

Documentation

MAGEconnect (TYPO3 Extension) & TYPO3connect (Magento Module)

 **Easy-to-implement solution of connecting Magento and TYPO3**

Language: de, en

Keywords: magento, typo3, mageconnect, typo3connect

Author: Bastian Zagar

E-Mail: kontakt@magedeveloper.de

Contents of this Manual:

1	Introduction
2	What is Single SignOn/SignOff?
3	Login sequence
4	Screenshots (Login & Product Display)
5	Installation and Configuration Magento
6	Installation and Configuration TYPO3
7	TYPO3 Adding Login form
8	TYPO3 Displaying Products
9	TYPO3 Forgot Password Ajax Form
10	TYPO3 Display Category Tree
11	Import TYPO3 Pages to Magento
12	Fluid View Helper
13	Important things to know
14	Configuration
15	Getting dynamic customer attributes
16	Useful login integration examples
17	Changelog

Easy-to-implement solution of connecting Magento and TYPO3

- **Single-SignOn/SignOff to TYPO3 and Magento**
- **Magento Product Display in TYPO3 with different display types**
- **Import TYPO3 Content-Pages to Magento**
- **Category Navigation / Category Products**
- **Store View Compatibility**

The module package consists of two parts:

MAGEconnect:

TYPO3 Extension for adding Single-SignOn and product display in TYPO3 aswell as product display methods.

TYPO3connect:

Magento Module that delivers an interface for login connection of the MAGEconnect Login form.

Both systems talk to each other with http Requests and SOAP Calls, so they don't have to be in the same location.

The Modules are designed for easy implementation and administration.

Once you've installed them to your systems, TYPO3 and Magento, you only have to setup a few things like Url and Encryption Key.

Since you don't have an SSL Certificate on your server the modules also encrypt/decrypt posted data for higher security.

Features of both plugins:

- A small login form for TYPO3 that is included as plugin to the page.
- Customers are directly imported to the TYPO3 fe_users database
- You can choose SysFolder and Usergroup for Users; if you change the usergroup for customers, they will stay in the changed group, either you change it again.
- Highly customization methods as described in the integration example in this documentation
- Imports the Magento users to TYPO3
- Returning to different locations after login (configurable in plugin)

2

WHAT IS SINGLE SIGNON/SIGNOFF

The modules connect the login/logout procedure of Magento and TYPO3. If you login to TYPO3, you're being logged in also to Magento.

The process is not directly viewable by your customers, it happens invisible on both systems.

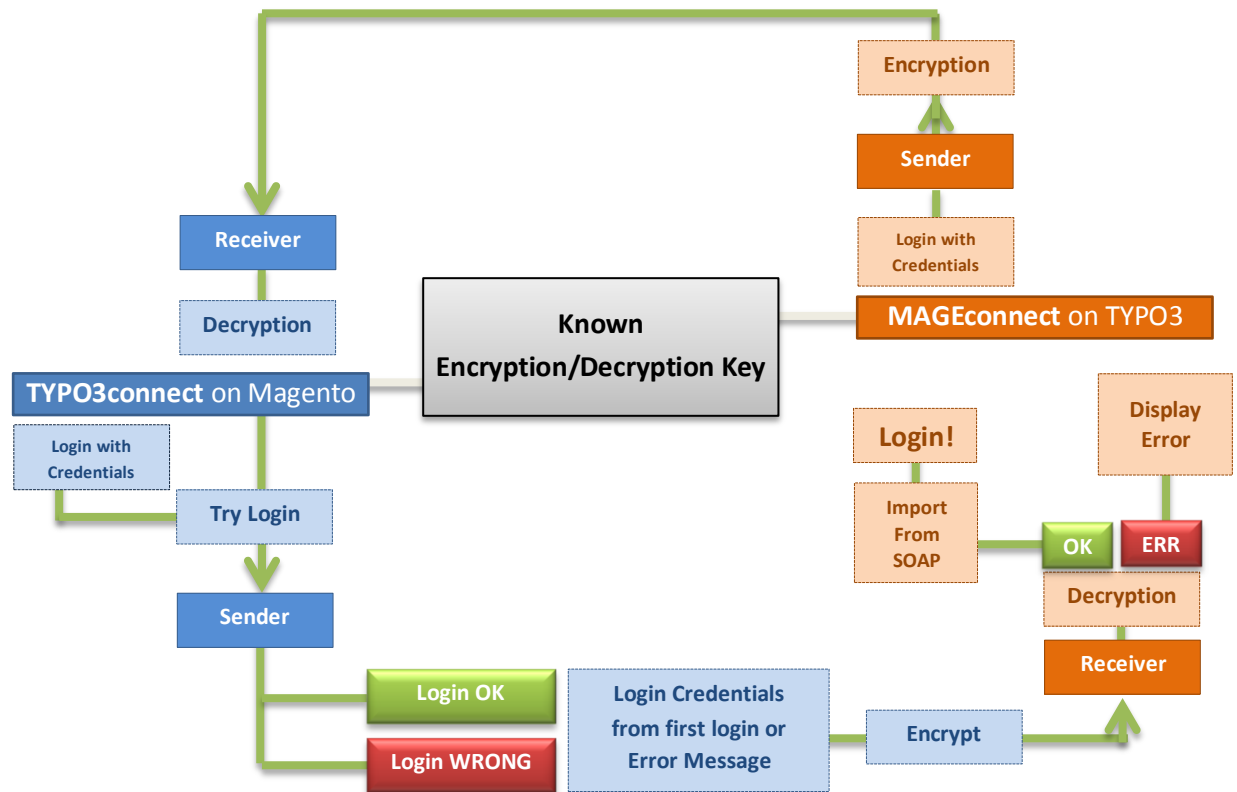
After the user has logged in and the credentials are valid, the user will be imported to TYPO3. This means that the Magento-System is the major location for customer data.

With the single SignOn, you also have the possibility to read your custom attributes that you added by your own customer modules in Magento.

As written in the Part 9 "Getting dynamic customer attributes" in this documentation you can fully access customer information that was added to the Magento Customer.

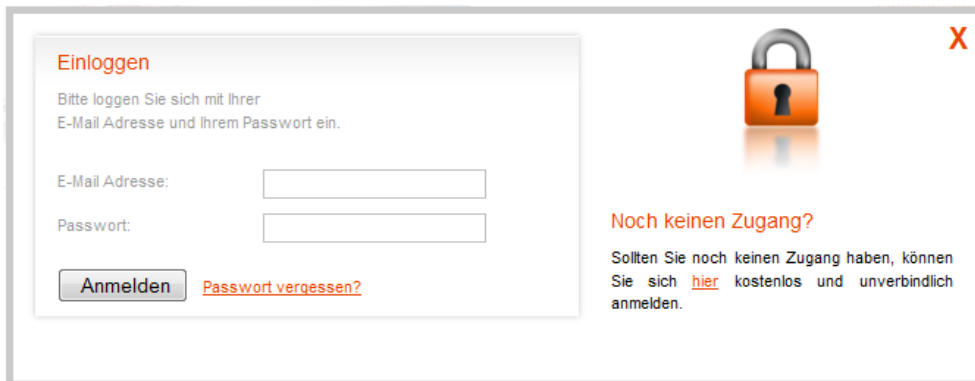
This means that you can add custom text-fields that will also automatically imported and are accessible in TYPO3.

As mentioned, TYPO3connect is the adapter for Magento and MAGEconnect the opposite on TYPO3. Here is some information about the Login-Sequence of both modules:



Every side has the same encryption/decryption key that the messages can be decrypted/encrypted. There happens a bit more in the sequence but this is the major procedure.

As fading modal dialog on every page:

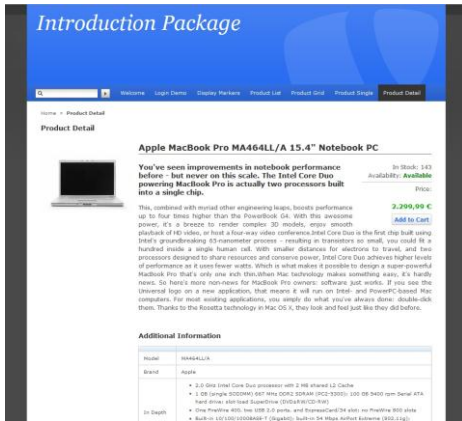


The screenshot shows a login modal dialog box with a white background and a grey border. On the left side, there is a form titled "Einloggen" (Login). Below the title, it says "Bitte loggen Sie sich mit Ihrer E-Mail Adresse und Ihrem Passwort ein." (Please log in with your email address and password). There are two input fields: "E-Mail Adresse:" and "Passwort:". Below the password field is a button labeled "Anmelden" (Login) and a link "Passwort vergessen?" (Forgot password?). On the right side of the modal, there is a large orange padlock icon. Below the icon, it says "Noch keinen Zugang?" (Still no access?). Underneath, it says "Sollten Sie noch keinen Zugang haben, können Sie sich [hier](#) kostenlos und unverbindlich anmelden." (If you still have no access, you can register for free and without commitment [here](#)).

As login form on a single page:

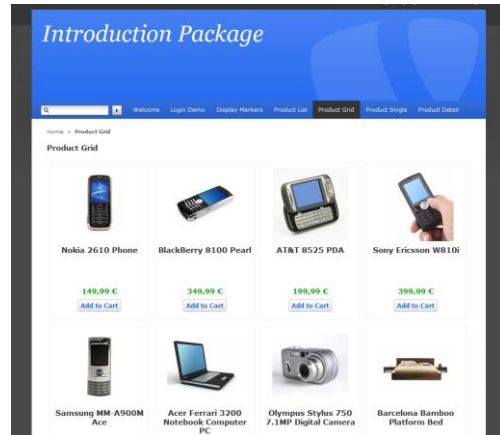


The screenshot shows a login form on a single page with an orange background. At the top left, it says "Login" next to a small icon. Below this, there are two input fields: "Username" and "Password". At the bottom, there is a button labeled "Einloggen" (Login) and a link "Passwort vergessen?" (Forgot password?).



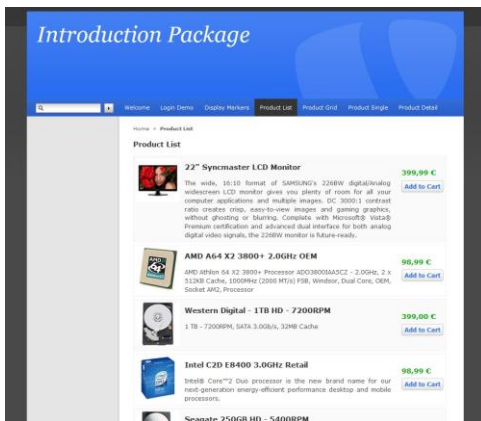
Product Detail View

Display a detailed product page with attribute information



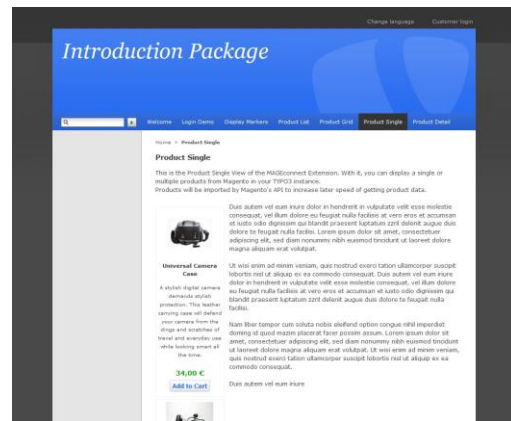
Product Grid View

Display chosen products in a grid.



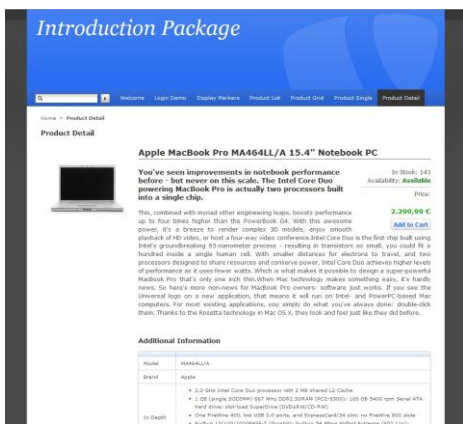
Product List View

Display chosen products in a list



Product Single View

Add chosen products to the left (CSS defined) part of any content



Dynamic Detail View

Display product details within TYPO3

Shop	Produktportfolio	Dienstleistungen	Unternehmen	Termine	Service	Kontak
Produktübersichten	>	Beschriftungsgeräte		DVI-Kabel		
MobiltelefoneDE	>	Computer u. Mini PC		HDMI-Kabel		
Netzbetreiber Produkte	>	Computerkabel	>	Netzwerkkabel		
Festnetz	>	Computer Peripherie u. Zubehör	>	USB-Kabel		
Home Entertainment	>	Drucker und Multifunktionsgeräte		VGA-Kabel		
Computer	>	Monitore/Displays		XGA-Kabel		
Navigation / Car Entertainment	>	Netzwerkprodukte				
Zubehör	>	Notebooks				
Foto und Video	>	Software/Garantieverlängerungen				
Produkte für Senioren	>	Tablet PC				
Audio-Video-Konferenz	>	Tablet PC Zubehör				
Werbematerial	>	Verbrauchsmaterial	>			

- Furniture (0)
 - Living Room (4)
 - Bedroom (2)
- [Test] (1)
- Electronics (0)
 - Cell Phones (6)
 - Cameras (1)
 - Accessories (3)
 - Digital Cameras (5)
 - Computers (7)
 - Build Your Own (3)
 - Laptops (4)
 - Hard Drives (4)
 - Monitors (4)
 - RAM / Memory (3)
 - Cases (2)
 - Processors (4)
 - Peripherals (4)
- Apparel (0)
 - Shirts (19)
 - Shoes (0)
 - Mens (22)
 - Womens (34)
 - Hoodies (1)
- NewFromTestStoreEnglish! (0)

E-Mail Adresse:

If there is an account associated with kontakt@magedeveloper.de you will receive an email with a link to reset your password.

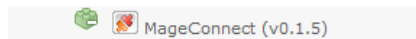
Requirements

Please note the following things to let the plugin work properly:

- **Install the PHP SOAP Module in case you get an “Class 'soapclient' not found” Error**
- **Since Magento 1.7 was released, there were a few rewrite changes for catching the WSDL. We recommend that you use [index.php/](#) in the Magento URL, if you’re experiencing SOAP Errors!**

1. Install the extension to your TYPO3 installation

Install the extension by local download or via Extension Manager. After installation it should appear in your “Available Extensions” List



2. Configure the extension

By clicking on the extension name, you will come to the main configuration of the extension.

Magento URL [magento_url]

Frontend URL to your Magento Shop (e.g. <http://www.yourshop.com/de/>)

Magento SOAP Username [magento_api_username]

Username of the user that has Magento API access

Magento SOAP Key [magento_api_key]

Magento API Key

Encryption/Decryption Key [encrypt_decrypt_key]

Key that is used to encrypt/decrypt the connection data between "TYPO3connect" and "MAGEconnect". Please enter the same value in your Store Config in Magento!

Backend Cache Lifetime [cache_lifetime]

Set the backend cache lifetime in seconds

Debug Mode [debug_mode]

Enable for debug mode

Redirect location on global logout [redirect_on_logout]

Use this setting to set up the redirect location on logout in typo3

Description of the fields can be found on the next page!

5

INSTALLATION AND CONFIGURATION (TYPO3)

Magento Shop URL

Please enter your Magento Base Shop URL. You should use https for security!

If you are experiencing SOAP Errors, you can try adding [index.php/](#) to your URL.

To determine the correct URL, you can try to open the WSDL of the API directly within your browser like this
Add [api/v2_soap?wsdl=1](#) to your url, e.g. http://shop.magedeveloper.de/index.php/api/v2_soap?wsdl=1

If it is the correct URL, you will get information similar to the following screenshot:

```
-<definitions name="Magento" targetNamespace="urn:Magento">
  -<types>
    -<schema targetNamespace="urn:Magento">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" schemaLocation="http://schemas.xmlsoap.org/soap/encoding/" />
    -<complexType name="associativeEntity">
      -<all>
        <element name="key" type="xsd:string"/>
        <element name="value" type="xsd:string"/>
      </all>
    
```

Magento SOAP Username

Add a SOAP User in Magento (*please read Part 6 "Installation and Configuration Magento" of this Manual*)

Magento SOAP Key

Enter the SOAP Key for the Magento SOAP User

Encryption/Decryption Key

Use this key in TYPO3 and Magento for correct communication between the systems.

Backend Cache Lifetime

Enter the cache lifetime in seconds for the stored product list in backend.

Debug Mode

Enable debugging information while login procedure

Please, only use this in development environment! It shows passwords and user details!

Redirect location on global logout

With this option, you can set where Magento or TYPO3 is redirecting, when a user logs out.

For logout on the same system, please choose "System of logout"

Developer Settings

Redirect Method on successful login

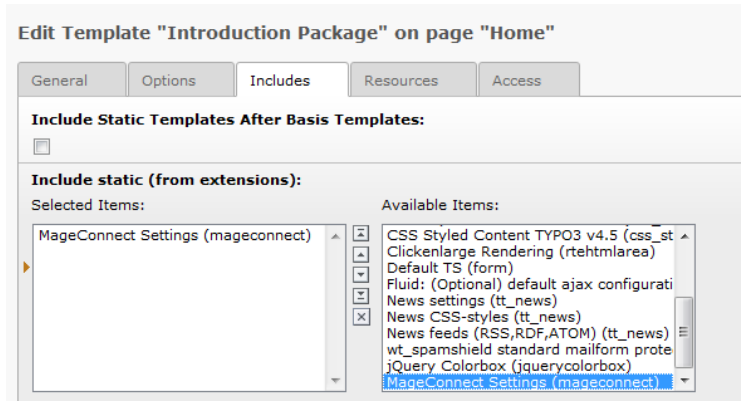
Change this option if you have redirect problems.

5

INSTALLATION AND CONFIGURATION (TYPO3)

3. Add static

Go to your page template and add the MageConnect Static Template like shown in the next picture:



6

INSTALLATION AND CONFIGURATION (MAGENTO)

1. Install the module to your Magento installation

Install the module via the Magento Connect Manager. After installation it should appear in your “Existing Extensions” List.

TYPO3connect	0.1.2 (beta)	<input type="text"/>	Connects TYPO3 and Magento.
--------------	--------------	----------------------	-----------------------------

2. Configure the module

Go to System – Configuration and find the row that is named “Services”. Within this, you should find a new option called “TYPO3connect”. Click the name to start configuration.

TYPO3connect

General		
URL of the TYPO3 Login Site	<input type="text" value="http://cms.magedeveloper.de/mageconnect/login-1"/>	[STORE VIEW]
	<small>▲ Please enter the URL of the TYPO3 Login Site</small>	
Shared Key for decryption	<input type="text" value="vnjGik3T2TfGn9pb"/>	[STORE VIEW]
	<small>▲ Please enter the same key as used in TYPO3</small>	
Display Login Template	<input type="text" value="Yes"/>	[STORE VIEW]
	<small>▲ Displays an login template, when TYPO3 connects to Magento at login</small>	

URL of the TYPO3 Login Site

Please enter the direct URL of the TYPO3 Login Site.

Please note, that we highly recommend the use of SSL e.g. <https://typo3site.de/login/>

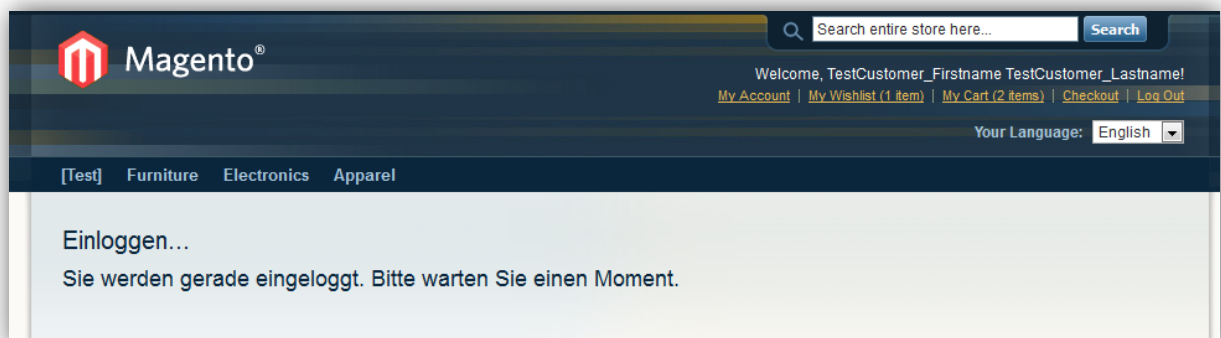
Shared Key for decryption

Please enter the same key, that you’ve entered in TYPO3

Display Login Template

Enable this option if you want a visible login procedure (see following Screenshots)

If Enabled:



6

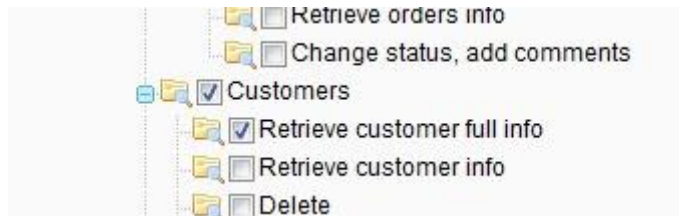
INSTALLATION AND CONFIGURATION (MAGENTO)

3. Creating access right for the SOAP User

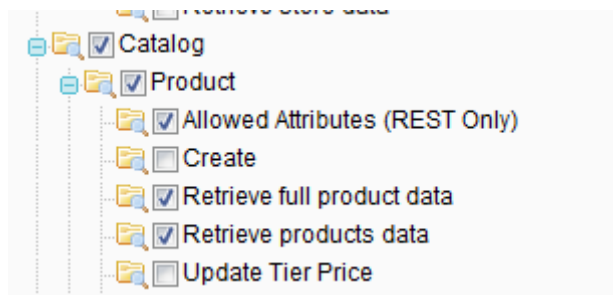
Go to System – Web Services – Roles and add a new role to create access for the user, you are going to add in Step 4.



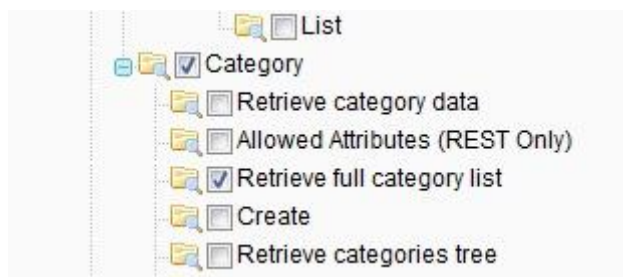
You don't need to give full right, just check the following boxes for the login!



For the product display, please check the following boxes:



For the category tree display, please check the following boxes:



6

INSTALLATION AND CONFIGURATION (MAGENTO)

4. Adding the SOAP User

Go to System – Web Services – Users and add a new user.

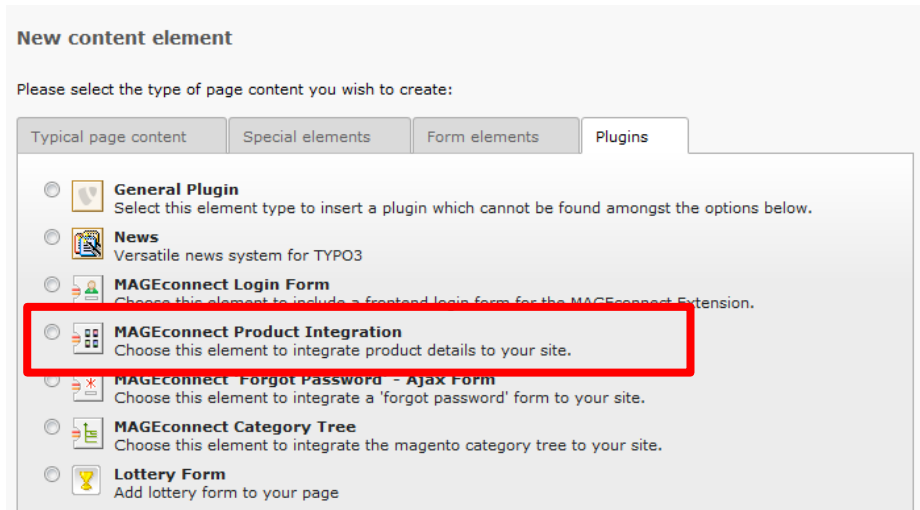
Account Information	
User Name *	<input type="text" value="soapuser"/>
First Name *	<input type="text" value="MAGEconnect"/>
Last Name *	<input type="text" value="SOAPuser"/>
Email *	<input type="text" value="kontakt@magedeveloper.de"/>
API Key *	<input type="text" value="....."/>
API Key Confirmation *	<input type="text" value="....."/>
This account is	<input type="text" value="Active"/> <input type="button" value="v"/>

Don't forget to put the user in the role, you've made in Step 3.

User Information	New User								
User Info <input type="button" value="v"/>	Page <input type="text" value="1"/> of 1 pages View <input type="text" value="20"/>								
User Role <input type="button" value="v"/>	<table border="1"><thead><tr><th>Assigned</th><th>Role Name</th></tr></thead><tbody><tr><td>Any <input type="button" value="v"/></td><td></td></tr><tr><td><input type="radio"/></td><td>EVERYTHING</td></tr><tr><td><input checked="" type="radio"/></td><td>MAGEconnect</td></tr></tbody></table>	Assigned	Role Name	Any <input type="button" value="v"/>		<input type="radio"/>	EVERYTHING	<input checked="" type="radio"/>	MAGEconnect
Assigned	Role Name								
Any <input type="button" value="v"/>									
<input type="radio"/>	EVERYTHING								
<input checked="" type="radio"/>	MAGEconnect								

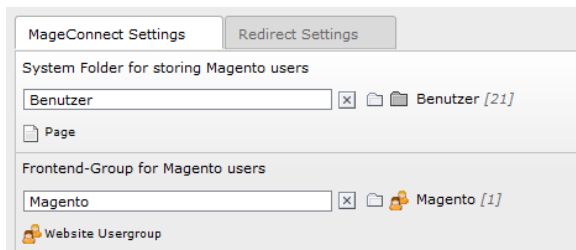
1. Add the Plugin to your site

By clicking the add content button to your site, you are now able to choose “MAGEconnect Login Form”.



2. Configure the Plugin

Tab MageConnect Settings



System Folder for storing Magento users

Choose a SysFolder where the imported Magento-Users will be stored.

Frontend-Group for Magento users

Choose the frontend group which imported Magento-Users will be assigned to.

After the user is imported, you can change to usergroup, it will not be overwritten, when the user is already in TYPO3

While login procedure, the customer is assigned to the usergroup that you've chosen in the login-plugin. Aside this, MAGEconnect will also try to assign the user to the customer group, that the customer has in Magento. This means, if an similar usergroup with the same name exists in TYPO3, the user will automatically assigned to this group.

Customer Information

- Customer View
- Account Information**
- Addresses
- Orders
- Billing Agreements
- Recurring Profiles (beta)
- Shopping Cart

TestCustomer_Firstname TestCustomer_Lastname

Account Information

Associate to Website *

Created From

Customer Group *

Disable Automatic Group Change Based on VAT ID

Create new Website Usergroup on page "Mageconnect Benutzer"

General Options Extended

Disable:

Group Title:
MyCustomCustomerGroup

Description:
Will be automatically assigned!

Disable:

Username:

Password:

Groups:

Selected Items:	Available Items:
<ul style="list-style-type: none"> MyCustomCustomerGroup Magento Benutzer 	<ul style="list-style-type: none"> Magento Benutzer MyCustomCustomerGroup NewTestGroup

Input in Lottery Field

Magento Customer Attributes

Last login:

While login procedure, the customer attribute that come from Magento will be synced with the TYPO3 User.

For example:

If customer in Magento has an attribute called "title" it will be synced with the fe_user Column "title".

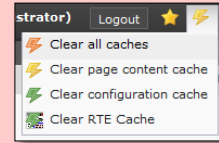
So, if you have the same

Magento Attribute Code ⇔ TYPO3 fe_user column name

the data is being synced!

Please note, that the product-display plugin first will load all possible products for displaying from the Magento API, so the first call of opening is slow. After first load the data is cached and you can access the plugin more quickly! Loading goes faster if Magento's Cache is enabled!

If you want to reload available products, you will have to clear the cache!



1. Include the product-display plugin to your site

By clicking the add content button to your site, you are now able to choose "MAGEconnect Product Integration".

2. Configure the plugin

Tab MageConnect Settings

System Folder for storing Magento products

Since the Magento API is very slow, it is necessary to cache the magento products. For this purpose, you have to set a folder, where the product-data can be stored within.

System Folder for storing Magento categories

Please choose a sys-folder where MAGEconnect can store the Magento category structure.

Reload products on page load

You can enable this option if you want to update the product data that is stored in the Magento products folder. If you go to the frontend page of the plugin while the checkbox is enabled, it will always load and import the product data from the Magento API

Tab Product Configuration

Optional: Store View Code (Reloads the product list!)
otherstore

Display Type
Products list

Product IDs Source
Manual product selection

Show Products

Selected Items:

- [17] BlackBerry 8100 Pearl
- [20] Samsung MM-A900M Ace TESTSTORE
- [16] Nokia 2610 PhoneT
- [28] Toshiba M285-E 14"

Available Items:

- [16] Nokia 2610 PhoneT
- [17] BlackBerry 8100 Pearl
- [18] Sony Ericsson W810i
- [19] AT&T 8525 PDA
- [20] Samsung MM-A900M Ace TESTSTORE
- [25] Apple MacBook Pro MA464LL/A 15.4" Notebo
- [26] Acer Ferrari 3200 Notebook Computer PC
- [27] Sony VAIO VGN-TXN27N/B 11.1" Notebook P
- [28] Toshiba M285-E 14"
- [29] CN Clogs Beach/Garden Clog

Page of Dynamic Detail View (If set, all links go to this page)
Detail View Page

Edit Store View

Store View Information

Store * Main Store

Name * My Other Store

Code * otherstore

Status * Enabled

Sort Order 0

Store View Code

Please enter the store view code, where you want the product list to be fetched from. You may leave it blank in order to not use any store view.

To refresh the product list for an store view, enter your store code and click somewhere outside the text field

Display Type

Choose between the different display types for the products.

Please refer to the screenshots section on part 4 of this documentation to get information about the display types.

Display available template markers

This option show available template markers that you may use in the templates for the display types. Since the product markers are included dynamically, you can use this possibility to have a look, what you can possible use.

Home > Display Markers

Display available markers

Marker	Value
##DETAIL_BIRTH_ID###	18
##DETAIL_ATTRIBUTE_SET_ID###	38
##DETAIL_TYPE_ID###	simple
##DETAIL_BIRTH_TYPE_ID###	10
##DETAIL_NAME###	Sony Ericsson W810i
##DETAIL_MODEL###	W810i
##DETAIL_ID###	10610
##DETAIL_WEIGHT###	13.4000
##DETAIL_MANUFACTURER###	2
##DETAIL_QTY_ID###	
##DETAIL_STATUS###	1
##DETAIL_TAX_CLASS_ID###	2
##DETAIL_URL_KEY###	sony-ericsson-w810i
##DETAIL_VISIBILITY###	4
##DETAIL_NEWS_FROM_DATE###	

Product IDs Source

Please choose if you want either to manual select products for displaying or select a category with all their products.

- Selected Category**
Please select a category. All products of this category will displayed a the display type you've chosen
- Show Product(s)**
Select the desired products, that you want to display. The products will be displayed exactly as you added them in the "Selected Items"-Row. This means you also can add the same product twice or more.
- By Category/Tags**
You can enter Category Names and Tag Names to filter products

Page of Dynamic Detail View

Please set up the page if you want to display product details in TYPO3. The page set here has to have the product integration plugin with Display Type "Dynamic Detail View".

If this isn't set, the links to the products will refer to Magento's Product Pages.

The screenshot shows the 'Product Configuration' section of the MageConnect Settings. It includes the following fields:

- Optional: Store View Code (Reloads the product list!)**: Input field with 'otherstore'.
- Display Type**: Dropdown menu with 'Products list' selected.
- Product IDs Source**: Dropdown menu with 'Selected Category' selected.
- Selected Category**: Dropdown menu with '[10] Furniture (Count: 5)' selected.
- Page of Dynamic Detail View (If set, all links go to this page!)**: This field is highlighted with a red box. It contains a dropdown menu with 'Detail View Page' selected, followed by a close icon (x), a folder icon, and the text 'Detail View Page [95]'.
- Page**: A checkbox labeled 'Page' is visible at the bottom.

MAGEconnect now offers dynamic product detail views within TYPO3. The following instructions will guide you to setup an dynamic product catalog.

Listener Page / Dynamic Detail View Page

You have to create a new page in your TYPO3 instance for using it as the detail site for different products.

You also have to configure, which products are allowed for using with the Dynamic Detail View.

If you leave it empty, every product is allowed.

Here is an setup example:

When a product is not allowed for Dynamic Detail View, the user will get an error in the frontend.

The store view option counts, so you need to watch for the right store view code, if you use this feature!

Grid-, List-, Single-View Setup for Dynamic Detail View

Generally MAGEconnect refers every product to Magento's Product Detail Site. In order to view the product details in your TYPO3 instance you have to setup the Dynamic Detail View in the Product Integration Plugin.

Please refer to the following screenshot.

ADDING PRODUCT DISPLAY PLUGIN (TYPO3) Single Product Always Auto-Refresh

With the new MAGEconnect Version, you have the possibility to let a single product auto-refresh on page load. To enable this function, you only need to edit the product record and enable the “Always refresh”-Checkbox.

Go to your folder, where the products are stored and edit the product, you want to reload every page refresh.

The screenshot displays the Mageconnect Products management interface. At the top, there are navigation tabs for 'Mageconnect Benutzer' and 'Mageconnect Products'. Below this, a modal window titled 'Mageconnect Products' is open, showing a list of products under the heading 'Product (4)'. The products listed are:

- [Store: -][ID 16] Nokia 2610 Phone
- [Store: -][ID 17] BlackBerry 8100 Pearl
- [Store: -][ID 18] Sony Ericsson W810i
- [Store: -][ID 41] Akio Dresser

Each product row includes an 'Edit record' button (pencil icon), an information icon, a refresh icon, and a delete icon. A red box highlights the 'Edit record' button for the first product.

Below the product list, the 'Always refresh' checkbox is checked and highlighted with a red box. The main content area shows the details for the selected product (ID 16):

- Product Id:** 16
- SKU:** n2610
- Productname:** Nokia 2610 Phone
- Short Description:** The words "entry level" no longer mean "low-end," especially when it comes to the Nokia 2610. Offering advanced media and calling features without breaking the bank

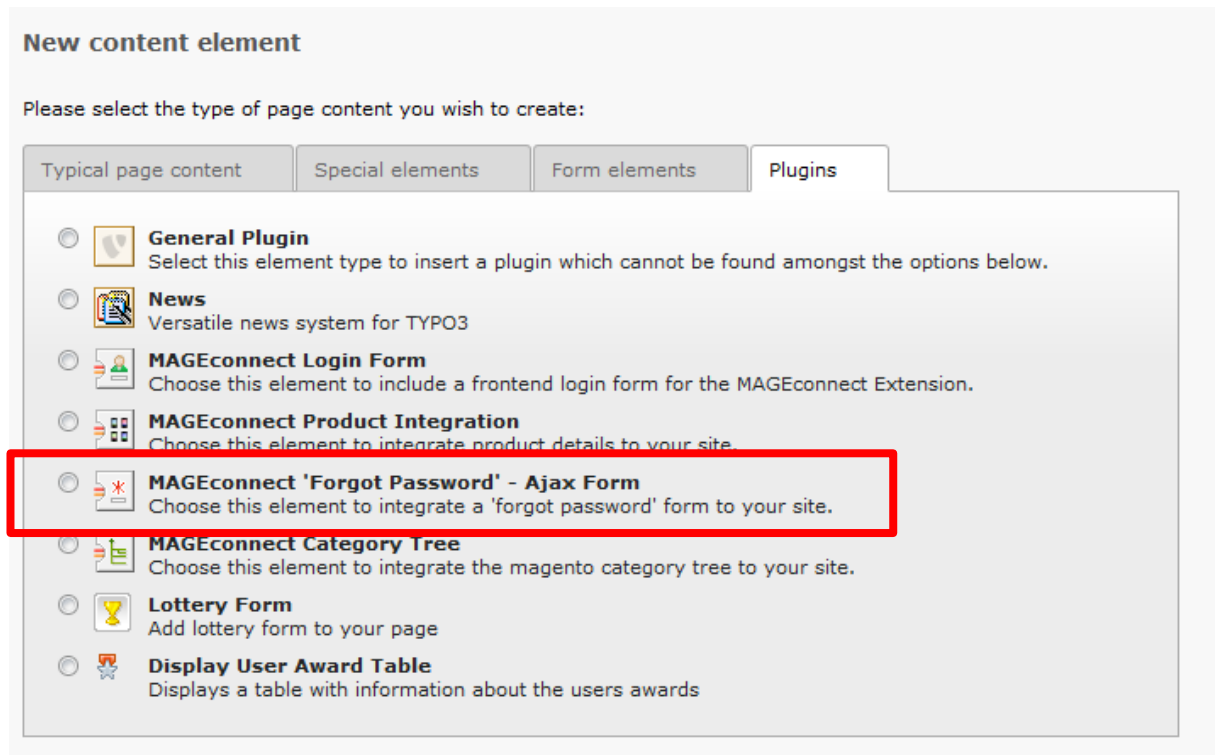
The left sidebar shows the TYPO3 navigation tree, including 'Home', 'MAGEconnect', 'Login Demo', 'Forgot password Demo', 'Category Tree', 'Display Markers', 'Product List', 'Product Grid', 'Product Single', 'Product Detail', 'Dynamic Detail View', 'Detail View Page', 'TYPO3 Pages to Magento', 'Other Extensions', 'XML Adapter - Pages List', 'Lottery', 'Awards', 'Award Data', 'Mageconnect Benutzer', 'Mageconnect Products', 'Mageconnect Categories', 'Special pages', 'Generated content', 'News entries', 'Frontend users and groups', and 'TypoScript Templates'.

Please note, that the forgot-password plugin uses jQuery, so please install t3jquery or include jQuery to your website-template.

You can download the t3jquery-Extension under the following URL:

<http://typo3.org/extensions/repository/view/t3jquery>

To include a forgot password form to your page, just include the plugin to your page.



New content element

Please select the type of page content you wish to create:

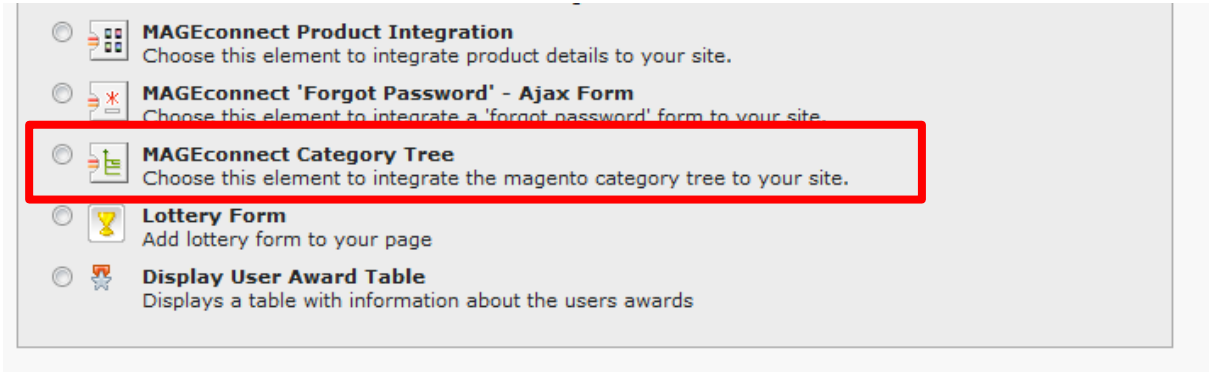
Typical page content Special elements Form elements **Plugins**

- General Plugin**
Select this element type to insert a plugin which cannot be found amongst the options below.
- News**
Versatile news system for TYPO3
- MAGEconnect Login Form**
Choose this element to include a frontend login form for the MAGEconnect Extension.
- MAGEconnect Product Integration**
Choose this element to integrate product details to your site.
- MAGEconnect 'Forgot Password' - Ajax Form**
Choose this element to integrate a 'forgot password' form to your site.
- MAGEconnect Category Tree**
Choose this element to integrate the magento category tree to your site.
- Lottery Form**
Add lottery form to your page
- Display User Award Table**
Displays a table with information about the users awards

There is no more setup needed for this plugin.

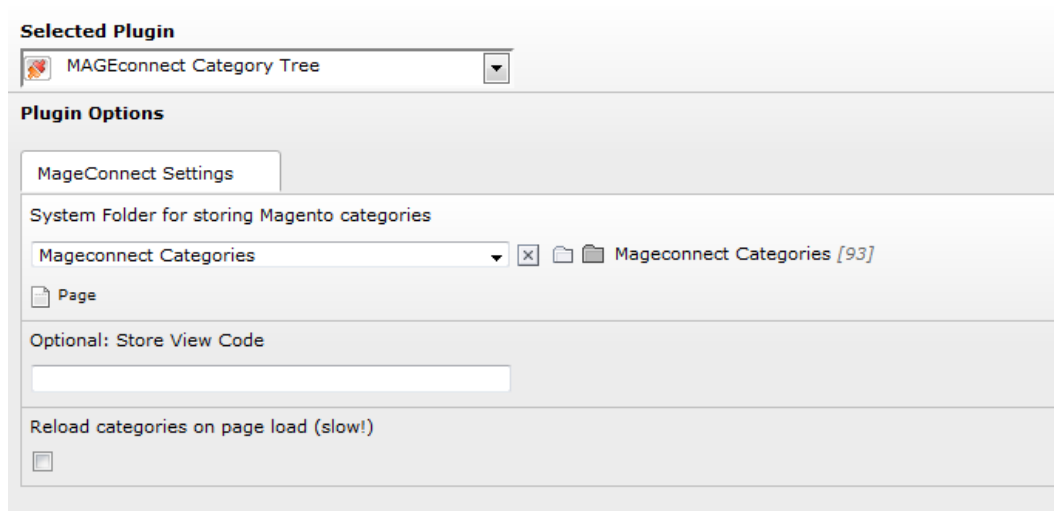
1. Include the plugin to your site

To include the category tree in the content, please include the plugin to your page.



2. Configure the plugin

Tab MageConnect Settings

A screenshot of the MageConnect Settings configuration page. The 'Selected Plugin' dropdown is set to 'MAGEconnect Category Tree'. Under 'Plugin Options', there is a 'MageConnect Settings' tab. The 'System Folder for storing Magento categories' is set to 'Mageconnect Categories'. There is a 'Page' section and an 'Optional: Store View Code' input field. At the bottom, there is a checkbox for 'Reload categories on page load (slow!)' which is currently unchecked.

System Folder for storing Magento categories

Choose a System-Folder, where the plugin can save the imported categories.

Store View Code

Please enter your store view code, if you want to use this feature.

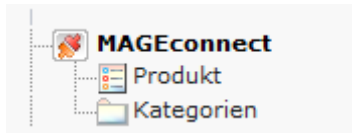
Reload categories on page load

Enable this option, if you want all categories to be reloaded on page refresh.

Manual creation of category elements

If you want to add navigation elements to the category tree, you can add a new category record to the category list in the sys-folder of your categories.

Just insert a new record



If you want this record to stay on the next category refresh, you need to set the checkbox “User Defined”
The sort order is the same order as in the backend shown. You may manually edit the order if you want.

Important fields to fill out:

Parent Category Id -> Magento Root Category Id (**Standard 2**)

Category Name

Category Url

StoreView if needed

Please note, that you need the TYPO3 Extension “xml_adapter” to be able to import content pages to Magento!

You can download the “xml_adapter”-Extension under the following URL:

http://typo3.org/extensions/repository/view/xml_adapter

TYPO3 Configuration

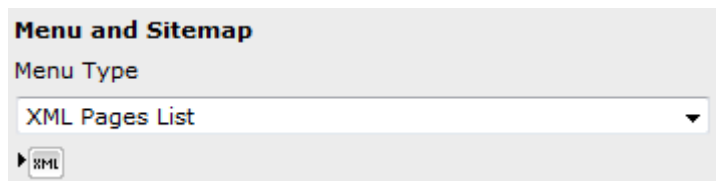
1. Install “xml_adapter”-Extension

Just install the extension and you will now be able to open all content pages as xml by adding “type=777” to your URL. E.g. <http://www.yourpage.com/welcome/?type=777>

It will show you the xml data of this page.

2. Include the sitemap plugin for the XML pages list

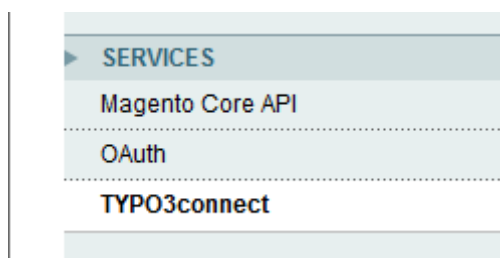
Create a new page in your TYPO3 instance and add a sitemap plugin to it. When you edit the sitemap, choose “XML Pages List” on the Menu and Sitemap Configuration in the “General”-Tab



Magento Configuration

1. Configuration of URL's and typeNum settings.

Go to System – Configuration and find the TYPO3connect Tab under “Services”



XML Adapter		
TYPO3 Base URL	<input type="text" value="http://cms.magedeveloper.de/"/>	[STORE VIEW]
	▲ Please enter the URL of TYPO3 Base (e.g. http://www.mydomain.de/)	
Pages List URL	<input type="text" value="http://cms.magedeveloper.de/other-extensions/xml"/>	[STORE VIEW]
	▲ Please enter the URL of the site, where the XML-Adapter Plugin "Pages List" is included (e.g. http://www.domain.tld/pages/)	
TypeNum	<input type="text" value="777"/>	[STORE VIEW]
	▲ Please enter the TypeNum, where the XML Data is delivered (Standard 777)	

TYPO3 Base URL

Enter the Base URL of your TYPO3 Page URL

Pages List URL

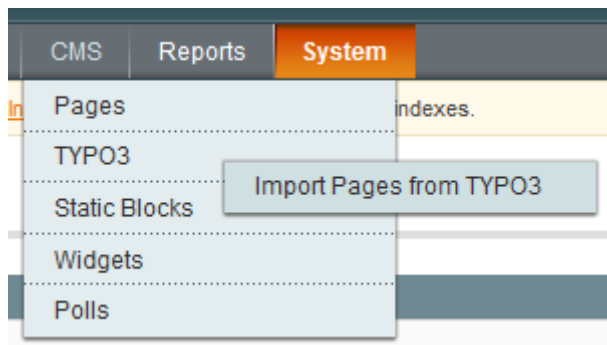
Enter the URL of the TYPO3 Page, where you've added the XML Pages List Sitemap Plugin

TypeNum

Enter the typeNum, where Magento can retrieve the XML Data. **The Default Value is 777!**

Importing Content-Pages to Magento

Go to Magento Admin Panel - CMS – TYPO3 – Import Pages from TYPO3 like shown below:



When Magento has found the XML Pages List, you should see your TYPO3 Page tree on the left side, otherwise check your URL Configuration.

Click within the page tree to receive the page data into Magento. The full page information will appear on the right side. There are several tabs with page information, that you may adjust before the import can start.

A click on the “Import Page”-Button at the top right corner will start importing the page with the content to Magento.



You will find the imported page in CMS – Pages, when the import worked.

A note for developers:

TYPO3 Pages can be loaded directly by the model like shown below:

```
// Page UID in TYPO3
$uid = 81;

// Page Model
$page = Mage::getModel('typo3connect/typo3_page')->loadById($uid);

// Getting XML Content:
$title = $page->getTitle();
$content = $page->getContent();
$description = $page->getDescription();
```

```
<title>cms.magedeveloper.de TYPO3 Testsite</title>
```

Usage of the MAGEconnect Fluid View Helper

Here are some samples how to use the View Helper

```
<f:section name="productsById">
    {namespace mageconnect = Tx_Mageconnect_ViewHelpers}

    <div style="border:1px solid #c0c0c0; padding:5px; margin:5px; width:100%;">
        <strong>productsById ids: '16,18,20,166'</strong><br />
        <f:for each="{mageconnect:productsById(ids: '16,18,20,166', store: '', pid: '75')}}"
as="product">
            <f:then>
                Id {product.entity_id}: {product.name}<br />
            </f:then>
        </f:for>
    </div>
</f:section>

<f:section name="productsByFilter">

    <div style="border:1px solid #c0c0c0; padding:5px; margin:5px; width:100%;">
        <strong>productsByFilter tags: 'Camera', categories: 'Cell Phones', refreshAll: '1',
enableFilterCache: '0'</strong><br />
        <f:for each="{mageconnect:productsByFilter(tags: 'Camera', categories: 'Cell Phones',
store: '', pid: '75', refreshAll: '1', enableFilterCache: '0')}}" as="product">
            <f:then>
                Id {product.entity_id}: {product.name}<br />
            </f:then>
        </f:for>
    </div>
</f:section>

<f:section name="secondProductsByFilter">

    <div style="border:1px solid #c0c0c0; padding:5px; margin:5px; width:100%;">
        <strong>productsByFilter tags: '', categories: 'Cell Phones'</strong><br />
        <f:for each="{mageconnect:productsByFilter(tags: '', categories: 'Cell Phones', store:
'', pid: '75')}}" as="product">
            <f:then>
                Id {product.entity_id}: {product.name}<br />
            </f:then>
        </f:for>
    </div>
</f:section>

<f:section name="thirdProductsByFilter">

    <div style="border:1px solid #c0c0c0; padding:5px; margin:5px; width:100%;">
        <strong>productsByFilter tags: 'Camera', categories: ''</strong><br />
        <f:for each="{mageconnect:productsByFilter(tags: 'Camera', categories: '', store: '',
pid: '75')}}" as="product">
            <f:then>
                Id {product.entity_id}: {product.name}<br />
            </f:then>
        </f:for>
    </div>
</f:section>
```

Always use https!

The data in this module is being encrypted for higher security, but you should really use SSL on the way of both systems!

Change encryption/decryption key after installation

The standard key, that has been entered by us is just for testing, you should really use your own key!

Use a complicated encryption/decryption key

Make it complicated! More letters, more numbers, more special signs. Look at the example key.

The module stage is still beta

It will be continuously advanced but we don't give any advice and are not responsible in any case!

Update your installations

You should always have the last version of these modules, important bugfixes and security fixes could be made!

Remember that products will be first loaded by soap and then cached

To increase stability and performance, the product data will be cached after first load. This means that the first load could be slow, so please be patient on your first setup.

Give a feedback

We always want to improve our work, so please give us a little feedback if you have the time.

The first load is the heaviest

The products will first be loaded by Magento's API. After load, everything will be cached for faster access.

Enable Magento's Cache

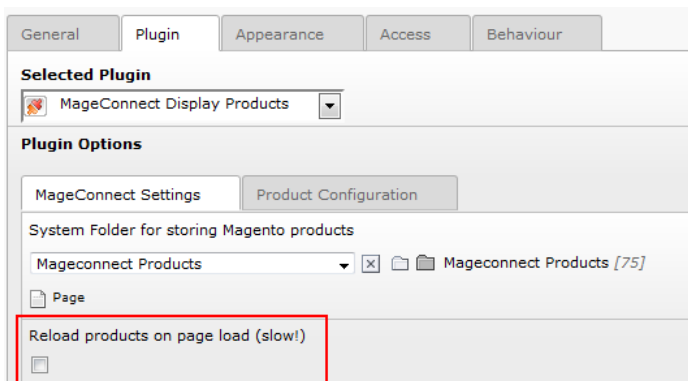
To increase overall speed, you should enable Magento's Cache

Clear the TYPO3 Cache

If you want to reload the product list for the product display plugin, clear the TYPO3 Cache

Reloading products from Magento's API

If you want to reload cached products, just set the option like in the following picture and reload the page in frontend. If you let the checkbox checked, the products will always be loaded from the API.



Execution Time

Product data may load several minutes. It could be necessary to increase the php execution time aswell as the socket timeout.

Please check the following settings in your [php.ini](#):

```
max_execution_time =  
default_socket_timeout =
```

Use your customized Template for the login form

Using your own template for the login would look like this:

```
plugin.tx_mageconnect_pi1 {  
  
    templateLogin = fileadmin/templates/mageconnect_login.html  
  
}
```

Use your customized templates for the product display types and storage pids

To use you own templates you can add following lines to you typoscript

```
plugin.tx_mageconnect_pi2 {  
  
    imagePath = uploads/tx_mageconnect  
  
    ## Storage Pid for the product data  
    productsStoragePid =  
  
    ## Storage Pid for the category data  
    categoriesStoragePid =  
  
    ## Template for Product List  
    templateProductList = fileadmin/templates/mageconnect/product_list.html  
  
    listImage = IMAGE  
    listImage.file.maxW = 90  
    listImage.file.maxH = 90  
  
    ## Template for Product Grid  
    templateProductGrid = fileadmin/templates/mageconnect/product_grid.html  
  
    gridImage = IMAGE  
    gridImage.file.maxW = 187  
    gridImage.file.maxH = 150  
  
    ## Template for Single Product  
    templateProductSingle = fileadmin/templates/mageconnect/product_single.html  
  
    singleImage = IMAGE  
    singleImage.file.maxW = 135  
    singleImage.file.maxH = 100  
  
    ## Template for Product Details  
  
    templateProductDetail = fileadmin/templates/mageconnect/product_detail.html  
  
    detailImage = IMAGE  
    detailImage.file.maxW = 200  
    detailImage.file.maxH = 270  
  
    ## Template for Product Markers  
  
    templateProductMarkers = fileadmin/templates/mageconnect/product_markers.html  
  
}
```

If you set the storagePid and storeViewCode, the settings in the plugin will be ignored!

Use your customized templates for the forgot password form

```
plugin.tx_mageconnect_pi3 {  
    ## Template for MAGEconnect Forgot Password Form  
    templateForgotPassword = EXT:mageconnect/res/forgot_password.html  
}
```

Include magento category tree as navigation**Use customized template for category tree**

```
plugin.tx_mageconnect_pi4 {  
  
    ## Template for MAGEconnect Category Tree  
    templateCategoryTree = EXT:mageconnect/res/category_tree.html  
  
    ## Storage Pid for the category data  
    storagePid =  
  
    ## Store View Code  
    storeViewCode =  
  
}
```

Include category tree as navigation

```
lib.navigation < plugin.tx_mageconnect_pi4  
lib.navigation {  
  
    templateCategoryTree = fileadmin/templates/mysite/category_tree.html  
    storagePid = 123  
    storeViewCode = otherstore  
  
}
```

Single TYPO3 Logout

If you want only the TYPO3 user is being logged out then just add param `single=true` to you logout link, for example:

```
10 = TEXT
10.value = <span>Logout</span>
10.wrap = |
10.typolink.parameter = 1 _self button
10.typolink.additionalParams = &logintype=logout&single=true
```

Using dynamic markers for all imported attributes (including custom) in RTE

If you have an custom attribute in your magento user, you can also access it in TYPO3 with the following lines of TypoScript Code:

```
lib.parseFunc_RTE.nonTypoTagStdWrap.encapsLines.innerStdWrap_all.insertData = 1

### DEBITOR ACCOUNT ###
tt_content.text.12 = LOAD_REGISTER
tt_content.text.12.debitAccount.cObject = USER
tt_content.text.12.debitAccount.cObject.includeLibs = EXT:mageconnect/class.tx_mageconnect_details.php
tt_content.text.12.debitAccount.cObject.userFunc = tx_mageconnect_UserDetails->showUserDetail
tt_content.text.12.debitAccount.cObject.setParameter = debtor_account
tt_content.text.22 = RESTORE_REGISTER
```

The Code above lets you access the data (in this case an attribute called “debtor_account” of the current logged in user.

You can display it in RTE by using this text: {REGISTER:debtorAccount}

Another example for usage:

```
lib.debitAccount = COA
lib.debitAccount {
    10 = USER
    10.includeLibs = EXT:mageconnect/class.tx_mageconnect_details.php
    10.userFunc = tx_mageconnect_UserDetails->showUserDetail
    10.setParameter = debtor_account
}
```

Creating a modal login dialog on every page

The following lines of code will give you a better way to integrate the MAGEconnect Login Form to every page as a modal dialogue that comes up, when the user clicks a login button.

This example uses jquery, so you should add this to your installation, if you want to use the following code.

Add this jQuery Script to your page:

```
jQuery.noConflict();
jQuery(document).ready(function() {

    var errorMsgInUrl = jQuery.getUrlVar('error');
    if (errorMsgInUrl)
    {
        jQuery("#nav_overlay").fadeTo('normal', 0.9);
        jQuery("#login_overlay").fadeIn();
        jQuery("#close").fadeIn();
    }

    jQuery("#nav_overlay").click(function() {
        jQuery("#nav_overlay").fadeOut();
        jQuery("#login_overlay").fadeOut();
        jQuery("#close").fadeOut();
    });

    jQuery(".login").click(function(e) {
        e.preventDefault();
        jQuery("#nav_overlay").fadeTo('normal', 0.9);
        jQuery("#login_overlay").fadeIn();
        jQuery("#close").fadeIn();
    });
    jQuery("#login_overlay #close").click(function() {
        jQuery("#nav_overlay").fadeOut();
        jQuery("#login_overlay").fadeOut();
        jQuery("#close").fadeOut();
    });
});

jQuery.extend({
    getUrlVars: function(){
        var vars = [], hash;
        var hashes = window.location.href.slice(window.location.href.indexOf('?') + 1).split('&');
        for(var i = 0; i < hashes.length; i++)
        {
            hash = hashes[i].split('=');
            vars.push(hash[0]);
            vars[hash[0]] = hash[1];
        }
        return vars;
    },
    getUrlVar: function(name){
        return jQuery.getUrlVars()[name];
    }
});
```

Add this CSS to your page css:

```
#nav_overlay {
    background-color:#fff;
    position:fixed;
    width:100%;
    height:100%;
    z-index:50;
    display:none;
}
#login_overlay {
    position: fixed;
    left:50%;
    top:50%;
    width:660px;
    height:250px;
    border:5px solid #cacaca;
    background-color:#fff;
    color:#000;
    display:none;
    z-index:99999;
    margin-top:-125px;
    margin-left:-330px;
}
#close {
    display:none;
    float: right;
    right:10px;
    margin:-5px 0px 5px 5px;
    font-size:20px;
    cursor:pointer;
    color:#e74300;
    font-weight: bold;
    text-decoration: none;
    position: absolute;
}
```

Add this HTML to your page template after the <body> Tag:

```
<div id="nav_overlay"></div>
<div id="login_overlay">###LOGIN###</div>
```

Fill out the login_overlay with the login plugin of MAGEconnect and use the following template for inclusion.

Add this TypoScript to your page template:

```
lib.login_overlay = RECORDS
lib.login_overlay {
    tables = tt_content
    source = 1234 #Enter the ID of the MAGEconnect Login Plugin
}
```

Add this class to your login button, that you've placed in your template:

```
class=login
```

The jQuery Function binds the class "login" to open the modal dialog. The modal dialog contains individually content. For example place the id of an empty Box made with TemplaVoila on an hidden page with the MAGEconnect Plugin in it and some other stuff.

Add this CSS to your page css to get a hover navigation:

```

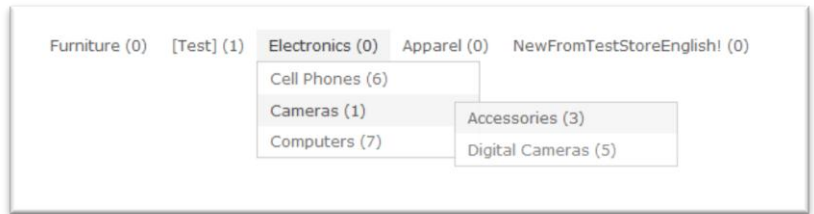
/* Root Level List */
#tx_mageconnect_category_tree ul {
    display: block;
    padding: 0;
    margin: 0;
    list-style: none;
}
#tx_mageconnect_category_tree a {
    display: block;
    text-decoration:none;
    color:#585858;
}
#tx_mageconnect_category_tree ul li ul a {
    width:160px;
    left: -999em;
}
#tx_mageconnect_category_tree :hover > a {
    background-color:#f2f2f2;
    color:#333;
}
#tx_mageconnect_category_tree ul li {
    float: left;
    list-style:none;
}
#tx_mageconnect_category_tree ul li a {
    padding:4px 10px;
}
#tx_mageconnect_category_tree ul li ul {
    position: absolute;
    display:block;
    width:180px;
    background-color: #fff;
    border:1px solid #c0c0c0;
    margin-top:0px;
    left: -999em;
    float:left;
    text-align:left;
    opacity: 0.90;
}
#tx_mageconnect_category_tree li ul ul { /* third-and-above-level lists */
    margin-left:160px;
    margin-top:-20px;
    background-color:#fff;
}

#tx_mageconnect_category_tree li li ul ul { /* third-and-above-level lists */
    background-color:#fff;
    width:140px;
}
#tx_mageconnect_category_tree li ul ul a {
    width:160px;
}

#tx_mageconnect_category_tree li:hover ul ul,
#tx_mageconnect_category_tree li:hover ul ul ul {
    left: -999em;
}

#tx_mageconnect_category_tree li:hover ul,
#tx_mageconnect_category_tree li li:hover ul,
#tx_mageconnect_category_tree li li li:hover ul {
    left: auto;
}

```



2013-07-30 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Feature: View Helper with Product Filter (by Tags/Categories)
- * Bugfix: Missing product markers
- * Bugfix: Wrong URL when no url_path
- * Bugfix: Logout on same page id now possible!
- * Change: Product Currency is now imported in a separate field

2013-05-07 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Feature: Automatic assignment of customer group (if exists)
- * Feature: Fluid View Helper (see manual how to use!)
- * Feature: Auto-Sync same customer attributes (if exist)
- * Bugfix: Backend Caching of large product lists now possible
- * Bugfix: Deleted products will be ignored
- * Bugfix: Wrong plugin text when product source is category
- * Bugfix: Magento Login Redirect Fix (beforeAuthUrl)
- * Bugfix: Better HTML Output on login redirection with template

2013-02-26 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Feature: Category Tree Display
- * Feature: Category Product Display
- * Feature: Single Product Refresh
- * Added Dutch Translation, thanks to Rik
- * Eye-Candy Images for plugin integration

2013-02-20 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- Feature: Category Tree Import
- Feature: Single Product Refresh on Page Load

2013-02-05 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- Feature: Store View Compatibility; See manual for more information!
- Feature: Selectable redirect method
- Bugfix: Redirect Bugfixes
- Bugfix: Language Translations
- Bugfix: Three column layout for loginPost Action

2013-01-03 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Bugfix: Final redirect works now

2012-09-23 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Feature: Import Content Pages from TYPO3 to Magento!
You will need xml_adapter Extension in your TYPO3 instance
- * Bugfix: Remove deprecated t3lib_div::readLLXMLfile

2012-08-02 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Added compatibility to TYPO3 4.7
- * Feature: Dynamic Product Detail View in TYPO3
- * Fixed bug: Empty Images on Magento 1.7
- * Fixed bug: Price/Special Price Import now without tags
- * Fixed bug: Error with Actions in Magento Customer/AccountController
- * Fixed bug: Error decrypting some credential constellations
- * Grid/List CSS improvements
- * Images for display types in BE

2012-06-14 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Added compatibility to Magento 1.7 (please refer to manual!)
- * Added debug possibilities for login procedure
- * Better logout handling/sync

2012-04-04 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Released version 0.5.0
- * Feature: **New Magento Product implementation for TYPO3**
Further information can be found in the manual

2012-03-27 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Feature: New SOAP Request for increasing speed
- * Experimental: Display customer data with dynamic template markers
- * Documentation updated
- * Changed Namespace of Magento TYPO3connect
- * Fixed bug: Final redirect

2012-01-20 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * Released version 0.2.0
- * Feature: Usage of Magento SOAP API V2
- * Fixed bug: Usergroup will not change anymore when user already exists
- * Fixed bug: Changed E-Mail Address Verification (Regex now for .co.uk)
- * Fixed bug: Changed encryption/decryption of data
- * Security: Clearing of POST Parameters
- * Feature: Rewritten Manual

2011-02-21 Bastian Zagar (aixdesign.net) <zagar@aixdesign.net>

- * 0.1.2 Documentation changes, single TYPO3 Logout
- * 0.1.1 initial code