

Dynaflex im praktischen Einsatz

flexible Flexforms für's TYPO3 Backend

Einführung

- Extension entwickelt von Thomas Hempel
- Extkey: `dynaflex`
- Tutorial Extension: `dynaflex_tut`

Warum DynaFlex ?

- Flexible Darstellung von Formularfeldern nach Eingabewerten
- dynamisches Hinzufügen von Eingabefeldern und Sheets anhand bestimmter Eingabewerte

Ausgangspunkt

- statische Flexforms
- Formular wird per XML beschrieben
- Formularefelder werden per XML in der Datenbank gespeichert

Einbindung von DynaFlex

- Hook in `ext_localconf.php` der Extension ansprechen
- Muss für jedes Flexform angelegt werden, was durch DynaFlex dynamisiert werden soll

```
$GLOBALS['T3_VAR']['ext']  
['dynaflex']['<tabelle>'][] =  
'EXT:dynaflex_test/  
class.tx_dynaflextest_pichange.php:  
tx_dynaflextest_pichange';
```

**Verweis auf die Klasse, die das
DCA beinhaltet**

DCA Aufbau

- SCA (Single Configuration Array)
- MA (Modification Array)
- Abgelegt als Instanzvariable einer separaten Klasse

```
class tx_dynaflex_dfconfig {  
    var $DCA = array (  
        ....  
        ....  
    )  
}
```

Pfade in Dynaflex

- tt_content/columns/pi_flexform/config/ds/flash_slideshow_pi1
- <tabelle>/columns/<spaltenname>/config/...

SCA

- Beinhaltet alle Modifications
- Ein SCA kann mehrere Modifications enthalten
- Gibt Pfad an, wo die Datenbearbeitung stattfinden soll

MA

- Enthält Beschreibung der Modifications
- method: add / move / remove / usermethod
- type:
- condition: bedingte ausführung
- source: Datenquelle

Source

- Mögliche Quellen:
- Felder im Pagedatensatz
- Felder aus der Datenbank
- TS - Select Abfragen
- XML Flexform Felder auch möglich!
- Beschreibung wie die Daten ermittelt werden sollen

Field / Sheet Config

- Konfiguration der Felder analog zum FlexForm / TCA
- z.B.:

```
array(  
    'label' => 'Y Position',  
    'name' => 'ypos_###SINDEX###',  
    'config' => array(  
        'type' => 'input',  
        'size' => 5  
    )  
)
```

Marker

- ###SINDEXT### => Sheetindex
- ###FINDEXT### => Feldindex
- Durchlaufend nummeriert

Condition

- Alle Möglichkeiten der TYPOScript if Abfrage stehen zur Verfügung
- Legt fest wann eine Transformation ausgeführt werden soll
- Datenquelle kann über Source konfiguriert werden

Field / Sheet Config

- Hinzufügen mehrerer Felder z.B. über einen numerischen Wert aus einem Eingabefeld möglich
- Für Feld / Sheetnamen können Marker verwendet werden

Zugriff über pibase

```
$this->getFFvalue($flexData, '<fieldname>', sheetname);
```

- Feldname wird dynamisch durch Marker erzeugt
- Möglicherweise wird kein Wert zurückgegeben

```
$DCA (  
  path => 'pfad/zum/feld'  
  modifications = array(  
    array(  
      method => 'add/remove/move',  
      type => 'sheets/fields/field',  
      source => 'sheets/fields/field',  
      source_type => 'sheets/fields/field',  
      source_config => array (  
        ...  
      )  
      condition => array(  
      )  
      sheet_config => array(  
        ... TCA Config  
      )  
  
      field_config => array (  
        ... TCA Config  
      )  
    )  
  )  
);
```

Getting Started

1. Dynaflex aus dem TER installieren
2. Klasse mit DCA Variable erstellen
3. DynaFlex per Hooks einbinden

Hints

```
var $cleanUpField = 'pi_flexform';
```

```
var $rowChecks = array (  
    'list_type' => 'dynaflex_tut_pi1',  
    'CType' => 'list'  
);
```