

TYPO3: Extension for Google Map with flexform

by *Bernd Warken* at *netcos AG*

0. Introduction	3
<i>0.1 Preface</i>	3
<i>0.2 License</i>	3
1. Code for Google Map	4
<i>1.1 Google Apis</i>	4
<i>1.2 Input in the TypoScript template</i>	4
2. Kickstarter	5
<i>2.1 Creation of the Extension</i>	5
3. Adjust the Source Code	6
<i>3.1 Changing Files</i>	6
4. Usage of the Extension	10
<i>4.1 Configuration</i>	10
<i>4.2 Errors</i>	10

0. Introduction

0.1 Preface

This document describes the background in the creation of the TYPO3 extension **netcos_googlemap** for the usage of one or more Google maps within TYPO3. This is an official extension, it is available in typo3.org at http://typo3.org/extensions/repository/view/netcos_googlemap.

The values for latitude, longitude, zoom level, and map type can be configured within the plugin, such that even several maps can be generated.

0.2 License

This document is part of the TYPO3 project on **netcos_googlemap**. The actual version of this document **1.3.3** of 2 April, 2008. The author is **Bernd Warken** at [netcos AG](#).

All elements of the project (this document and the source code) are put under the license **GPL (GNU General Public License)** version **3**. The original text of this license is available at <http://www.gnu.org/licenses/gpl.html>.

Copyright 2007,2008 Bernd Warken

1. Code for Google Map

To use a Google map on a domain, it is necessary to get a Google map code for this domain from Google.

1.1 Google Apis

The Google map code is generated at <http://google.com/apis/maps>. There you need to provide a URL for the domain to create the code. For each domain name, a suitable code with a length of about 80 characters is generated. This code should be copied and remembered. We store this code in the following TypoScript constant,

```
google_map_key = ABQIAAAARokCoxWjWuwH4eEaCOyBkBQuBzwEBiC_X9c-oZXEt-cQgHLphTnVyn
```

But this code is not yet complete, it has been abbreviated to the line length.

If a TYPO3 site is just run on *localhost* without a domain name it is not necessary to create a Google map code. Each arbitrary value suits for *localhost*.

1.2 Input in the TypoScript Template

In TypoScript, the values of the HTML header can be provided by the parameter **headerData**. It is necessary to run a JavaScript program within the header. Two TypoScript commands perform this task,

```
page.headerData.900 = TEXT
page.headerData.900.value (
    <script src="http://maps.google.com/maps?file=api&v=2&key={$google_map_key}"
        type="text/javascript">
    </script>
)
```

page relates to the object assigned by **PAGE**, having **typeNum 0**. Maybe it is called differently with you. The value **900** is arbitrarily chosen. You should just take care that this value is not used by **headerData** at a different place. **google_map_key** is the constant storing the Google code.

2. Kickstarter

The extension *Kickstarter* is the TYPO3 tool to create extensions.

2.1 Creation of the Extension

The extension *Kickstarter* must be installed first. By that a new entry *Make new extension* is created in the menu of the *Extension Manager*. We choose this entry.

In the arising form, a name in the field *Enter extension key*: should be provided first, in this case it is **netcos_googlemap**. Affirm with the button *Update...*

In the following, the option list *KICKSTARTER WIZARD* is dealt with in the form. First the plus character of *General info* is pushed. Therein the *Title Netcos googlemap* is created. As *Category*, the *Frontend Plugins* is chosen. *State* is *Beta*. For *Dependencies*, the value **cms** is submitted. *Author* is added. Push the button *Update...*

Next choose the plus character of *Setup languages* and the language *German* therein. Affirmation by *Update...* The default language of an extension is always English, German is an additional language.

Now click the plus character of *Frontend Plugins*. Provide *title: Netcos googlemap* in both languages. Push *Add icon to 'New Content Element' wizard*:. Update.

Next we add to the existing table *tt_content* the datasets that will be available in the plugin later on. We choose the plus character of *Extend existing Tables*. Under *Which table* choose *Content (tt_content)*. Now we add the data names in *NEW FIELD*: .

- First the *Field name* **width** with *Field title*: **Width** (in English) and **Breite** (in German), as *Field type*: **String input**. Update. In the arising fields, we add **4** to *Field width* and *Max characters*. *Required* is not applied.
- In the same way, another dataset is added with the following values: **height**, **Height**, **Höhe**, **String input**, **4**.
- A further dataset with the values: **latitude**, **Latitude**, **Breitengrad (Latitude)**, **String input**, **10**, **15**.
- And: **longitude**, **Longitude**, **Längengrad (Longitude)**, **String input**, **10**, **15**.
- As well as: **zoom**, **Zoom Level**, **Zoomlevel**, **String input**, **5**, **5**.
- Finally: **maptype**, **Map Type**, **Kartentyp**, **String input**, **9**, **9**.
- Next we push the plus character of *Frontend Plugins*. *title: Netcos Google Map* and **Netcos Google-Map** (for German). Push *Add icon to 'New Content Element' wizard*:. Update.

The input is now finished. So choose the field button *View result*. There click on *WRITE*. The extension is then stored to **typo3conf/ext/netcos_googlemap/**.

The extension ist now complete and could be installed on *Install Extensions*. But still no map is shown in the frontend. Only some default text generated by the *Kickstarter* wizard is displayed. So it is necessary to change the source code in the generated files.

3. Adjust the Source Code

As soon as some file in the source code (in `typo3conf/ext/netcos_googlemap/`) is changed the *Kickstarter* may not be applied any more, because it overwrites all files with default values.

3.1 Changing Files

It seems as if the files `wizard_form.*` in the source code in `typo3conf/ext/netcos_googlemap/doc/` may be deleted.

You can change a file in the source code by the *Ext Manager*. Choose the menu entry *Install extensions*. As the generated extension is now installed in `typo3conf/ext` it is displayed there. Click the extension name to get to its details. In the arising menu choose *Edit files*. This displays all source code files in the extension. By clicking on *Edit file* right to the file name, an editor for change is raised with this file.

We want to change to *flexform*. So the following files must be adapted, `ext_tables.php`, `locallang_db.xml` and `pi1/class.tx_mygooglemap_pi1.php`; the file `flexform_ds_pi1.xml` is created as new file. Documentation for *flexform* is available at http://wiki.typo3.org/index.php/Extension_Development%2C_using_Flexforms.

- In `ext_tables.php`, the element `tx_netcosgooglemap_` must be deleted within the array `$TCA['tt_content']['types']['list']['subtypes_addlist']` and `tt_content` must be replaced by `flexform_pi1`. Moreover the following lines must be added:

```
// include flexform for this extension
$TCA['tt_content']['types']['list']['subtypes_addlist']
[$_EXTKEY.'_pi1']='pi_flexform';
// add flexform parameters
t3lib_extMgm::addPiFlexFormValue($_EXTKEY.'_pi1', 'FILE:EXT:'.
$_EXTKEY.'/flexform_ds_pi1.xml');
```

- In `locallang_db.xml`, only the lines containing `tx_netcosgooglemap_` are changed. Again `tx_netcosgooglemap_` is deleted and `tt_content` is replaced by `flexform_pi1`. The lines with the index `tt_content.list_type_pi1` are unchanged. When necessary name for the *default sheet*, the header during the input, can be given. There the following line is added in both language regions, eventually with a different content:

```
<label index="flexform_pi1.sheet_default">Default Sheet</label>
```

- The main file for change is the file `pi1/class.tx_netcosgooglemap_pi1.php`. In it, the region

```
function main($content,$conf) {
    ...
}
```

is replaced by the following code

```

function main($content,$conf) {
    $this->conf=$conf;
    $this->pi_setPiVarDefaults();
    $this->pi_loadLL();
    $this->pi_initPIflexForm();
    $this->lConf = array();
    $piFlexForm = $this->cObj->data['pi_flexform'];
    foreach ( $piFlexForm['data'] as $sheet => $data )
        foreach ( $data as $lang => $value )
            foreach ( $value as $key => $val ) {
                $this->lConf[$key] = $this->pi_getFFvalue($piFlexForm, $key, $sheet);
                //          t3lib_div::debug($this->lConf[$key], $key);
            }
    $width = strval(intval($this->lConf['width']));
    if ($width == 0) { $width = 400; }
    $height = strval(intval($this->lConf['height']));
    if ($height == 0) { $height = 400; }
    $latitude = $this->lConf['latitude'];
    $latitude = str_replace(',', '.', $latitude);
    $latitude = strval(floatval($latitude));
    $longitude = $this->lConf['longitude'];
    $longitude = str_replace(',', '.', $longitude);
    $longitude = strval(floatval($longitude));
    if ($latitude == 0 || $longitude == 0) {
        $latitude = 48.1456;
        $longitude = 11.6156;
    }
    $zoomlevel = strval(intval($this->lConf['zoom']));
    $type = strtoupper($this->lConf['type']);
    switch($type) {
        case 'NORMAL':
        case 'SATELLITE':
        case 'HYBRID':
            break;
        default:
            $type = 'HYBRID';
    }
    $type = 'G_'. $type. '_MAP';
    $content="
    <div id='map' " " .
        "style='width: ".$width."px; " " .
        "height: ".$height."px"></div>
    <script type='text/javascript'>
        //<![CDATA[
            if (GBrowserIsCompatible()) {
                var map = new GMap2(document.getElementById('map'));
                map.addControl(new GLargeMapControl());
                map.addControl(new GMapTypeControl());
                map.setCenter(new GLatLng(".$latitude.", ".$longitude."), ".
$zoomlevel.", ".$type.");
            }
        //]]>
    </script>
    ";
    return $this->pi_wrapInBaseClass($content);
}

```

The JavaScript code is based on the TYPO3 DVD *Einstieg in TYPO3 4.0* by *Thomas Kötter*. In the *function main* some lines are added. Moreover all `tx_` elements are replaced `$this->lConf['...']` within *content*.

- The file *flexform_ds_pi1.xml* is newly created and looks like

```

<T3DataStructure>
  <sheets>
    <sDEF>
      <ROOT>
        <TCEforms>
          <sheetTitle>LLL:EXT:netcos_googlemap/locallang_db.xml:flexform_pil.sheet_default</sheetTitle>
        </TCEforms>
        <type>array</type>
      <el>

        <width>
          <TCEforms>
            <label>LLL:EXT:netcos_googlemap/locallang_db.xml:flexform_pil.width</label>
            <config>
              <type>input</type>
              <size>4</size>
            </config>
          </TCEforms>
        </width>

        <height>
          <TCEforms>
            <label>LLL:EXT:netcos_googlemap/locallang_db.xml:flexform_pil.height</label>
            <config>
              <type>input</type>
              <size>4</size>
            </config>
          </TCEforms>
        </height>

        <latitude>
          <TCEforms>
            <label>LLL:EXT:netcos_googlemap/locallang_db.xml:flexform_pil.latitude</label>
            <config>
              <type>input</type>
              <size>10</size>
            </config>
          </TCEforms>
        </latitude>

        <longitude>
          <TCEforms>
            <label>LLL:EXT:netcos_googlemap/locallang_db.xml:flexform_pil.longitude</label>
            <config>
              <type>input</type>
              <size>4</size>
            </config>
          </TCEforms>
        </longitude>

        <zoom>
          <TCEforms>
            <label>LLL:EXT:netcos_googlemap/locallang_db.xml:flexform_pil.zoom</label>
            <config>
              <type>input</type>
              <size>2</size>
            </config>
          </TCEforms>
        </zoom>
      </el>
    </sDEF>
  </sheets>
</T3DataStructure>

```

```
        </TCEforms>
    </zoom>

    <type>
        <TCEforms>
            <label>LLL:EXT:netcos_googlemap/locallang_db.xml:flexform_pi1.typ
e</label>
            <config>
                <type>input</type>
                <size>10</size>
            </config>
        </TCEforms>
    </type>

    </el>
</ROOT>
</sDEF>
</sheets>
</T3DataStructure>
```

In this code, some lines are too long. Please take care that these lines are not broken.

4. Usage of the Extension

The extension can be used by clicking the button *Create new element* on a page in the TYPO3 module *Page*; in the resulting form you choose the plugin of the extension.

4.1 Configuration

In the plugin, 6 fields can be set: the width and height of the map image, the latitude and longitude for the position of the center of the map, the zoom level for the map, and the map type. These fields can be set freely. The documentation of the extension at *doc/manual.sxw* provides more details.

4.2 Errors

Now the map should be displayed. If not it is mostly your fault, not an error of Google. Possible errors include:

- wrong Google map code
- the float numbers of the degrees might be wrong
- the zoom level is an integer between 0 and 19
- the map type must be chosen from a set of 3 fixed values