gezeigt.

### Publish Prozess

Die Sprache wird hierüber live geschaltet. Dies geschieht zunächst über den internen Kopiervorgang `wwv_flow_translation_util_api.flow_copy` bei dem das Language Repository mit der neuen „Translated Application ID“ angelegt wird. Dann werden die Updates des vorherigen Prozesses

„Translate Prozess“ aus der Tabelle `wwv_flow_translatable_text$` über die Prozedur `wwv_flow_translation_util_api.sync_translations` auf die jeweiligen Tabellen des Language Repositories übertragen. Als Metadaten Referenz gilt hier ebenfalls `wwv_flow_translatable_cols$`. Gestartet wird der Prozess über:

```
apex_lang.publish_application (p_application_id => p_app_id
                             ,p_language      => lv_trans_lang);
```

## Messages

Nachdem das User Interface übersetzt wurde muss dies noch für die Success und Failure Messages erfolgen (Abbildung 2). Hierzu muss die Apex Tabelle `wwv_flow_messages$` befüllt werden. Das erfolgt über den Aufruf von:

```
wwv_flow_api.create_message (p_flow_id      => p_app_id
                             ,p_name        => upper (p_message_name)
                             ,p_message_language => lower (p_target_lang)
                             ,p_message_text => p_message_text);
```

Der Parameter `P_MESSAGE_NAME` legt für eine bestimmte Sprache den Message-Identifizier fest.

Das Update der Message erfolgt über die folgende API:

```
apex_lang.update_message (p_id      => lv_trans_id
                          ,p_message_text => p_upd_message_text);
```

Zu guter Letzt noch das Löschen einer Message: Dies erfolgt über die gleiche API wie das Anlegen einer Message mit dem feinen Unterschied, dass zuvor der globale API Mode auf „REMOVE“ gesetzt werden muss.

```
wwv_flow_api.g_mode := 'REMOVE';
wwv_flow_api.create_message (p_id => lv_trans_id);
```

Ein zuvor kreierter Message-Identifizier (hier im Beispiel „CREATE\_LANG\_MAP\_SUCCESS“) wird im PL/SQL Code über die Funktion `apex_lang.message` aufgerufen. Es können bis zu 10 verschiedene Parameter dem Identifizier mitgegeben werden (hier im Beispiel werden „lv\_lang“ und „p\_trans\_app\_id“ mitgegeben). Diese Parameter flexibilisieren die zu erzeugende Message.

```
apex_lang.message ('CREATE_LANG_MAP_SUCCESS'
                  ,lv_lang
                  ,p_trans_app_id);
```

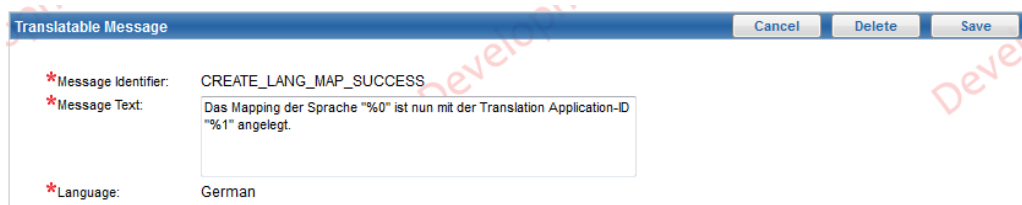



Abbildung 3: DML für den Message-Identifizier „CREATE\_LANG\_MAP\_SUCCESS“ mit zwei Parametern (%0 und %1)

## Umschalten der Sprache

Für jeden neuen Anwender der BASfA Anwendung wird sein Sprachenprofil in einer Tabelle festgehalten. Das Application Item FSP\_LANGUAGE\_PREFERENCE wird dabei über einen Post Authentication Process befüllt.

In den Globalisierungsparametern unter “Shared Components“ muss die „Application Derived Language“ dabei auf „Item Preference ...“ eingestellt werden.

Application Language Derived From  

Nachdem alle Sprachkomponenten übersetzt wurden kann jeder Anwender sein eigenes Sprachenprofil ändern und in die zur Verfügung stehende Sprache umschalten.

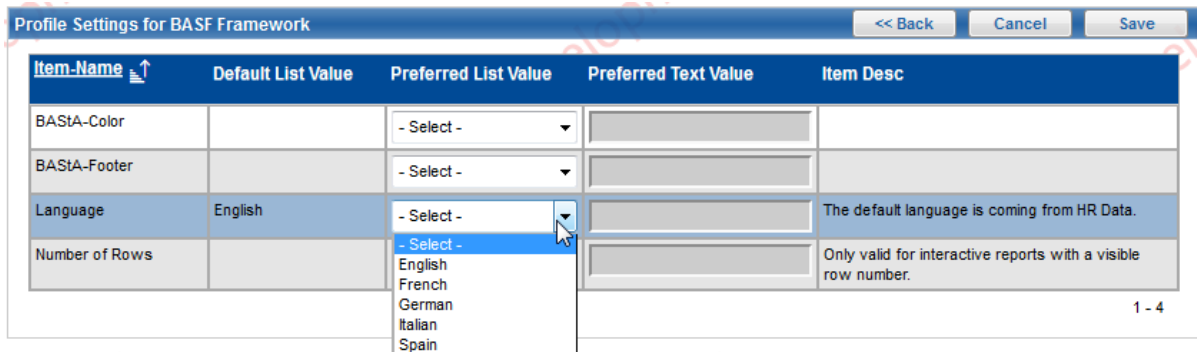


Abbildung 4: individuelle Sprachprofileinstellung in BASfA

## Fazit.

Das BASF Framework BASfA wurde sowohl unter Apex 4.2.5 als auch Apex 5.0.1 getestet. Seit der Version 4.2.3 sind von den API Methodiken keine Neuerungen dazugekommen.

Das Mehrsprachigkeitsmodul von Apex ist mit seinen bestehenden Bibliotheken angenehm in eine Applikation einzubinden um daraus ein eigenes Framework aufzubauen.

## Kontaktadresse:

Stefan Röß  
BASF Business Services GmbH  
Pfalzgrafenstraße 1  
D-67061 Ludwigshafen

Telefon: +49 (0) 621 60-49176  
E-Mail: Stefan.Roess@basf.com  
Internet: www.information-services.basf.com/