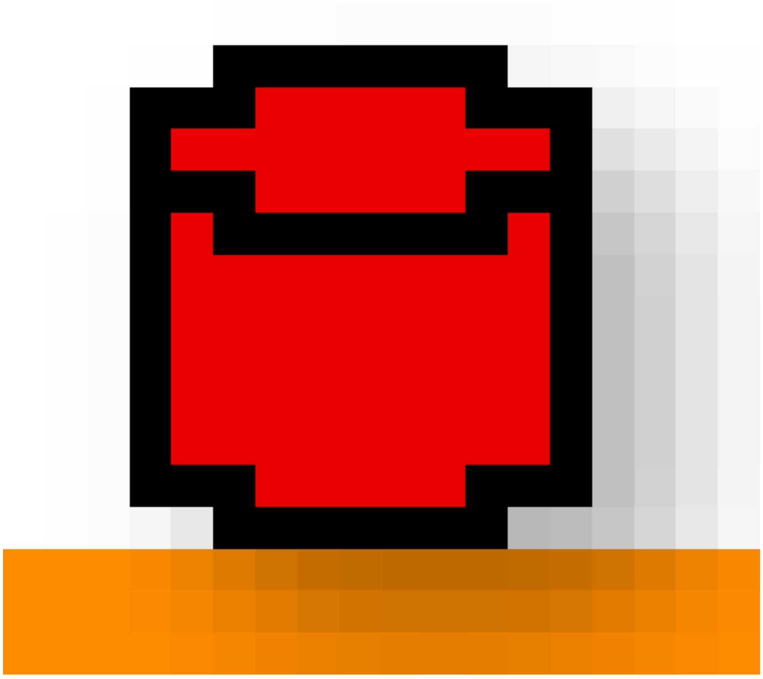
SQL 2 TYPO3

Import data from SQL tables to TYPO3 tables. TypoScript enables the controlling of sources and destinations and of wrapping values.



Version: 0.0.1, 2010-01-29

Extension Key: sql2typo

Language: en

Keywords: forAdmins, forDevelopers, forBeginners, forAdvanced, sql, converting, import, database

Table of Contents

Screen Shots

Backend

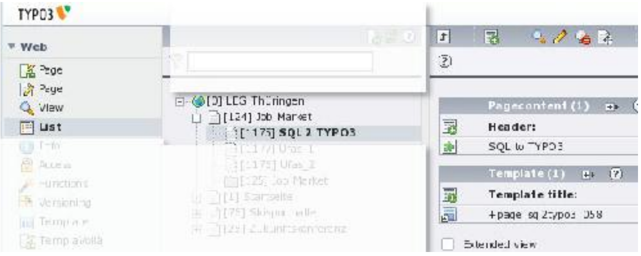


Illustration 1: Page with plugin and template

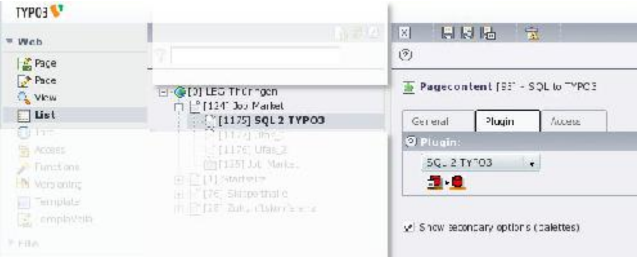
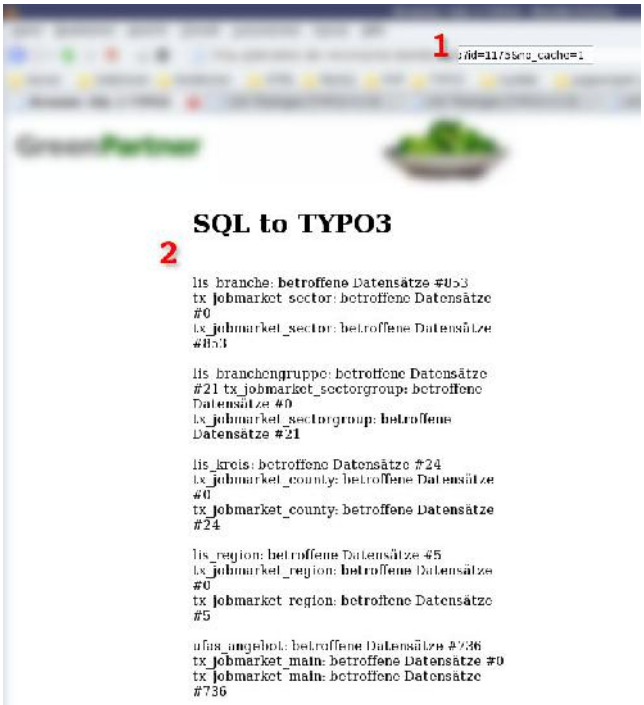


Illustration 2: SQL 2 TYPO3 plugin

Frontend



SQL to TYPO3

lis_branche: betroffene Datensätze #853
tx_jobmarket_sector: betroffene Datensätze #0
tx_jobmarket_sector: betroffene Datensätze #853

lis_branchengruppe: betroffene Datensätze #21 tx_jobmarket_sectorgroup: betroffene Datensätze #0
tx_jobmarket_sectorgroup: betroffene Datensätze #21

lis_kreis: betroffene Datensätze #24
tx_jobmarket_county: betroffene Datensätze #0
tx_jobmarket_county: betroffene Datensätze #24

lis_region: betroffene Datensätze #5
tx_jobmarket_region: betroffene Datensätze #0
tx_jobmarket_region: betroffene Datensätze #5

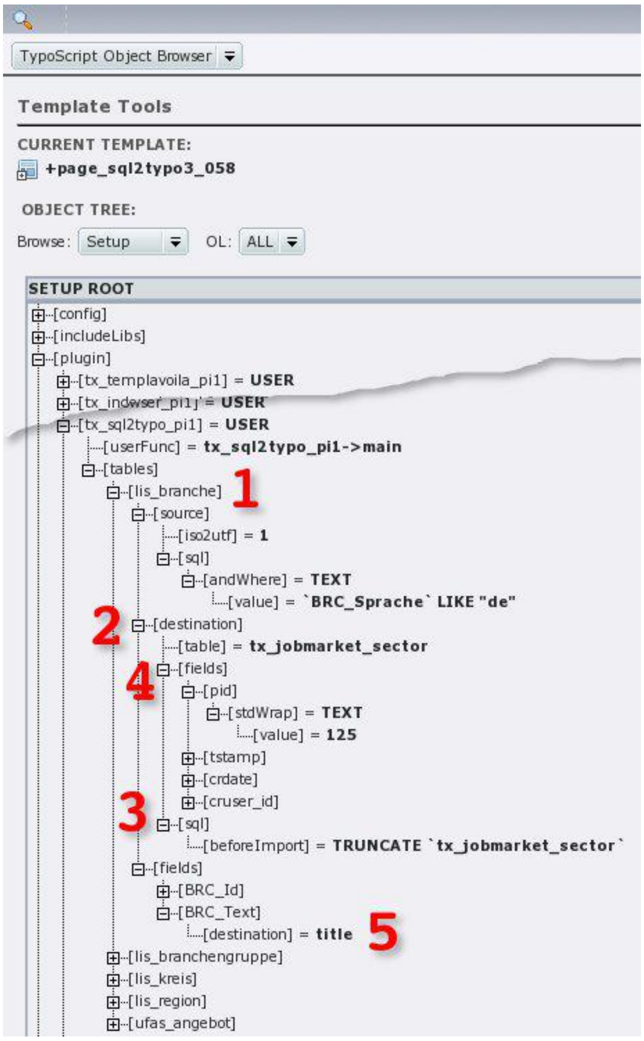
ufas_angebot: betroffene Datensätze #736
tx_jobmarket_main: betroffene Datensätze #0
tx_jobmarket_main: betroffene Datensätze #736

Illustration 3: SQL 2 TYPO3 report

Controlling by TypoScript

Short Description

1. Source table is lis_branche
2. Destination table is tx_jobmarket_sector
3. The table tx_jobmarket_sector will be truncated before the import.
4. The destination got some fixed values like pid or tstamp.
5. Source filed BRC_Text will import to destination field title.



CURRENT TEMPLATE:
+page_sql2typo3_058

OBJECT TREE:
Browse: Setup OL: ALL

```
SETUP ROOT
[config]
[includeLibs]
[plugin]
    [tx_templavoila_pi1] = USER
    [tx_indexer_pi1] = USER
    [tx_sql2typo_pi1] = USER
        [userFunc] = tx_sql2typo_pi1->main
        [tables]
            [lis_branche]
                [source]
                    [iso2utf] = 1
                    [sql]
                        [andWhere] = TEXT
                            [value] = `BRC_Sprache` LIKE "de"
                [destination]
                    [table] = tx_jobmarket_sector
                    [fields]
                        [pid]
                            [stdWrap] = TEXT
                                [value] = 125
                        [tstamp]
                        [crdate]
                        [cruser_id]
                    [sql]
                        [beforeImport] = TRUNCATE `tx_jobmarket_sector`
                [fields]
                    [BRC_Id]
                    [BRC_Text]
                        [destination] = title
            [lis_branchengruppe]
            [lis_kreis]
            [lis_region]
            [ufas_angebot]
```

Illustration 4: SQL 2 TYPO3 is controlled by TypoScript

Introduction

What does it do?

- SQL 2 TYPO3 is a benefit, if data are maintained in a third party database but should published in the world wide web with TYPO3.
- It is a benefit, if you have to import this data at regular intervals like monthly, weekly or daily.
- The extension is able to wrap any value with the TYPO3 stdWrap property and some PHP methods while the import.
- It projects 1:1-relations between tables.
- It is a frontend plugin. You are starting the import by visiting the page with the plugin in your TYPO3 frontend.
- SQL 2 TYPO3 is controlled by TypoScript.

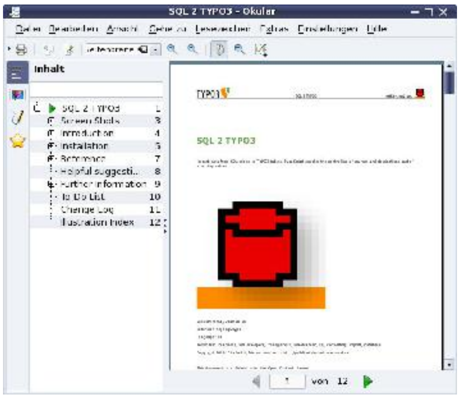


Illustration 5: Manual in PDF format

Manual in PDF

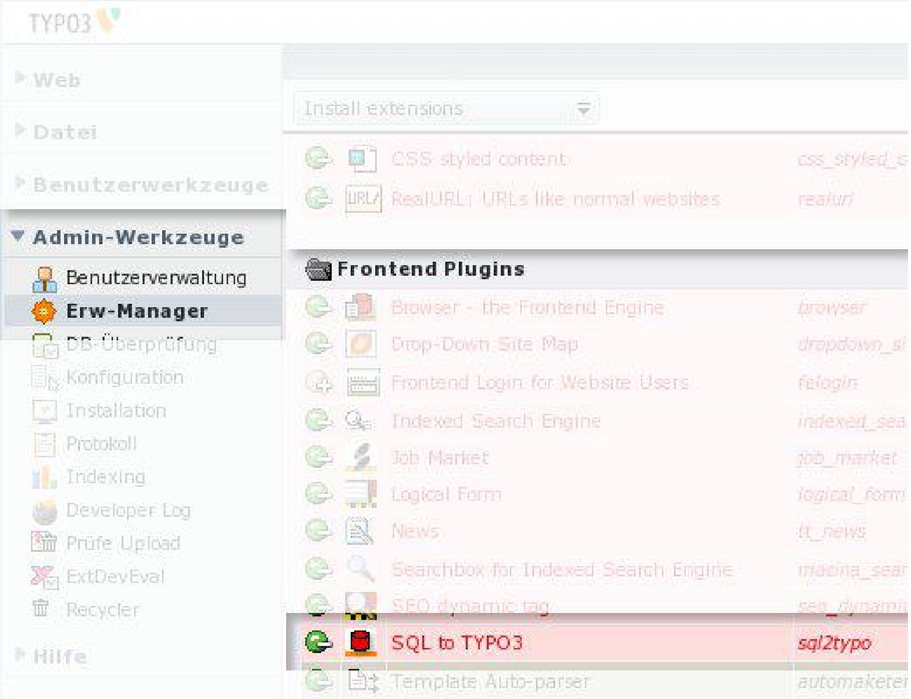
You find this manual as PDF file at

- doc/manual.pdf
- http://typo3.org/extensions/repository/view/sql2typo3/current/info/?tx_terfe_pi1%5BdownloadFile%5D=doc%252Fmanual.pdf

Installation

Extension Manager

- Install this extension (sql2typo).



CSS styled content	css_styled_c
RealURL: URLs like normal websites	realurl
Browser - the Frontend Engine	browser
Drop-Down Site Map	dropdown_si
Frontend Login for Website Users	felogin
Indexed Search Engine	indexed_sea
Job Market	job_market
Logical Form	logical_form
News	tt_news
Searchbox for Indexed Search Engine	macina_sear
SEO dynamic tag	seo_dynamic
SQL to TYPO3	sql2typo
Template Auto-parser	automaketer

Illustration 6: Extension Manager with the loaded Extension SQL 2 TYPO3 (sql2typo)

Page Tree

- Insert the page "SQL 2 TYPO3" (in our example below it has the ID 1175).

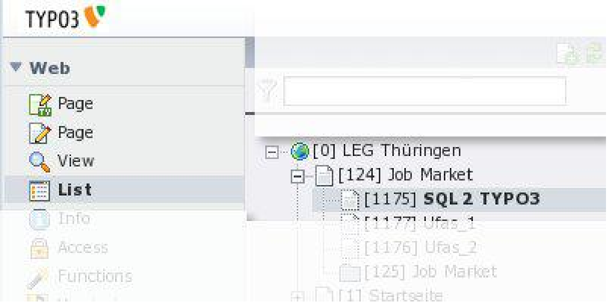
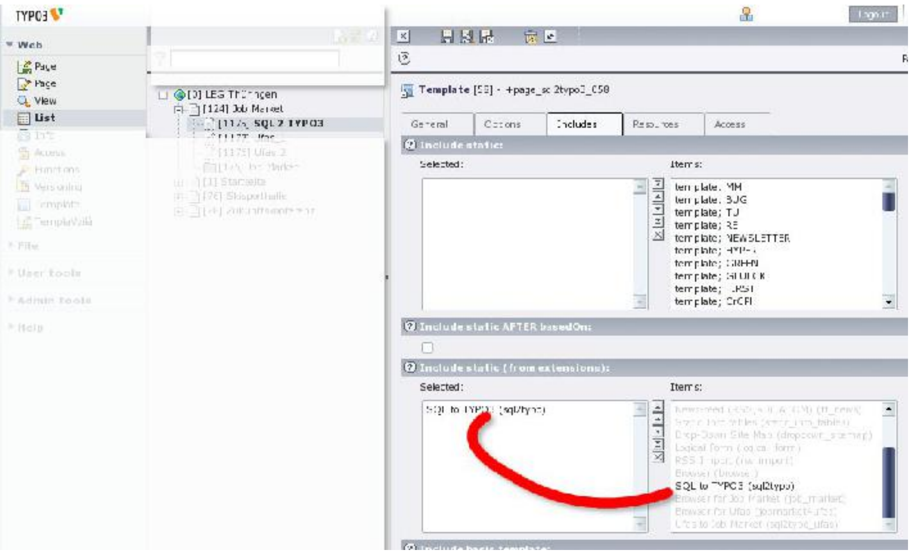


Illustration 7: Page

Extension Template

- Add an extension template to your page "SQL 2 TYPO3" (1175)
- We recommend to use this syntax for the name of the template:
  - +type_page_templateID
  - We call it: +ext_sql2typo3_038
- Include the static templates from extensions:
  - SQL 2 TYPO3 (sql2typo)



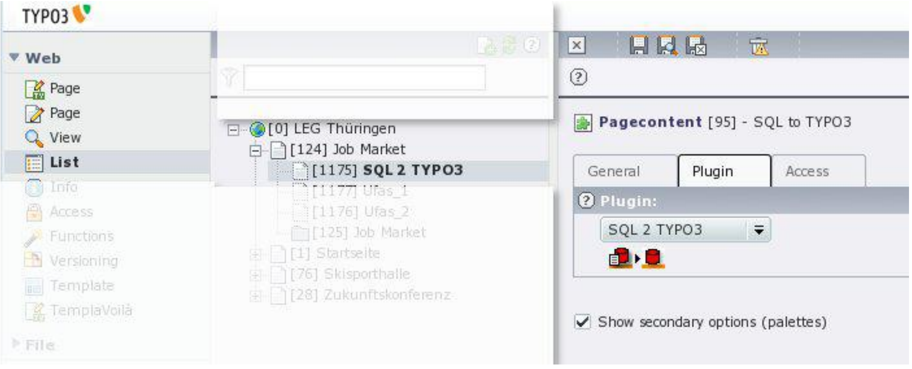
Template [38] - +page_sc 2typo3_038

Selected: SQL to TYPO3 (sql2typo)

Illustration 8: The extension template includes the static template SQL 2 TYPO3 (sql2typo)

Plugin

- Insert the plugin SQL 2 TYPO3



Pagecontent [95] - SQL to TYPO3

Plugin: SQL 2 TYPO3

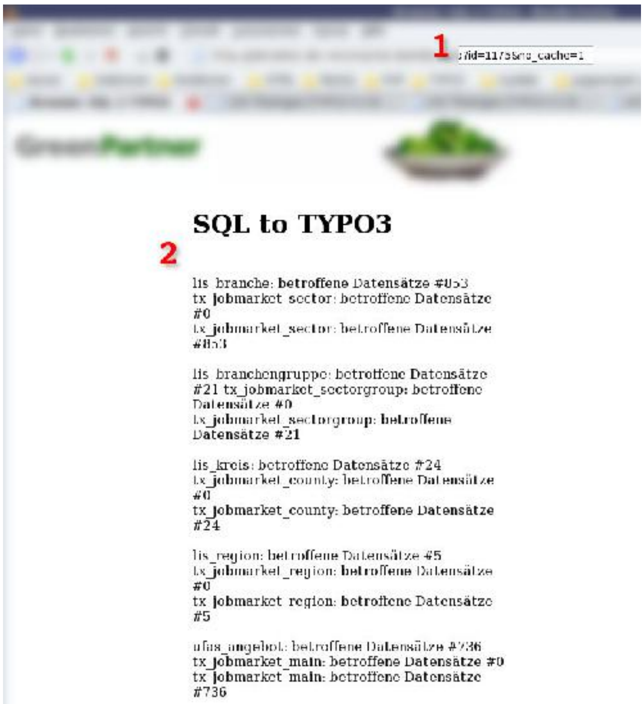
☑ Show secondary options (palettes)

Illustration 9: Plugin SQL 2 TYPO3

Workflow

1. Installation
2. Import Source Tables (see section Requirements below)
3. Configuring TypoScript (see section References below)
4. Visit the SQL 2 TYPO3 page in the frontend:
   Data will import

Frontend

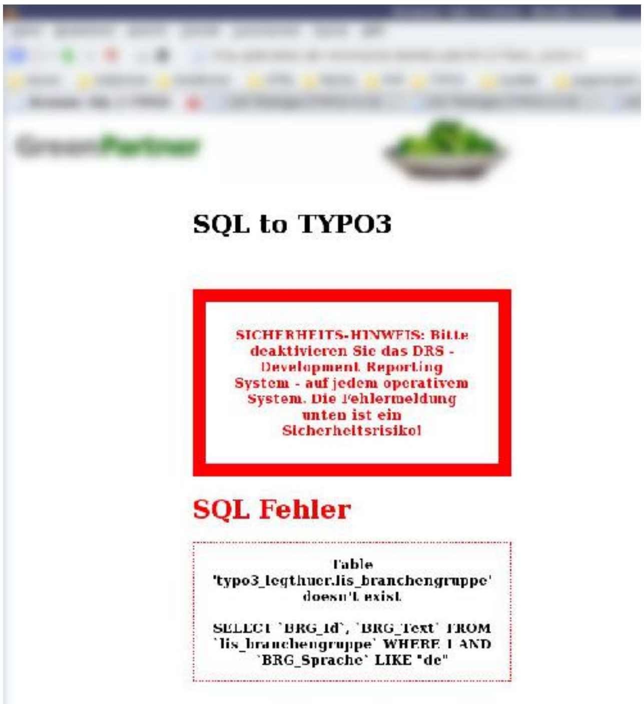


```
?id=1175&no_cache=1
```

SQL to TYPO3

```
lis_branche: betroffene Datensätze #853
tx_jobmarket_sector: betroffene Datensätze
#0
tx_jobmarket_sector: betroffene Datensätze
#853

lis_branchengruppe: betroffene Datensätze
#21 tx_jobmarket_sectorgroup: betroffene
Datensätze #0
tx_jobmarket_sectorgroup: betroffene
Datensätze #21

lis_kreis: betroffene Datensätze #24
tx_jobmarket_county: betroffene Datensätze
#0
tx_jobmarket_county: betroffene Datensätze
#24

lis_region: betroffene Datensätze #5
tx_jobmarket_region: betroffene Datensätze
#0
tx_jobmarket_region: betroffene Datensätze
#5

ufas_angebot: betroffene Datensätze #736
tx_jobmarket_main: betroffene Datensätze #0
tx_jobmarket_main: betroffene Datensätze
#736
```

Illustration 10: SQL 2 TYPO3 Report



SQL to TYPO3

SICHERHEITS-HINWEIS: Bitte deaktivieren Sie das DRS - Development Reporting System - auf jedem operativem System. Die Fehlermeldung unten ist ein Sicherheitsrisiko!

SQL Fehler

```
Table
'typo3_legthuer.lis_branchengruppe'
doesn't exist

SELECT `BRG_Id`, `BRG_Text` FROM
`lis_branchengruppe` WHERE 1 AND
`BRG_Sprache` LIKE "de"
```

Illustration 11: DRS error report in case of errors

1. URL of SQL 2 TYPO3 page in the frontend:
   here: id=1175&no_cache=1
2. The report in case of a proper rresult.

You will get a qualified error report in case of errors and if you have activated the DRS - Development Reporting System (see section DRS below).

Requirements

There is the requirement only, that all tables have to be part of the same database.

If you want to export data from your source table to the destination table, you have to export your source table into your TYPO3 database first.

Example

See the illustration on the right hand.

1. TYPO3 database
   here: typo3_legthuer
2. One of the source tables:
   ufas_angebot
3. One of the destination tables:
   tx_jobmarket_main.

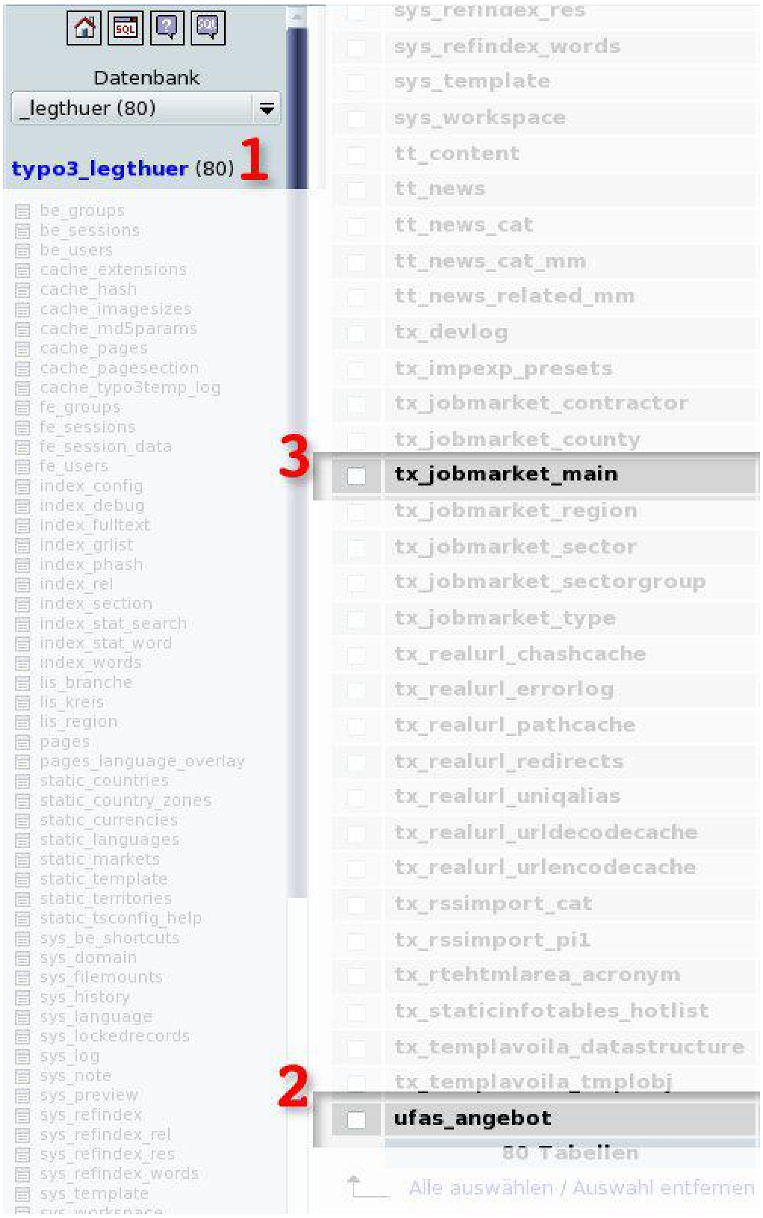


Illustration 12: Same Database

Reference

tables

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th><th>Default:</th></tr>
<tr><td>tables</td><td>array
-&gt; <i>source-table</i></td><td>Array with the names of the source tables.
Every name has to be the name of the source table exactly.

<b>Example:</b>
<pre>
plugin.tx_sql2typo_pi1 {
  tables {
    lis_branche {
      // -> see tables.sourcetable
    }
    ...
  }
}
</pre></td><td></td></tr>
</table>

tables.*sourcetable*

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th><th>Default:</th></tr>
<tr><td><i>sourcetable</i></td><td>string / array</td><td>Array with properties.

<b>Example:</b>
<pre>
plugin.tx_sql2typo_pi1 {
  tables {
    lis_branche {
      source {
        // -> see tables.sourcetable.source
      }
      destination {
        // -> see tables.sourcetable.destination
      }
      fields {
        // -> see next row: sourcetable.fields
      }
    }
    ...
  }
}
</pre></td><td></td></tr>
<tr><td><i>sourcetable</i>.fields</td><td>array</td><td>Array with existing fields in the source table. Values of each filed will be exported to the destination table.
Every field has the TYPO3 stdWrap property.

destination [string]
• Name of the field in the destination table

stdWrap [array / stdWrap]
• TYPO3 stdWrap property

<b>Example:</b>
<pre>
plugin.tx_sql2typo_pi1 {
  tables {
    lis_branche {
      ...
      fields {
        BRC_Id {
          destination = thirdparty_id
          stdWrap = TEXT
          stdWrap {
            trim = 1
          }
        }
        BRC_Text {
          destination = title
        }
      }
    }
    ...
  }
}
</pre></td><td></td></tr>
</table>

tables.*sourcetable*.fields.*field*.relation

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th><th>Default:</th></tr>
<tr><td>....fields.<i>field</i>.relation</td><td>array</td><td>Array with a relation. SQL 2 TYPO3 enables 1:1 relations only.
The extension builds simple relations only:
• YES: local_table > foreign_table
So called M-to-M relations aren't possible:
• NO: local_table > mm_table > foreign_table

<b>Properties:</b>
relation.table [string]
• Name of the destination table
relation.field [string]
• Field in the destination table

<b>Example:</b>

In the code below is
• ufas_angebot the source local table
• tx_jobmarket_main the destination local table
• ufas_angebot.bracnhe_id builds a relation to some table with "branche"
• tx_jobmarket_main.sector buidls the same relation to tx_jobmarket_sector.

The TypoScript below effects the following workflow:
• ufas_angebot is the source table.
• Values of ufas_angebot will export to the destination table tx_jobmarket_main in principal.
• Field branche_id has an array relations.
• SQL 2 TYPO3 will not export values out of branche_id to the destination field sector because of the relation array.
• SQL 2 TYPO3 will export values out of branche_id into the field thirdparty_id in the destination table tx_jobmarket_sector.
• Than SQL 2 TYPO3 will update the field sector in tx_jobmarket_main with the id the record in tx_jobmarket_sector.

<pre>
plugin.tx_sql2typo_pi1 {
  tables {
    ...
    ufas_angebot {
      destination {
        table = tx_jobmarket_main
      }
      fields {
        ...
        branche_id {
          destination = sector
          relation {
            table = tx_jobmarket_sector
            field = thirdparty_id
          }
        }
      }
    }
  }
}
</pre></td><td></td></tr>
</table>

tables.*sourcetable*.fields.*field*.sql2typo.*php_method*

Property:	Data type:	Description:	Default:
....fields.*field*.sql2typo	array	SQL 2 TYPO3 supports the both PHP methods • strtotime • str_replace **Example:** The code will have the effects: • All values in the field datum will be wrapped with the PHP method strtotime() • All values in the field anforderung will be wrapped with the PHP method strreplace() • The parameter devider is a pipe ("|") • str_replace will be processed five times. • Array 0 will become: $value = str_replace("}\r\n{", "</li><li>", $value);	

```
plugin.tx_sql2typo_pi1 {
  tables {
    ...
    ufas_angebot {
      ...
      fields {
        ...
        datum {
          destination = crdate
          sql2typo {
            strtotime = 1
          }
        }
        anforderung {
          destination = specification
          sql2typo {
            str_replace = 1
            str_replace {
              devider = |
              0 = }\r\n{|</li><li>
              1 = } \r\n{|</li><li>
              2 = }\r\n {|</li><li>
              3 = {|<ul><li>
              4 = }|</li></ul>
            }
          }
        }
        ...
      }
    }
  }
}
```

tables.*sourcetable*.source

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th><th>Default:</th></tr>
<tr><td>source</td><td>array</td><td>

Array with properties in context with the source table.

iso2utf [boolean]

0. nothing to do
1. values will be processed by the PHP method utf8_encode()

sql.andWhere [stdWrap / array]
sql.andWhere.value

- Will be added to the SQL WHERE statement of the query SELECT ... FROM *sourcetable* ...

Example:

```
plugin.tx_sql2typo_pi1 {
  tables {
    ...
    lis_branche {
      source {
        iso2utf = 1 // boolean
        sql {
          andWhere = TEXT
          andWhere {
            value = `BRC_Sprache` LIKE "de"
          }
        }
      }
      ...
    }
    ...
  }
}
```

</td><td></td></tr>
</table>

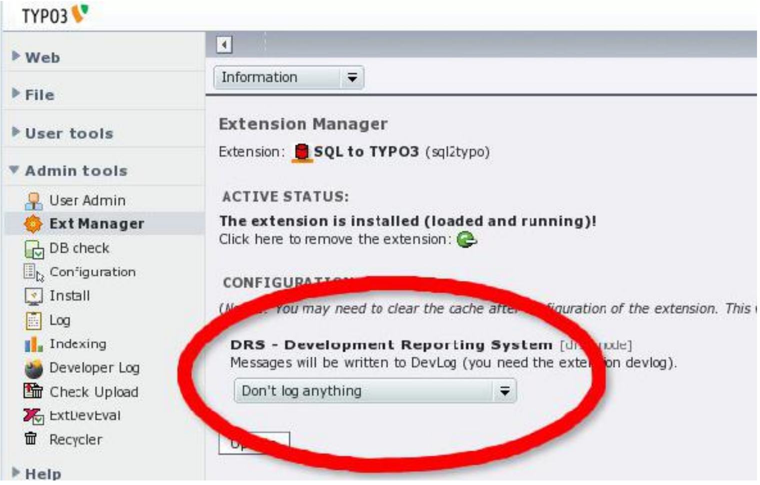
tables.*sourcetable*.destination

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th><th>Default:</th></tr>
<tr><td>destination</td><td>array</td><td>
<p>Array with properties in context with the destination table.</p>
<p>table [string]</p>
<ul><li>Name of the destination table in the database</li></ul>
<p>fields [array]</p>
<ul><li>Array with fields with in the destination table, which should get a value, which isn't a content in the source table. Examples for this fields are: pid, tstamp, crdate, cruser_id and others.</li><li>Every field has the stdWrap property.</li></ul>
<p>sql.beforeImport [string / SQL query]</p>
<ul><li>A SQL query which will be processed before the import will starts.</li></ul>
<p>sql.afterImport [string / SQL query]</p>
<ul><li>A SQL query which will be processed after the import is finished.</li></ul>
<p><b>Example:</b></p>
<pre>
plugin.tx_sql2typo_pi1 {
  tables {
    lis_branche {
      ...
      destination {
        table = tx_jobmarket_sector
        fields {
          // -> see next row: destination.fields
        }
        sql {
          beforeImport = \
            TRUNCATE `tx_jobmarket_sector`
          afterImport =
        }
        ...
      }
    }
    ...
  }
}
</pre>
</td><td></td></tr>
<tr><td>destination.fields</td><td>array</td><td>
<p>Array with existing fields in the destination table.
Every field has the TYPO3 stdWrap property.</p>
<p><b>Example:</b></p>
<pre>
plugin.tx_sql2typo_pi1 {
  tables {
    lis_branche {
      ...
      destination {
        ...
        fields {
          pid {
            stdWrap = TEXT
            stdWrap {
              value = 125
            }
          }
          tstamp {
            stdWrap = TEXT
            stdWrap {
              data = date:U
            }
          }
        }
      }
    }
    ...
  }
}
</pre>
</td><td></td></tr>
</table>

DRS - Development Reporting System

DRS means Development Reporting System. The DRS is supporting the management of TypoScript and if you like to develop the SQL 2 TYPO3, you will get support too.

You will get information about the work flow and the TypoScript configuration, if you choose one of the options of the DRS (see image below).



Extension Manager

Extension: SQL to TYPO3 (sql2typo)

ACTIVE STATUS:

The extension is installed (loaded and running)!

Click here to remove the extension:

CONFIGURA

(N . You may need to clear the cache afte figuration of the extension. This

DRS - Development Reporting System [d ode]

Messages will be written to DevLog (you need the exte on devlog).

Don't log anything

Illustration 13: DRS - Development Reporting System

Requirements

You need the extension devlog

Without the extension devlog you won't get any message in the backend.

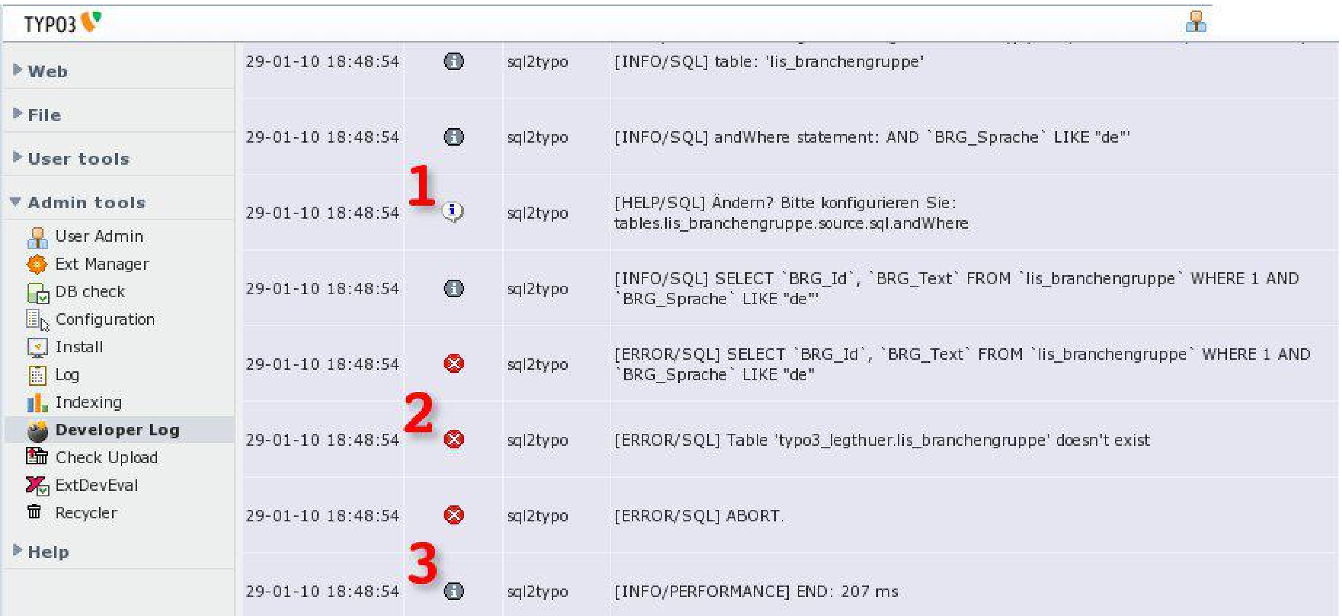
Languages

The DRS is in English and German.

Options

- Don't log anything
  This should be the default.
- Performance
  Parses the time.
- SQL Development
  Displays all generated SQL queries.
- All

Example



```
29-01-10 18:48:54    sql2typo    [INFO/SQL] table: 'lis_branchengruppe'
29-01-10 18:48:54    sql2typo    [INFO/SQL] andWhere statement: AND `BRG_Sprache` LIKE "de"'
29-01-10 18:48:54    sql2typo    [HELP/SQL] Ändern? Bitte konfigurieren Sie:
                                 tables.lis_branchengruppe.source.sql.andWhere
29-01-10 18:48:54    sql2typo    [INFO/SQL] SELECT `BRG_Id`, `BRG_Text` FROM `lis_branchengruppe` WHERE 1 AND
                                 `BRG_Sprache` LIKE "de"'
29-01-10 18:48:54    sql2typo    [ERROR/SQL] SELECT `BRG_Id`, `BRG_Text` FROM `lis_branchengruppe` WHERE 1 AND
                                 `BRG_Sprache` LIKE "de"
29-01-10 18:48:54    sql2typo    [ERROR/SQL] Table 'typo3_legthuer.lis_branchengruppe' doesn't exist
29-01-10 18:48:54    sql2typo    [ERROR/SQL] ABORT.
29-01-10 18:48:54    sql2typo    [INFO/PERFORMANCE] END: 207 ms
```

Illustration 14: DRS report (cut out)

1. Help for configuring TypoScript:
   There will be a speech balloon, if there is a possibility to control a property by TypoScript.

2. Error messages:
   The DRS should report a qualified error message, if there will be an error.

3. Performance:
   The DRS displays the needed time.

Helpful suggestions

If you have helpful suggestions, feel free to contact me: http://wildt.at.die-netzmacher.de.

Further Information

About the plugin icon

The concept of the icon:

- The red cylinder should symbolize databases.
- The color yellow is the color of Die Netzmacher, the company which created SQL 2 TYPO3.

Some other extensions published by Die Netzmacher

Browser - The Frontend Engine. It displays content from related tables. You need one line typoscript for a result list with a search form, a record browser and an a-z browser. Images are wrapped self-acting. SEO, Search Engine Optimisation is integrated. DRS - Development Supporting System - supports Bugfixing and Development.
http://typo3.org/extensions/repository/view/browser/current/

Browser for BZD Staff Directory. Adapt BZD Staff Directory to your demands.
http://typo3.org/extensions/repository/view/browser_bzdstaffdirectory/current/

Browser for Virtual Civil Services - It supersedes smarty and enables SEO - search engine optimisation -, Real URL, cache management. Adapt Virtual Civil Services to your demands.
http://typo3.org/extensions/repository/view/browser_civserv/current/

§§ Juridat - Database for Juridical Data. Juridat provides a juridical data base with backend functionality.
http://typo3.org/extensions/repository/view/juridat/current/

Job Market - A catalogue for jobs. Easy to adapt to your demands. Available from May of 2010.
http://typo3.org/extensions/repository/view/job_market/current/

green_cars: Database for cars with focus on environmental aspects. Case study for the Extension Browser.
http://typo3.org/extensions/repository/view/green_cars/current/

logical_form: A frontend plugin for evaluating forms. I.e you can evaluate mail addresses.
http://typo3.org/extensions/repository/view/logical_form/current/

Majordomo: For subcribing to and unsubscribing from a majordomo mailing list.
http://typo3.org/extensions/repository/view/majordomo/current/

seo_dynamic_tag: Search Engine Optimisation. You can generate values dynamically with this extension especially for the <title>-tag, for the <meta>-tag description and the <meta>-tag keywords.
http://typo3.org/extensions/repository/view/seo_dynamic_tag/current/

tt_news Marker (extkey: ttnews_marker) enables to add any marker to a tt_news HTML template including links. Any marker has the std_wrap property.
http://typo3.org/extensions/repository/view/ttnews_marker/current/

SQL tt_news select configuration (extkey: ttnews_selectconf) enables to select tt_news by any SQL clause. I. e. it is possible to display tt_news items in dependence on the ownership of a fe_user.
http://typo3.org/extensions/repository/view/ttnews_selectconf/current/

Wine Catalogue is a localized database with list and single views optimized for wines.
http://typo3.org/extensions/repository/view/wine/current/

All extensions published by Die Netzmacher

http://typo3.org/extensions/repository/?tx_terfe_pi1[view]=search&no_cache=1&tx_terfe_pi1[sword]=netzmacher

To-Do List

Nothing to do.

Change Log

0.0.1

- Initial release

Illustration Index