Intelligent Webforms

*Interact with users using webforms*
Program

- Introduction
- Why a new extension for creating webforms?
- The FormMaker framework concept
- The FormMaker workflow
- What can I do with the FormMaker
- What has been changed since T3CON06
- Demo 'FormMaker in action'
- Sneak preview developments
- Questions and Answers
Introduction

- **My name:** Rogier Hosman
  Consultant & one of 3 founders

- **Drecomm Internet Intelligence BV**
  - Founded in 1999
  - 30+ full time employees
  - 18 Typo3 developers
  - Using TYPO3 since version 3.3
Why a new extension for creating webforms

- Multiple page webforms
- Using TYPO3 page & content operations:
  - Setting user rights within form objects
  - Start /Stop dates on fields
  - Copy /Paste elements /pages
  - Shortcut to pages /elements
  - Use insert record functions
  - Using back-end user rights for form objects
  - Including TYPO3 content objects in forms, like images, movies etc
- Interaction with 3rd party software
- Reusable form components
Concepts of the FormMaker framework

- Form field repository
- Javascript repository
- PHP scripts repository

Form
- Form group
  - Form field 1
  - Form field 2

<?XML?>

- Mysql
- Mysql

RE USABILITY + PREDEFINED FORMGROUPS + FLEXIBLE OUTPUT
The workflow

1. Configure & install
   - Create field repository
   - Include needed java scripts
   - Include needed php scripts

2. Group fields
   - Address group
     - Postal address
     - Postal code
     - Street + nr.

3. Output definition
   - Create output type(s)
   - Output generation wizzard
   - Lay-out output (CSS wizzard)

Operations
- Validations
- Calculations
- Data exchange
What has been changed since 2006

▸ Import & export forms, fields and repositories to other Typo3 installations
▸ Automated summery generator
▸ Reporting tool
▸ Soap exchange possiblities
So what can I do with FormMaker

- Centralised control of all webforms on your site
- Re-use scripts, formobjects and form structures within one or more Typo3 installations
- Create user dependant output in any form you like
- Easily create forms using predefined structures, as simple as shortcutting to a page
- Make calculations based on user input
- Create a variety of output formats which can be distributed in various ways
Demo

- Logics of the library
  - Formfields
  - Java scripts
  - PHP scripts
- Build formgroups
  - Multi level structure
- Enabling hidegroup functionality
- Adding javascripts (datepicker), validation
- Reporting
FormMaker developments

▸ Renewed back-end user interface
  ▸ Front-end editing concept
  ▸ Reduce the number of clicks needed to build forms
  ▸ Improve access to all functions & features
By clicking on preview, a popup will be opened.

Title:
Form with template

Temporary storage type
- cookies
- database

Number of columns:

- Email output
- PDF output
- External storage

By clicking next, the form settings will be saved and the first page is automatically made. Then you go to the page settings of that page.
Any questions?
Other of our interesting developments...

- **Content Creation templates**
  - Create input templates based on typo3 content elements

- **Knowledgebase extension**
  - Taxonomy based tagging
  - Dynamic content relations based on tagging

- **Front-End language file editor**
  - Modify language markers at the front-end, no more search for labels
### Connection wizard

**Path:**

<table>
<thead>
<tr>
<th>Name</th>
<th>ANVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>request</td>
</tr>
<tr>
<td>Protocol</td>
<td>HTTPS</td>
</tr>
<tr>
<td>Method</td>
<td>POST</td>
</tr>
<tr>
<td>Address</td>
<td>test.soap.server.com</td>
</tr>
</tbody>
</table>

**Reload the tree from server**
### Connection wizard

**Path:**

- **ANVA LIVE**

**MAPPING:**

- **Name:** mapping
- **Type:** XML
- **PHP:** ANVA get

---

**Reload the tree from server**
INSTRUCTIONS:
Here you can add, modify, hide or remove PHP functions using plain PHP coding that can be used with expressions throughout the forms in this TYPO3 installation.

PHP #248:

Name: ANVA get

Description:
Converts ANVA response XML to array and stores it in current session

Parameters: values,index

The function name and parameter list do not have to be included in the PHP code.

```
$rs = Array();
$id = $index['RELATIE'][0];

$id = Array(
    'RELATIE' => 0,
    'POLIS' => -1,
    'POLISDEKKING' => -1,
    'PAKKET' => -1,
    'PAKKETPOLIS' => -1,
    'PAKKETPOLISDEKKING' => -1
);

$category = Array('RELATIE');
while (count($index) > 0 &&
    $values[$i]['tag'] != 'RELATIE' ||
    ($values[$i]['type'] != 'close' &&
     $values[$i]['type'] != 'complete'
    )
)
{
    switch ($values[$i]['type']) {
        case 'open':
            switch ($values[$i]['tag']) {
                case 'PAKKETTER':
                    $category[] = 'PAKKET';
                    break;
                case 'POLISSEN':
                    switch (count($category)-1) {
                        case 'RELATIE':
                            $category[] = 'POLIS';
                    }
            }
        
```
FILE EDITOR:

```xml
<PAKKETEN>
  <PAKKET>
    <PK/>
    <L NR=""/>
    <POLISSEN>
      <PK/>
      <L NR=""/>
      <DEKKING>
        <PK/>
        <L NR=""/>
      </DEKKING>
    </POLISSEN>
    <WHERE>
      <L NR="10001" EQ="N210"/>
      <L NR="10141" EQ="1000"/>
      <L NR="10142" GE="1250"/>
      <L NR="10142" LE="1254"/>
    </WHERE>
    </POLISSEN>
  </PAKKET>
  <WHERE>
    <L NR="61000" EQ="30"/>
  </WHERE>
</PAKKETEN>
</RELATIE>
</WHERE>

<!--WHERE1-->""
<!--WHERE1-->

<!--WHERE2-->""
<!--WHERE2-->
```
