

Caddy – responsive Shopping Cart

Caddy is the TYPO3 responsive Shopping Cart. It is optimised for desktops, tablets and smartphones. Caddy is e-commerce for small budget and small-scale enterprises. You are welcome to install Caddy with one mouse-click - ready-to-use. See: TYPO3 Quick Shop. Caddy provides ready-to-use templates for both responsive frameworks: bootstrap and foundation. You can use caddy for your own needs with your own database and without Quick Shop.



Version: 6.3.1, 2015-12-20

Extension Key: caddy

Language: en

Keywords: caddy, shopping, shop, cart, e-commerce, e-payment, responsive

Copyright 2013-2015, Dirk Wildt, Die Netzmacher, <<http://wildt.at.die-netzmacher.de>>

This document is published under the Open Content License available from <http://www.opencontent.org/opl.shtml>
The content of this document is related to TYPO3 - a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

Caddy – responsive Shopping Cart	1	Individual sum-calculation (by_service_attribute_x_sum)	48
Screenshots	3	Individual max-calculation (by_service_attribute_x_max)	49
Frontend	3	Corporate Design	50
E-Mail	4	Colours	50
Backend	6	Icons	51
Installation	7	E-Mail	52
Introduction	8	HTML Template	52
What does Caddy do?	8	Attachments	53
What doesn't Caddy do?	9	Work-flow	58
What do you need?	9	Add, update and delete an item	58
Target Group	9	Calculating	58
Installation	10	Ordering	58
Quick Installation	10	Variants	60
One-Click-Installation (recommended)	10	TypoScript Snippet	60
Manually Installation (not documented)	12	Non possible features	64
Responsive Design	13	Archive audit proof	65
Screenshots	13	Database is read-only	65
Requirements	13	Disable audit proof	65
Setup	14	Reset numbers	66
E-Payment	16	Reference	67
Screenshots	16	Constant Editor	67
Requirements	17	Setup	67
Setup	17	Debugging	86
Debugging	18	DRS – Development Reporting System	86
PCI-DSS conform	18	Upgrades/Updates	89
Workflow	18	From Caddy 6.x to 6.3	89
Plugins	20	From Caddy 4.x to 6.x	89
Caddy	20	From Caddy 2.x/3.x to 4.x	89
Mini Caddy	22	From Powermail 1.x to Powermail 2.x	89
Sum Gross	24	From wt_cart 1.x to Caddy 2.x	89
Controlling Caddy	25	What's new?	90
Overview	25	Version 6	90
Plugin / Flexform	26	Version 4	90
Static Templates	32	Packages for Sponsors	93
Constant Editor	33	Fun – 50 EUR	93
Constant Editor	34	Logo – 150 EUR	93
MAIN	34	Patron – 500 EUR	93
BANKACCOUNT	34	Credits	94
DATABASE	35	Made in Nature	94
DEBUG	35	wt_cart Developer Team	94
E-PAYMENT	35	Known Bugs	95
GETPOST	36	PDF Attachments without any address data	95
HTML	37	Required argument "mail" is not set	95
Options	38	TCPDF-Error	95
PAGES	39	More Bugs	95
PDF	40	Further Information	96
SYMBOLS AND SERVICE ATTRIBUTES	42	Forum	96
TAX	42	TYPO3 forge	96
TEMPLATES	43	History of Caddy	96
URL	43	Caddy 2.x versus wt_cart 1.x	97
Calculating	44	Other extensions published by Die Netzmacher	98
Default code	44	Change Log	99
Based on price (by_price)	45	Illustration Index	105
Based on quantity (by_quantity)	46	Alphabetical Index	106
Based on each (each)	47		

Screenshots

Frontend

Responsive

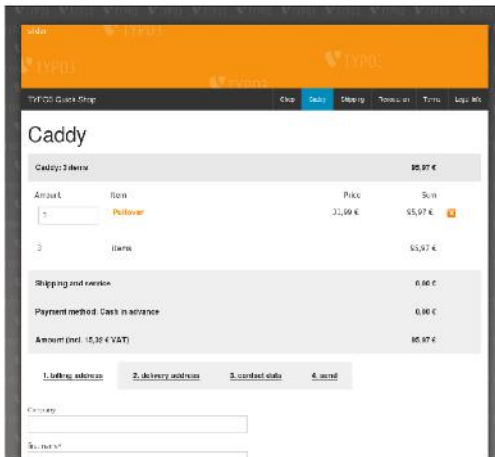


Illustration 4: Caddy on the desktop

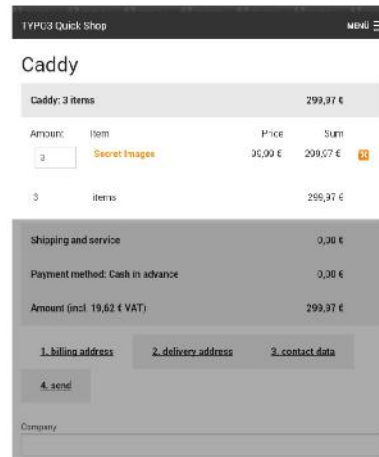


Illustration 5: Caddy with an opened accordion on the smartphone

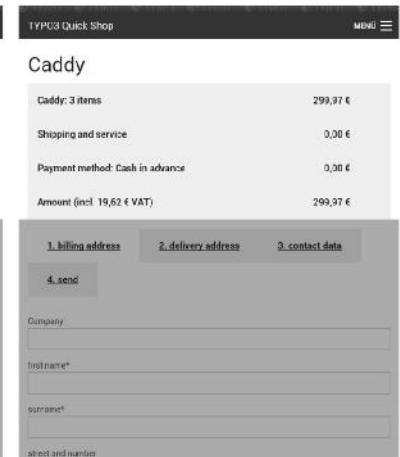


Illustration 6: Caddy with a closed accordion on the smartphone

Illustration 1: Caddy on a desktop and on a smartphone

Live

- <http://typo3-quick-shop.de/warenkorb/>

E-Payment

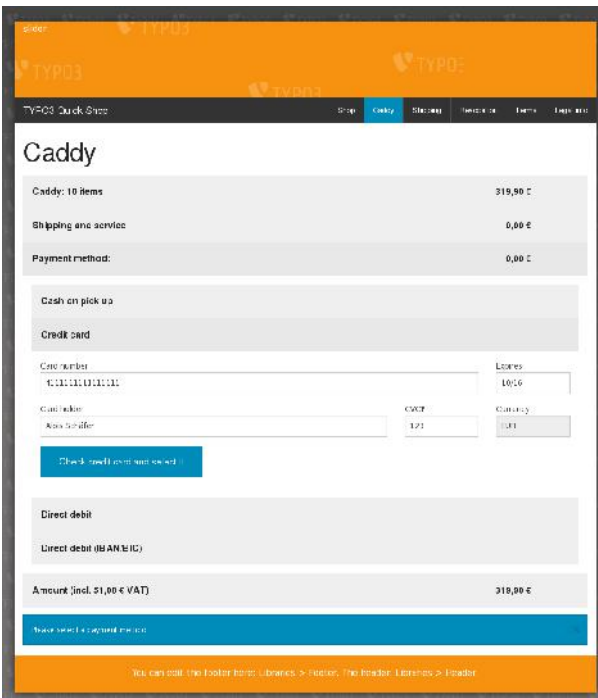


Illustration 2: Caddy with e-payment interface

You can enable an e-payment interface. See requirements and details at "E-Payment" on page 16 below.

E-Mail

Confirmation E-Mail

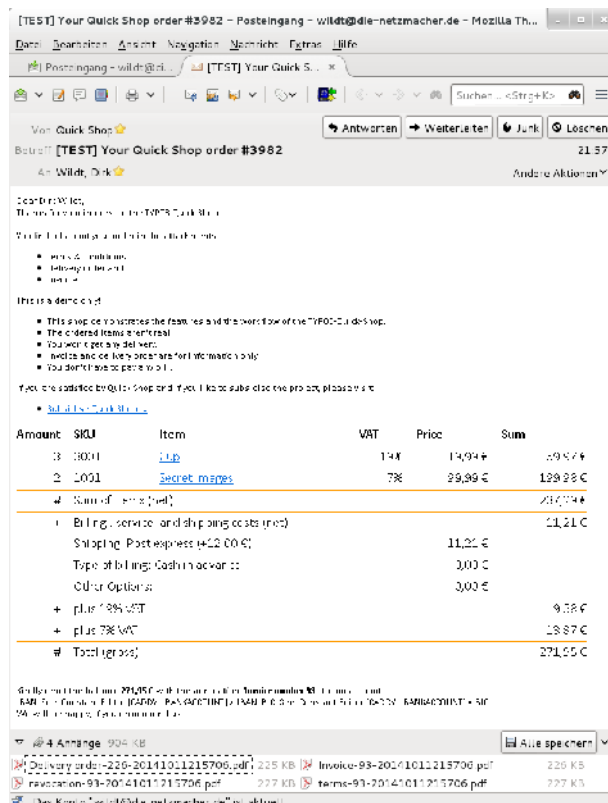


Illustration 3: Confirmation e-mail

The confirmation e-mail contains:

1. a text
2. the caddy
3. attachments

All is configurable of course.

Attachments



Illustration 4: The caddy e-mail attachments (here: Quick Shop)

Attachments by default are

- the invoice,
- the delivery order
- the power of revocation and
- the terms and conditions.

The corporate design of the attachments is configurable. See

- "Attachments" on page 53 below.

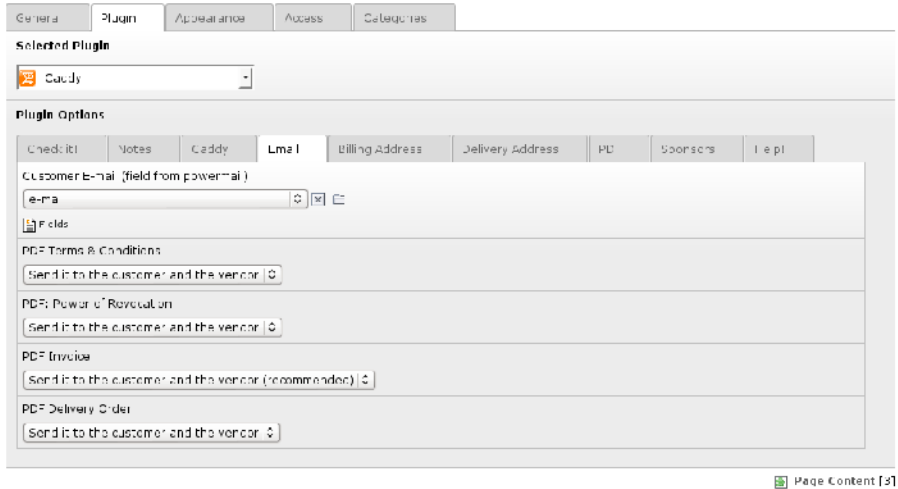
Customer and vendor can get different attachments. See

- "Plugin / Flexform > PDF" on page 30 below.

Backend

Plugin

Edit Page Content "Caddy" on page "Caddy"



☒ Show secondary options (palettes)

Illustration 5: The caddy plugin.

Caddy has a powerful plugin.

The plugin contains 9 tabs.

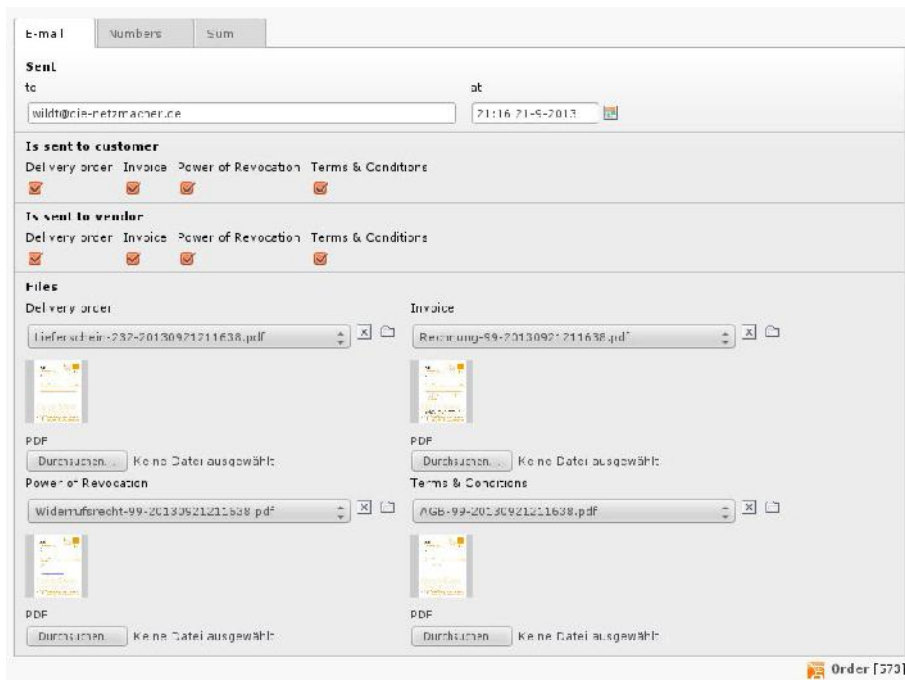
The plugin evaluate the configuration of

- the caddy plugin,
- the caddy TypoScript and
- the powermail plugin.

If there is a configuration missing or it is improper, you will get a prompt within the first tab [Check it!]

See details at "Plugin / Flexform" on page 26 below.

Order Records



The screenshot shows the 'Order [573]' record in the Caddy system. It includes a header with 'e-mail', 'Numbers', and 'Sum' tabs. The main content area is divided into sections for 'Sent to customer' and 'Sent to vendor', each with checkboxes for 'Delivery order', 'Invoice', 'Power of Revocation', and 'Terms & Conditions'. Below these are 'Files' sections for both customer and vendor, listing various PDF attachments like 'Lieferung-237-20130921211638.pdf' and 'Rechnung-89-20130921211638.pdf'. Each file entry has a 'Durchsuchen...' button and a status 'Keine Datei ausgewählt'.

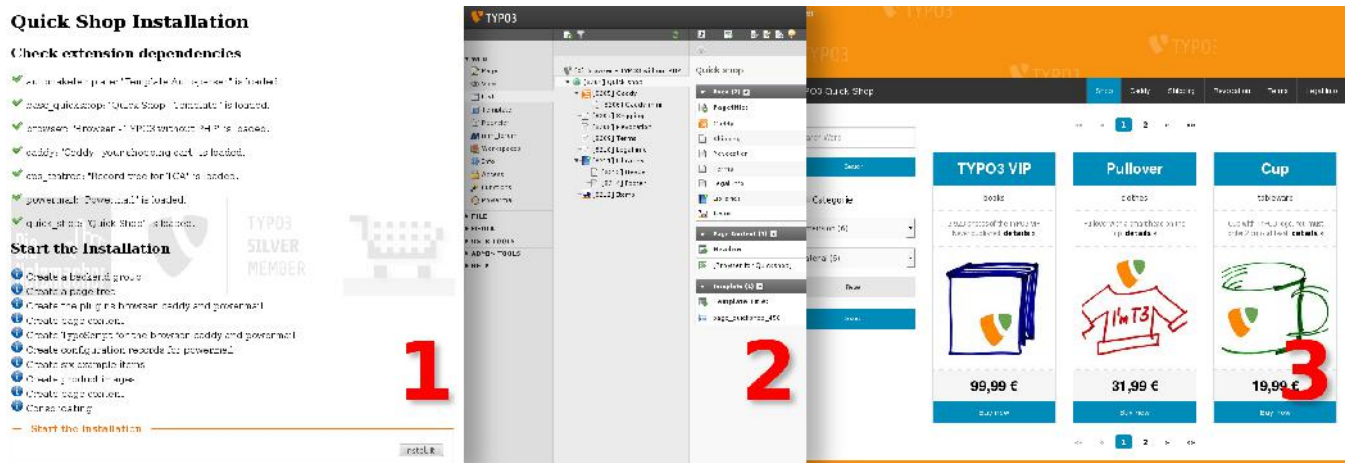
Illustration 6: A caddy order record

Caddy archives each order in a database audit proof.

An order record contains among others

- who got when the confirmation e-mail
- which attachments are sent
 - to the customer and
 - to the vendor
- the sent attachments.

Installation



The screenshot shows the 'Quick Shop Installation' process. It includes a 'Check extension dependencies' section with a list of extensions and their status. Below this is the 'Start the Installation' section with a list of steps. A large red number '1' is overlaid on the 'Start the Installation' button. To the right, a large red number '2' is overlaid on the 'Quick Shop' button. Further right, a large red number '3' is overlaid on the 'Caddy' button. The background shows a preview of the 'TYPO3 Quick Shop' interface with various product cards like 'TYPO3 VIP', 'Pullover', and 'Cup'.

Illustration 7: The Quick Shop Installer enables you, to install Caddy ready-to-use (2+3) with one mouse click (1).

Introduction

What does Caddy do?

- Caddy is a shopping cart for TYPO3 extensions.
- You can use Caddy for selling items, for reserving tickets or for ordering documents among others.
- Caddy has features like
 - PDF attachments in the e-mail to the customer and the vendor like
 - delivery order, invoice, power of revocation and terms and conditions. See "Attachments" on page 53.
 - minimum and maximum limitations
 - for each item and for all items
 - delivery costs depending on
 - quantity, size, weight
 - variants (one dimension only)
 - a mini cart
- Responsive
 - Caddy has both designs: the default and the responsive design.
 - Caddy provides ready-to-use templates for both responsive frameworks: bootstrap and foundation.
 - See "Responsive Design" on page 13 below.
- E-payment
 - Caddy has both payment interfaces: the default- and the e-payment-interface. See "E-Payment" on page 16.
- Caddy archives each order in a database audit proof.
- Caddy is using powermail for
 - creating order forms and confirmation forms
 - sending e-mails to the customer and the vendor
- Caddy evaluates the self-configuration and the needed Powermail properties.
- You can configure the colours of Caddy and Powermail by a user interface.
 - Caddy has ready-to-use templates for the colours
 - blue, green, orange, red.
 - Caddy is delivered with a lot of icons.
- You can test Caddy in your own TYPO3 instance with one mouse click!
 - The Organiser Installer installs a ready-to-use website with two caddies: one for reserving tickets, the other for ordering documents.
 - The Quick Shop Installer installs a ready-to-use online shop with TYPO3 sample items.
 See "One-Click-Installation (recommended)" on page 10 below.
- Caddy has a powerful debugging:
 - The DRS – the Development Reporting System – prompts the whole work flow to the backend.
 - Another modul prompts TypoScript properties to the frontend.

What doesn't Caddy do?

Caddy doesn't support features like

- cross selling and
- vouchers

What do you need?

- Powermail (version 1.x or 2.x¹)
- t3_tcpdf
- TYPO3 Introduction Package or another Bootstrap framework, if you like to Caddy in a responsive design supported by Bootstrap.
- TYPO3 Foundation or another Foundation framework, if you like to Caddy in a responsive design supported by Foundation.
- Caddy is tested in TYPO3 versions 4.5, 4.6, 4.7, 6.1 abd 6.2

Target Group

- TYPO3 agencies
- TYPO3 integrators
- TYPO3 developers

¹ If you are using TYPO3 6.x, powermail version must be 2.x because 1.x version doesn't work with different languages with 6.1. Powermail 2.x seems to be compulsory if the site is not in English.

Installation

Quick Installation

- Install Caddy
- Add a page "Caddy" to your page tree:
 - Add a TypoScript extension template.
 - Add a Caddy plugin.
 - Add a Powermail plugin.
- Configure Caddy supported by the Constant Editor.
- Configure Powermail:
 - Add the needed fields.
 - Add the needed fieldsets.
 - Configure the plugin.
- Configure Caddy by the plugin.

One-Click-Installation (recommended)

The one-click-installation is recommended¹.

Advantages

The advantages of the one-click-installations are:

- You will have a minimum of costs for the installation and the configuration:

You will need

 - the first time 30 minutes over all and
 - all next times 15 minutes over all.
- You will get a running sample with caddy and without
 - any unproper configuration and
 - any costs for debugging.
- A running sample is the best for understanding.
- You can test all features of caddy immediately. Play with Caddy!
- If something goes wrong and you can't cancel it, it doesn't matter.
 - Waste your installation.
 - Repeat the one-click-installation.

Work flow for your own needs

- The one-click-installation offers three Caddy samples:
 - for ordering documents²
 - for reserving tickets³
 - for selling items⁴
- If no one of the three samples doesn't match your needs,
 - waste Caddy,
 - look for another TYPO3 extension, which could match your needs, or

¹ The one-click-installation is obligated only, because there isn't any manual for a manual installation.

² See "Organiser Installer" on page 11 below

³ See "Organiser Installer" on page 11 below

⁴ See "Quick Shop Installer" on page 11 below

- develop your own extension.
- If one of the three samples matches your needs,
 - you can move the page with the Caddy within the page tree to your preferred position, and
 - you have to adapt Caddy with the Constant Editor to your needs.

Organiser Installer

The Organiser Installer installs a ready-to-use Organiser. The Organiser is TYPO3 for the lobby and for the organisers.

The Organiser supports the publishing of documents, events, headquarters, locations, news and staff.

The Organiser contains Caddy twice:

- for reserving tickets for events
- for ordering documents

See

- Organiser <http://typo3-organiser.de/en/>
- Organiser Installer¹ <http://typo3-organiser.de/en/one-click/>

Quick Shop Installer

Quick Shop Installation

Check extension dependencies

- ✓ ext_key: caddy (TYPO3 Extension) is loaded.
- ✓ ext_key: quickshop (Quick Shop - TYPO3) is loaded.
- ✓ ext_key: TYPO3 (TYPO3) is loaded.
- ✓ ext_key: Caddy (Caddy - TYPO3) is loaded.
- ✓ ext_key: Caddy (Caddy - TYPO3) is loaded.
- ✓ ext_key: Caddy (Caddy - TYPO3) is loaded.
- ✓ ext_key: Caddy (Caddy - TYPO3) is loaded.
- ✓ ext_key: Caddy (Caddy - TYPO3) is loaded.

Start the Installation

- 1 Create a backend group
- 2 Create a page tree
- 3 Create the page tree between caddy and government
- 4 Create page content
- 5 Create TypoLink to page between caddy and government
- 6 Create outgoing links records for powered
- 7 Create outgoing links
- 8 Create outgoing links
- 9 Create outgoing links
- 10 Create outgoing links
- 11 Create outgoing links
- 12 Create outgoing links
- 13 Create outgoing links
- 14 Create outgoing links
- 15 Create outgoing links
- 16 Create outgoing links
- 17 Create outgoing links
- 18 Create outgoing links
- 19 Create outgoing links
- 20 Create outgoing links
- 21 Create outgoing links
- 22 Create outgoing links
- 23 Create outgoing links
- 24 Create outgoing links
- 25 Create outgoing links
- 26 Create outgoing links
- 27 Create outgoing links
- 28 Create outgoing links
- 29 Create outgoing links
- 30 Create outgoing links
- 31 Create outgoing links
- 32 Create outgoing links
- 33 Create outgoing links
- 34 Create outgoing links
- 35 Create outgoing links
- 36 Create outgoing links
- 37 Create outgoing links
- 38 Create outgoing links
- 39 Create outgoing links
- 40 Create outgoing links
- 41 Create outgoing links
- 42 Create outgoing links
- 43 Create outgoing links
- 44 Create outgoing links
- 45 Create outgoing links
- 46 Create outgoing links
- 47 Create outgoing links
- 48 Create outgoing links
- 49 Create outgoing links
- 50 Create outgoing links
- 51 Create outgoing links
- 52 Create outgoing links
- 53 Create outgoing links
- 54 Create outgoing links
- 55 Create outgoing links
- 56 Create outgoing links
- 57 Create outgoing links
- 58 Create outgoing links
- 59 Create outgoing links
- 60 Create outgoing links
- 61 Create outgoing links
- 62 Create outgoing links
- 63 Create outgoing links
- 64 Create outgoing links
- 65 Create outgoing links
- 66 Create outgoing links
- 67 Create outgoing links
- 68 Create outgoing links
- 69 Create outgoing links
- 70 Create outgoing links
- 71 Create outgoing links
- 72 Create outgoing links
- 73 Create outgoing links
- 74 Create outgoing links
- 75 Create outgoing links
- 76 Create outgoing links
- 77 Create outgoing links
- 78 Create outgoing links
- 79 Create outgoing links
- 80 Create outgoing links
- 81 Create outgoing links
- 82 Create outgoing links
- 83 Create outgoing links
- 84 Create outgoing links
- 85 Create outgoing links
- 86 Create outgoing links
- 87 Create outgoing links
- 88 Create outgoing links
- 89 Create outgoing links
- 90 Create outgoing links
- 91 Create outgoing links
- 92 Create outgoing links
- 93 Create outgoing links
- 94 Create outgoing links
- 95 Create outgoing links
- 96 Create outgoing links
- 97 Create outgoing links
- 98 Create outgoing links
- 99 Create outgoing links
- 100 Create outgoing links

Start the Installation

1

2

3

Illustration 8: Quick Shop Installer - Installation with one mouse click

The Quick Shop Installer (1) installs a ready-to-use Quick Shop (2, 3) with one mouse click.

The Quick Shop is a small online shop. It sells TYPO3 items by default like TYPO3 clothes, cups and books. You can use Quick Shop for your own items.

The Quick Shop contains Caddy once:

- for selling items.

See

- Quick Shop <http://typo3-quick-shop.de/en/>
- Quick Shop Installer² <http://typo3-quick-shop.de/en/one-click/>



¹ You need the Organiser Installer from version 3.3. I purpose to publish it in September of 2013.

² You need the Quick Shop Installer from version 3.3. I purpose to publish it in September of 2013.

Manually Installation (not documented)

The manually installation of Caddy is complex and isn't recommended. The cause is:

- Caddy isn't a shop.
- Caddy is a shopping cart only.

Because Caddy is a shopping cart only, Caddy has interfaces for third party extensions for

- managing items (done by the Quick Shop, the Organiser or your extension),
- interactive actions with the customer (done by Powermail) and
- for the responsive design.

A manually installation has a lot of possibilities for an improper configuration.

And the manual for a manually installation has a lot of possibilities for improper or missing statements too.

I would be happy, if you are appreciative of a missing documentation for a manually installation.

Responsive Design

Screenshots

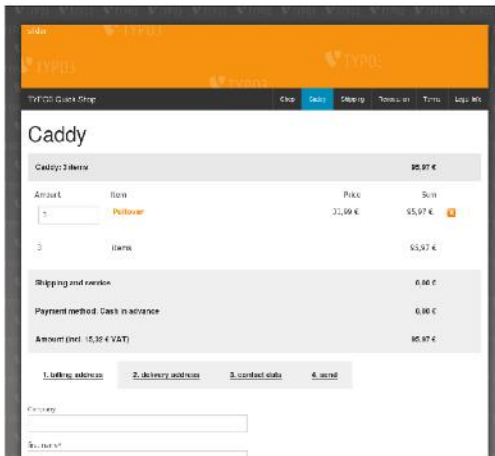


Illustration 4: Caddy on the desktop

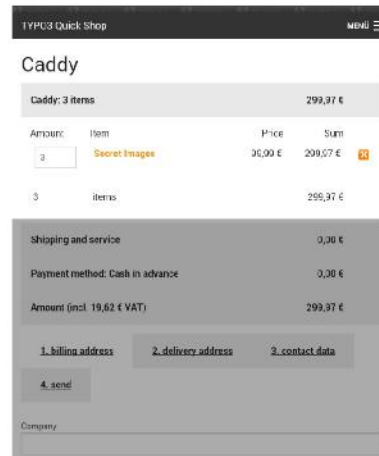


Illustration 5: Caddy with an opened accordion on the smartphone

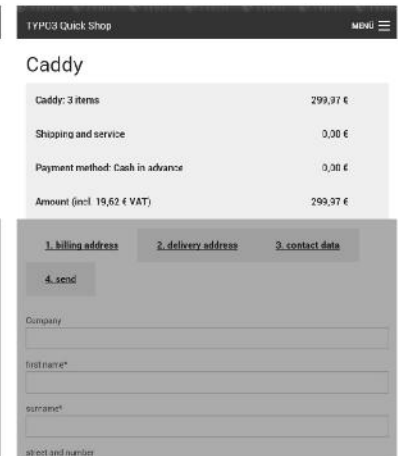


Illustration 6: Caddy with a closed accordion on the smartphone

Illustration 9: Caddy on a desktop and on a smartphone

Contracted Features

The development of the responsive design was contracted for

- the responsive frameworks Bootstrap 3.x, Foundation 4.x, 5.x and
- caddy with Powermail 2.x

Requirements

Powermail 2.x

Caddy is a team-play with Powermail 1.x or 2.x. If you like to use caddy responsive, you have to use Powermail 2.x.

TYPO3 6.x

Caddy fits Powermail 2.x in both responsive designs: Bootstrap and Foundation. The modification isn't compatible with TYPO3 4.x. You need TYPO3 6.x.

TYPO3 Foundation

If you don't know anything about Foundation or if you don't like to install and configure Foundation manually, you are welcome to use the extension TYPO3 Foundation (t3foundation). The extension has ready-to-use templates for TYPO3.

You don't need to know anything about this, if you want to use Foundation for Caddy only. But if you are interested in detailed information, please take a look here:

- PDF manual: <http://typo3-foundation.de/typo3conf/ext/t3foundation/doc/manual.pdf>
- typo3.org: <http://typo3.org/extensions/repository/view/t3foundation/>

Setup

If you are using TYPO3 Quick Shop or TYPO3 Organiser, it is recommended to upgrade it to version 6.x. 4.x is running too, but you have to maintain your CSS for caddy.

Caddy

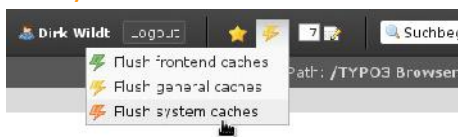


Illustration 10: Flush System Cache

You have to include static templates (TypoScript).

You must flush the system caches before, else updated templates won't displayed in the backend.

If you don't have the menu item [Flush system caches] like in the illustration above, please install the extension TSconfig Pages and Users by extManager (extkey: tsconf) and save the default configuration in the extension manager. Reload your backend, your cache menu will contain the item [Flush system caches].

Sample page tree

- **TYPO3**
 - Quick Shop [1] <- or your Shop
 - Caddy [2]

Page with the Shop

This sample is based on Quick Shop.

If you are using your own Shop, take the include-static-templates of your own shop.

Include static (from extensions):

- CSS Styled Content (css_styled_content)
- Browser [0] + Foundation Framework (browser)
- Browser [1] Basis (browser)
- Browser [5] + Foundation Templates (browser)
- Caddy [1] Basis (caddy)
- Caddy [2] + CSS (blue) (caddy) only without Foundation¹
- Caddy [2.1] + orange (caddy)
- Caddy [3] + Foundation 5.x (caddy) only with Foundation²
- Caddy [3.1] + Foundation 5.x CSS (caddy) only with Foundation³
- Caddy [7] + Language: German (caddy)
- Quick Shop [0] + Template (base_quickshop)
- Quick Shop [1] Basis (quick_shop)
- Quick Shop [2] + Caddy (quick_shop)

You have to setup the table of your database . See Constant Editor [CADDY – DATABASE] > Table.

¹ Without "Browser [0] + Foundation Framework (browser)"

² With "Browser [0] + Foundation Framework (browser)" or with any other Foundation Framework

³ dito

Please check the page ids. See Constant Editor [CADDY – PAGES]

Page with Caddy

Include static (from extensions):

- Main Template (powermail)
- Add Demo CSS (powermail)
- Caddy [5] Powermail 2.x (caddy)
- Caddy [5.1] + Powermail 2.x CSS fancy (caddy)

or if you have the responsive foundation framework installed:

- Main Template (powermail)
- Caddy [5] Powermail 2.x (caddy)
- Caddy [6] + Powermail 2.x Foundation 5.x (caddy)

E-Payment

Caddy has an interface for e-payment. You need a contract with an e-payment provider of course.

Caddy is supporting the e-payment provider Paymill in the current version only.

The interface is responsive.

Screenshots

Paymill

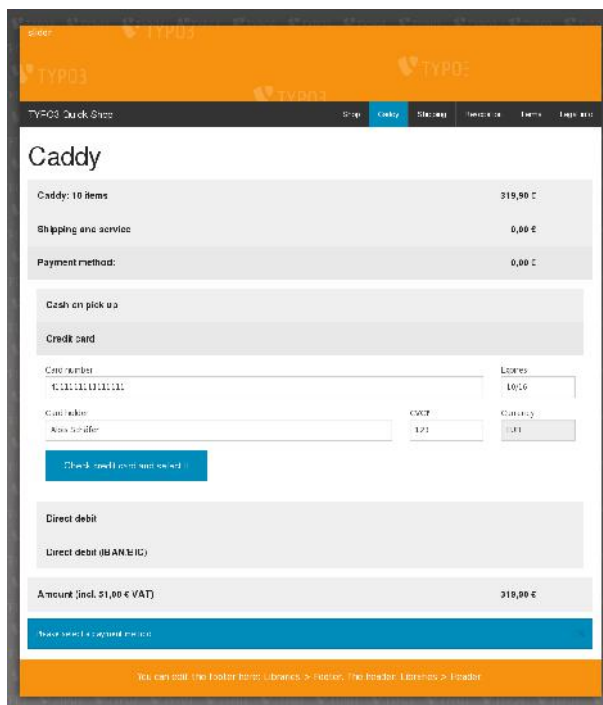


Illustration 11: Caddy e-payment (Paymill) in live mode

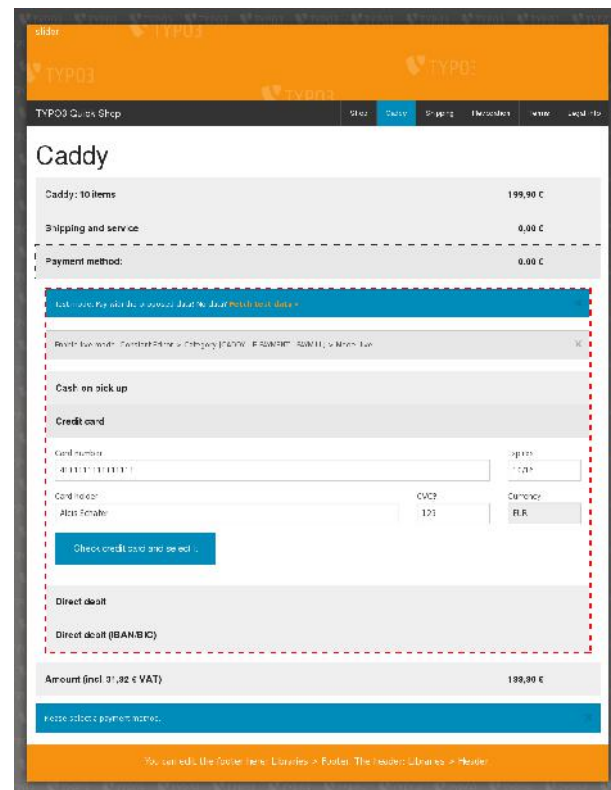


Illustration 12: Caddy e-payment (Paymill) in test mode

Paymill Transaction

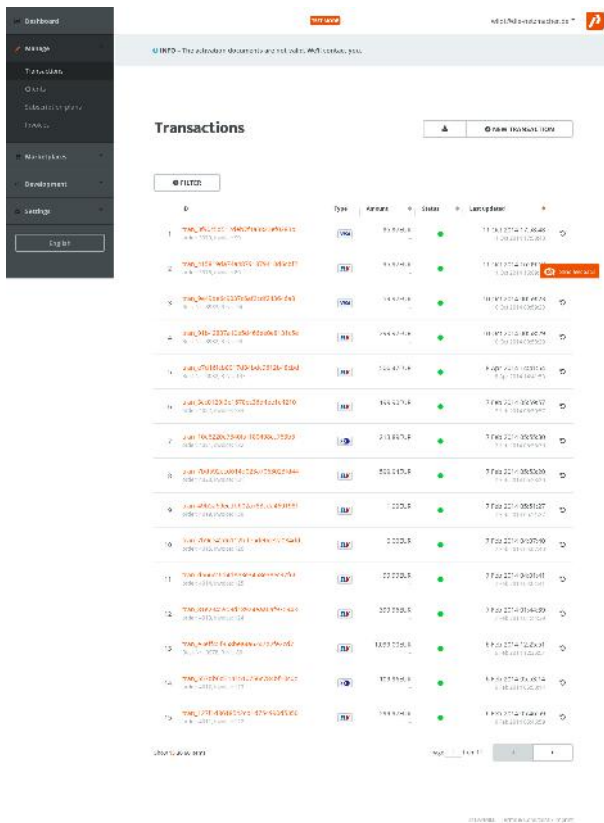


Illustration 13: Transactions (Paymill test mode)

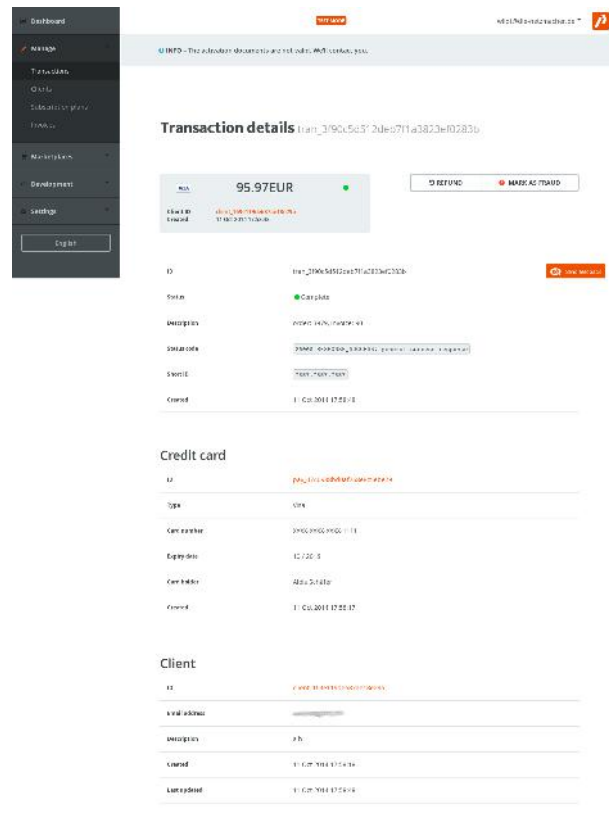


Illustration 14: Transaction (Paymill test mode)

Requirements

Contract

A contract with an e-payment provider (here: Paymill – see below)

PHP

The PHP class curl must be enabled.

Responsive

The e-payment form is responsive. Sorry, but there isn't any other HTML template and any other TypoScript template prepared. Thus you have to

- include the static templates for the responsive design.

See "Responsive Design" on page 13 above.

Setup

Include the e-payment template

Please add the e-payment template to the TypoScript of your page

Include static (from extensions):

- Main Template (powermail)
- Caddy [5] Powermail 2.x (caddy)
- Caddy [6] + Powermail 2.x Foundation 5.x (caddy)
- Caddy [8] + E-Payment: Paymill Foundation 5.x (caddy)

Configure Paymill

After including the e-payment template, you have a lot of new options for configuration. See

- TypoScript Constant Editor > Category: [CADDY – E-PAYMENT – PAYMILL*]

Please add your private and public keys.

Test- and Live-Mode

Paymill is running in test mode by default. You can test transactions without any real payment.

If you like to switch to the live mode, please configure

- TypoScript Constant Editor > Category: [CADDY – E-PAYMENT – PAYMILL*] > Mode

See screenshots with test- and live-mode at "Screenshots" on page 16 above.

Constant Editor

See "E-PAYMENT – PAYMILL" on page 35 below.

Debugging

You should enable all debugging properties:

- Extension manager
- Plugin
- TypoScript

See "Debugging" on page 86 below.

PCI-DSS conform

The e-payment interface is conform with Payment Card Industry Data Security Standard (PCI DSS). PCI DSS is in place to make the internet safer and more secure by ensuring that the sensitive data of your customers is handled in a recognised and secure manner.

PAYMILL supports merchants to be PCI DSS compliant. Your clients' sensitive data is sent directly to PAYMILL. The sensitive data never touches your servers and this drastically reduces the scope of your compliance requirements.

See details in the section Workflow below.

Workflow

Any knowledge about the e-payment workflow is needed. But if you are interested, read the section below.

Paymill

Caddy handles e-payment in the two steps:

- Evaluation
- Transaction

In detail:

1. data for e-payment are evaluated by JavaScript¹
 1. data will send to Paymill
 2. if data are proper, Paymill sends back a token
 3. if data aren't proper, Paymill tries to send a qualified error prompt or Caddy tries to generate a qualified error prompt.
2. if data for e-payment are proper, the customer can send the order to your server
 1. the order doesn't contain any e-payment account data but the Paymill token²
 2. your server sends the token and some other data like the amount and the invoice number to the server of Paymill for transaction

¹ Caddy is using the PAYMILL Javascript Bridge. The clients' sensitive data is sent directly to PAYMILL and is never touches your servers.

² dito

3. if the transaction is successful, caddy
 1. sends the confirmation e-mail to the shop owner and a copy to the customer
 2. stores some order data in the database
4. if the transaction fails, caddy
 1. outputs a HTML error page with a qualified prompt
 2. the customer can go back to the order
3. if data for e-payment aren't proper, the customer can't send the order to your server

Plugins

Caddy

Workflow

The screenshot shows the 'Caddy' plugin interface with a table of items. The table has columns for 'Amount', 'Item', 'Price', and 'Sum'. The items are 'Cup' and 'Secret images'. Below the table, there are sections for 'Shipping and service', 'Payment method: Cash in advance', and 'Amount (incl. 23,45 € VAT)'. At the bottom, there are four tabs: '1. billing address', '2. delivery address', '3. contact data', and '4. send'.

Amount	Item	Price	Sum
1	Cup	19,99 €	19,99 €
2	Secret images	99,99 €	199,98 €
3	Shipping		259,95 €

Shipping and service: 12,00 €

Payment method: Cash in advance: 0,00 €

Amount (incl. 23,45 € VAT): 271,55 €

1. billing address 2. delivery address 3. contact data 4. send

Illustration 15: Caddy with accordion items

The screenshot shows the 'Caddy' plugin interface with a table of items. The table has columns for 'Amount', 'Item', 'Price', and 'Sum'. The items are 'Cup' and 'Secret images'. Below the table, there are sections for 'Shipping and service', 'Payment method: Cash in advance', and 'Amount (incl. 23,45 € VAT)'. At the bottom, there are four tabs: '1. billing address', '2. delivery address', '3. contact data', and '4. send'.

Amount	Item	Price	Sum
1	Cup	19,99 €	19,99 €
2	Secret images	99,99 €	199,98 €
3	Shipping		259,95 €

Shipping and service: 12,00 €

Payment method: Cash in advance: 0,00 €

Amount (incl. 23,45 € VAT): 271,55 €

1. billing address 2. delivery address 3. contact data 4. send

Illustration 16: Caddy with accordion shipping and service

The screenshot shows the 'Caddy' plugin interface with a table of items. The table has columns for 'Amount', 'Item', 'Price', and 'Sum'. The items are 'Cup' and 'Secret images'. Below the table, there are sections for 'Shipping and service', 'Payment method: Cash in advance', and 'Amount (incl. 23,45 € VAT)'. At the bottom, there are four tabs: '1. billing address', '2. delivery address', '3. contact data', and '4. send'.

Amount	Item	Price	Sum
1	Cup	19,99 €	19,99 €
2	Secret images	99,99 €	199,98 €
3	Shipping		259,95 €

Shipping and service: 12,00 €

Payment method: Cash in advance: 0,00 €

Amount (incl. 23,45 € VAT): 271,55 €

1. billing address 2. delivery address 3. contact data 4. send

Illustration 17: Caddy with accordion payment

The screenshot shows the 'Caddy' plugin interface with a table of items. The table has columns for 'Amount', 'Item', 'Price', and 'Sum'. The items are 'Cup' and 'Secret images'. Below the table, there are sections for 'Shipping and service', 'Payment method: Cash in advance', and 'Amount (incl. 23,45 € VAT)'. At the bottom, there are four tabs: '1. billing address', '2. delivery address', '3. contact data', and '4. send'.

Amount	Item	Price	Sum
1	Cup	19,99 €	19,99 €
2	Secret images	99,99 €	199,98 €
3	Shipping		259,95 €

Shipping and service: 12,00 €

Payment method: Cash in advance: 0,00 €

Amount (incl. 23,45 € VAT): 271,55 €

1. billing address 2. delivery address 3. contact data 4. send

Illustration 18: Caddy with accordion amount

The screenshot shows the 'Caddy' plugin interface with a table of items. The table has columns for 'Amount', 'Item', 'Price', and 'Sum'. The items are 'Cup' and 'Secret images'. Below the table, there are sections for 'Shipping and service', 'Payment method: Cash in advance', and 'Amount (incl. 23,45 € VAT)'. At the bottom, there are four tabs: '1. billing address', '2. delivery address', '3. contact data', and '4. send'.

Amount	Item	Price	Sum
1	Cup	19,99 €	19,99 €
2	Secret images	99,99 €	199,98 €
3	Shipping		259,95 €

Shipping and service: 12,00 €

Payment method: Cash in advance: 0,00 €

Amount (incl. 23,45 € VAT): 271,55 €

1. billing address 2. delivery address 3. contact data 4. send

Illustration 19: Caddy thanks

Backend

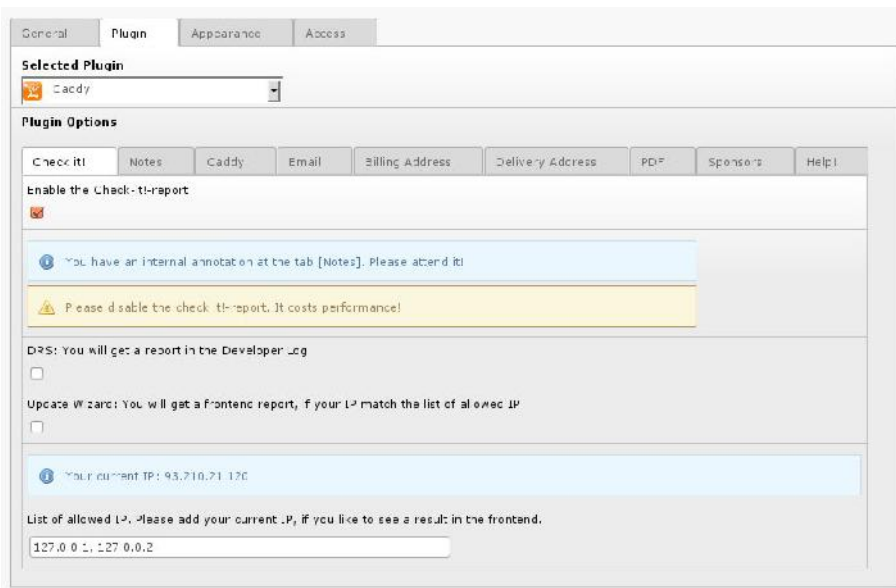


Illustration 20: Plugin Caddy (backend)

Controlling

See details at

- "Controlling Caddy" on page 25 below.

Mini Caddy

Screenshots



Illustration 21: Desktop: the icon, if caddy contains items.

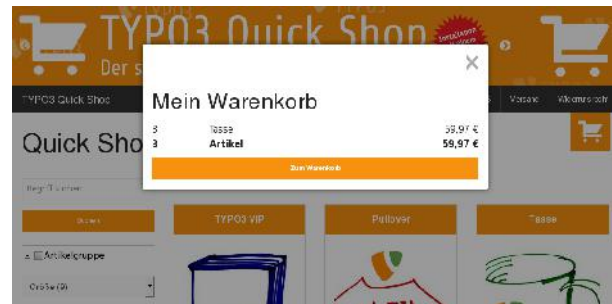


Illustration 22: Click on the mini caddy icon opens a dialog box.

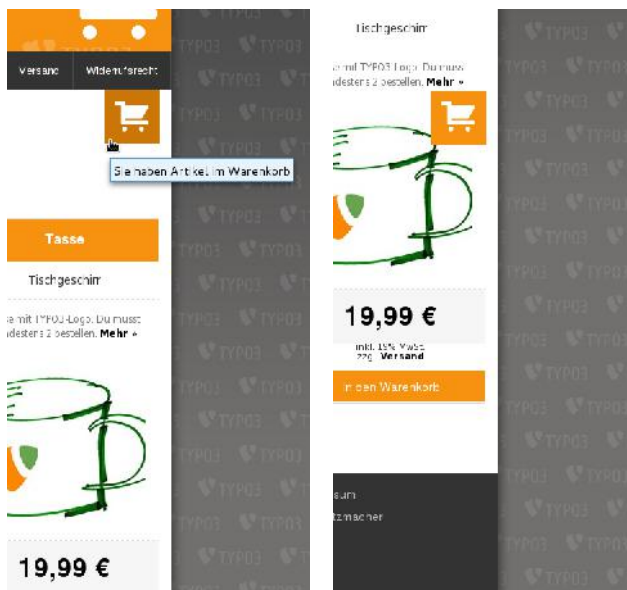


Illustration 23: The position of the icon is fixed.



Illustration 24: Smartphone: Caddy without mini caddy



Illustration 25: The icon, if caddy contains items.



Illustration 26: The position of the icon is fixed



Illustration 27: The dialog box takes the full screen.

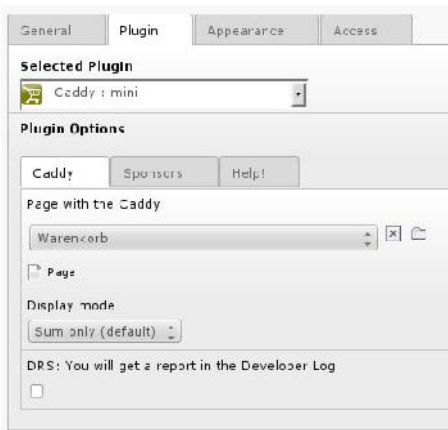


Illustration 28: Interface of the mini caddy

Controlling

- Add the Plugin "Caddy: Mini" to your page.
- Configure the Plugin:
 - Enter the page with the Caddy plugin.
 - Display mode:
 - Items and sum
 - Sum only (default)

TypoScript

You have to include the static template:

- Caddy [3.2] + Foundation Minicaddy Reveal (caddy)

Caddy is using the foundation icons. If you don't have included it, please

- add the extension "Starter Kit Responsive" (start) and
- include the static template
Start [3] Foundation Icons 3 (start)

Snippet

Snippet for displaying the mini caddy website wide:

```
// minicaddy in top of the content
lib.content = COA
lib.content {
    // Content of the page with the plugin "Caddy: Mini"
    10 < styles.content.get
    10 {
        select.pidInList = {$plugin.caddy.pages.caddymini}
    }
    // Content of the current page
    20 < styles.content.get
}
```

Setup

See "Setup > tx_caddy_pi2" on page 79 below.

Sum Gross

Documentation for Caddy 4.x – maybe proper for version 6.x

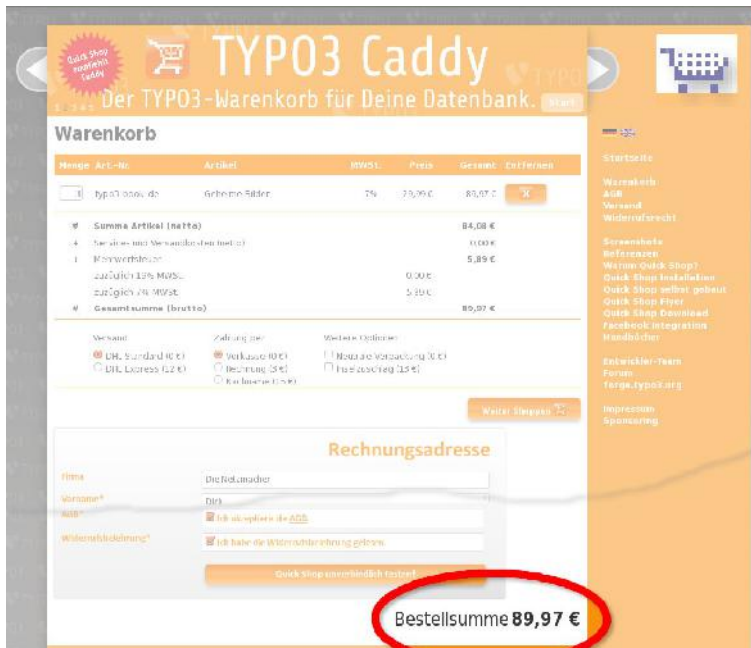


Illustration 29: Plugin Caddy Mini (frontend)

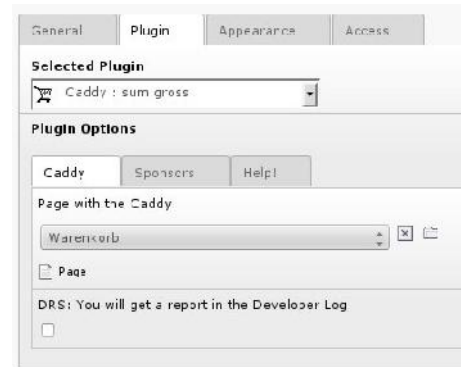


Illustration 30: Plugin Caddy Mini (backend)

Controlling

- Add the Plugin "Caddy: sum gross" to your page.
- Configure the Plugin:
 - Enter the page with the Caddy plugin.

TypoScript

See "Setup > tx_caddy_pi3 – Overview" on page 80 below.

Controlling Caddy

You are controlling Caddy by both user interfaces:

- the Caddy plugin / flexform and
- the Constant Editor.

If you have high demands, you can configure Caddy by TypoScript directly.

Overview

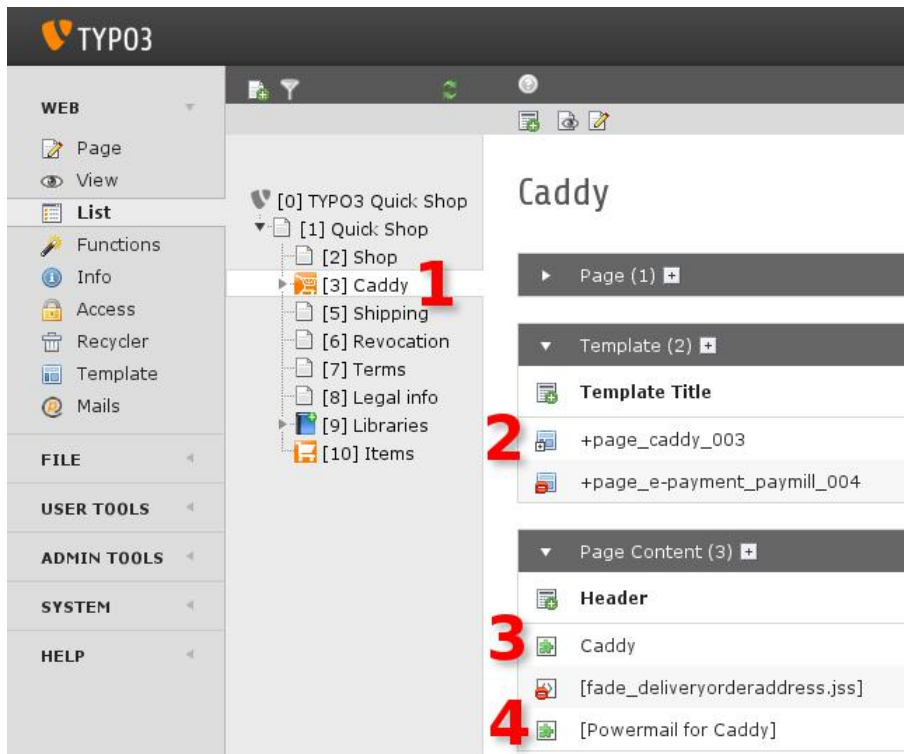


Illustration 31: Sample with Quick Shop

1. The page Caddy should contain:
2. a TypoScript extension template
3. the Caddy plugin
4. the Powermail plugin¹
5. and Powermail items like²
 - forms,
 - pages and
 - fields

¹ Caddy supports both versions: Powermail 1.x and 2.x

² Depends on version 1.x and 2.x

Plugin / Flexform

Check it!

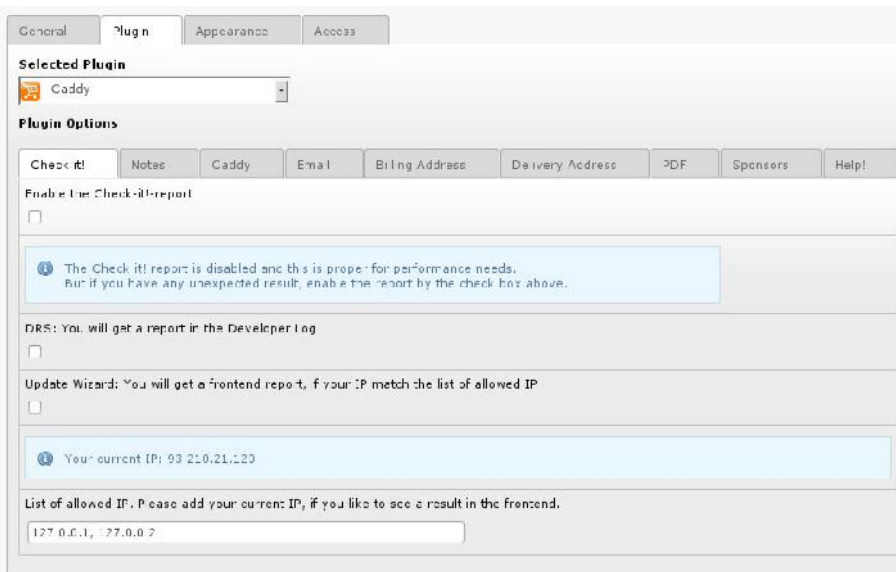


Illustration 32: Plugin [Check it!]

Enable the Check-it!-Report

- If the report is enabled, Caddy will check your Cady configuration and your Powermail configuration.
- If there is an unproper property, you will get a prompt probably with helpful information.

DRS: You will get a report in the Developer Log

- If the DRS is enabled, Caddy will prompt the work flow of the plugin to devlog.
The DRS supports TYPO3 integrators for a proper configuration and for bugfixing.
The DRS supports TYPO3 developers.
- You need the extension devlog, if you want to use this feature.

Update Wizard

- Without any effect in the current version.

Notes



Illustration 33: Plugin [Notes]

Notes

- For your internal notes

Caddy

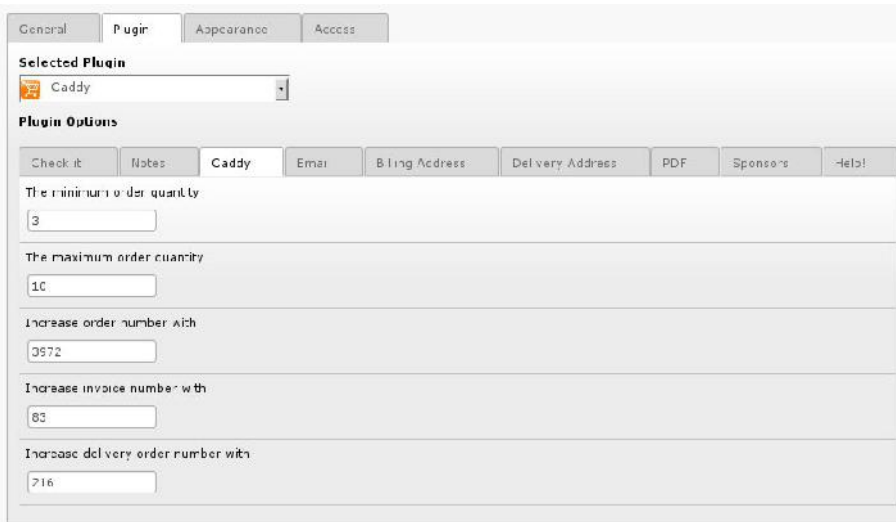


Illustration 34: Plugin [Caddy]

The minimum order quantity

- Minimum order quantity over all (for all items in the caddy)

The maximum order quantity

- Maximum order quantity over all (for all items in the caddy)

Increase order number with

- Caddy starts counting from 0. If you need a higher start number like 1.000, please adapt the starting number to your needs.

Increase invoice number with

- See above.

Increase delivery order number with

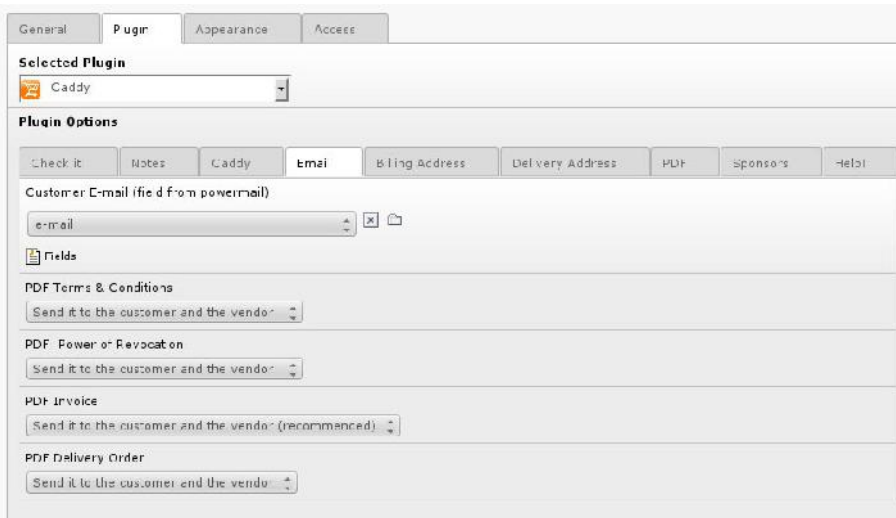
- See above.

Reset numbers

If there is a need, to reset the counter of the numbers, please refer to

- "Reset numbers" on page 66 below.

Email



The screenshot shows the 'Email' configuration tab for the Caddy plugin. At the top, there are tabs for 'General', 'Plugin', 'Appearance', and 'Access'. The 'Selected Plugin' dropdown is set to 'Caddy'. Below this, the 'Plugin Options' section contains several sub-tabs: 'Check it', 'Notes', 'Caddy', 'Email' (which is active), 'Billing Address', 'Delivery Address', 'PDF', 'Sponsors', and 'Help'. The 'Email' sub-tab contains a 'Customer E-mail (field from powermail)' section with a text input field containing 'e-mail' and a 'Fields' button. Below this are five sections, each with a dropdown menu: 'PDF Terms & Conditions' (set to 'Send it to the customer and the vendor'), 'PDF Power of Revocation' (set to 'Send it to the customer and the vendor'), 'PDF Invoice' (set to 'Send it to the customer and the vendor (recommended)'), and 'PDF Delivery Order' (set to 'Send it to the customer and the vendor').

Illustration 35: Plugin [Email]

Customer E-mail (field from powermail)

- Caddy must know the email from the customer.
- You have to link this field with the corresponding field of the powermail form.

PDF Terms & Conditions

- Attach the PDF to
 - nobody,
 - the customer,
 - the vendor or
 - both

PDF: Power of Revocation

- See above

PDF Invoice

- See above

PDF Delivery Order

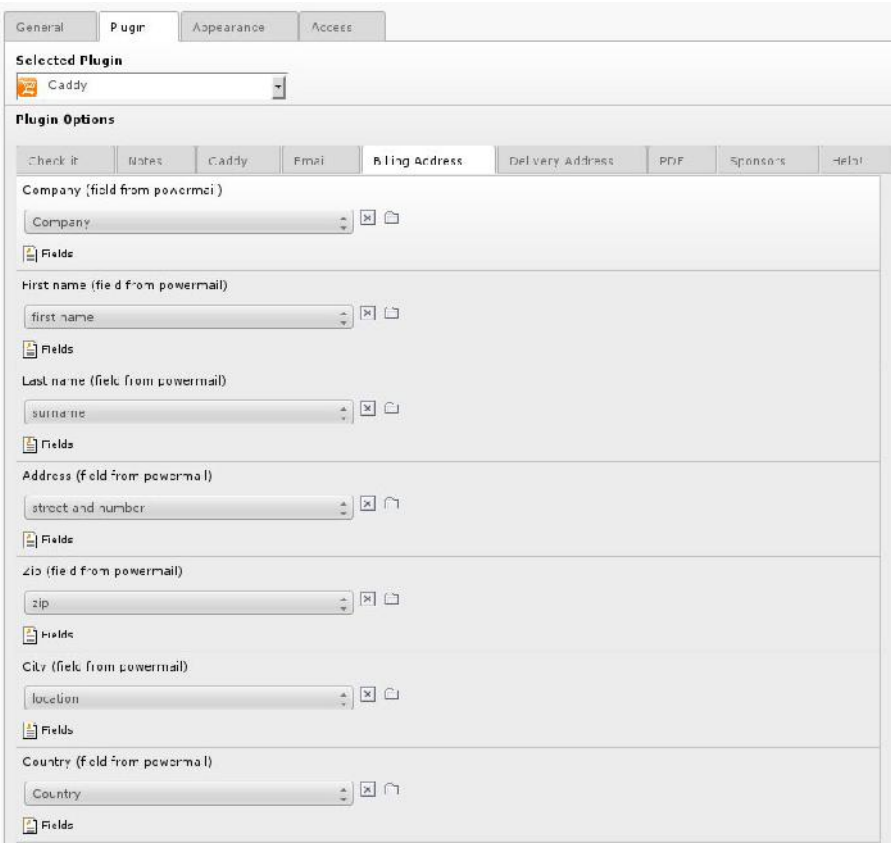
- See above

Design of the attachments

See

- "Plugin / Flexform > PDF" on page 30 below
- "Corporate Design > Attachments" on page 53 below

Billing Address



The screenshot shows the Caddy plugin configuration window. At the top, there are tabs for 'General', 'Plugin', 'Appearance', and 'Access'. The 'Plugin' tab is selected, and 'Caddy' is chosen from the 'Selected Plugin' dropdown. Below this, the 'Plugin Options' section is active, with sub-tabs for 'Check it', 'Notes', 'Caddy', 'Email', 'Billing Address', 'Delivery Address', 'PDF', 'Sponsors', and 'Help'. The 'Billing Address' sub-tab is selected, displaying a list of fields to be mapped from a Powermail form. Each field has a dropdown menu to select the corresponding Powermail field name, a delete icon (X), and a help icon (H).

Field Label	Powermail Field	Delete	Help
Company (field from powermail)	Company	X	H
First name (field from powermail)	first name	X	H
Last name (field from powermail)	surname	X	H
Address (field from powermail)	street and number	X	H
Zip (field from powermail)	zip	X	H
City (field from powermail)	location	X	H
Country (field from powermail)	Country	X	H

Illustration 36: Plugin [Billing Address]

This is the Powermail connector.

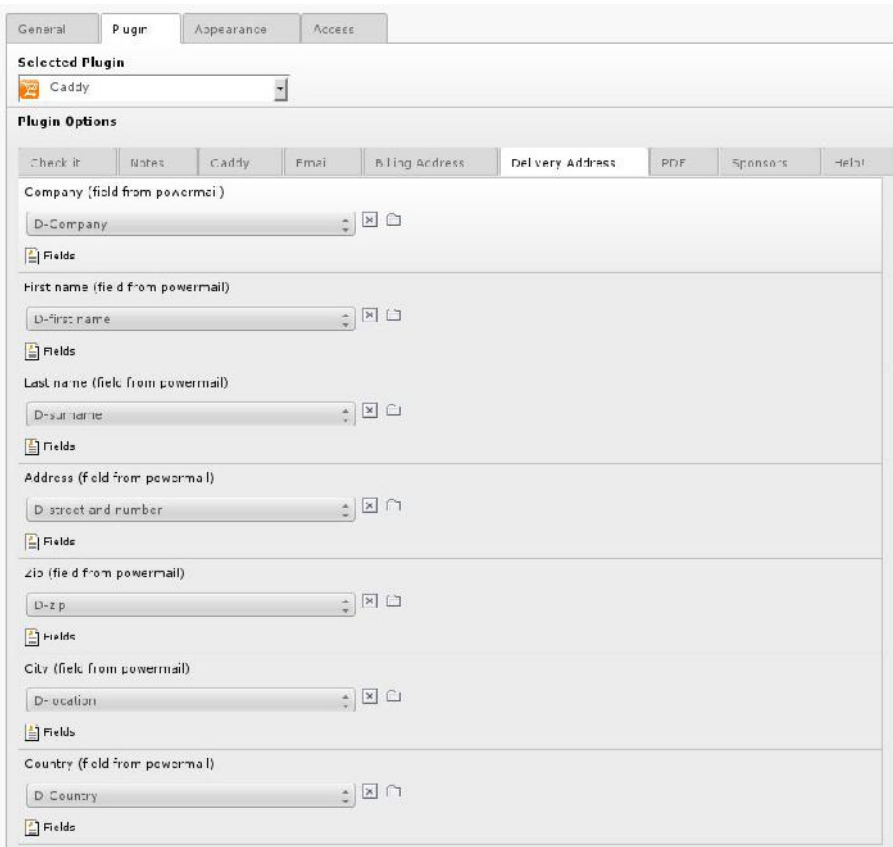
Caddy needs these values for the billing address in PDF attachments.

Each field should be linked with the corresponding field of the powermail form.

It is recommended to have all Powermail fields configured. And to hide a field or fieldset, if you don't need the field or fieldset.

It isn't recommended to leave needed Powermail fields, because Caddy will prompt a warning in the Check-it!-Report.

Delivery Address

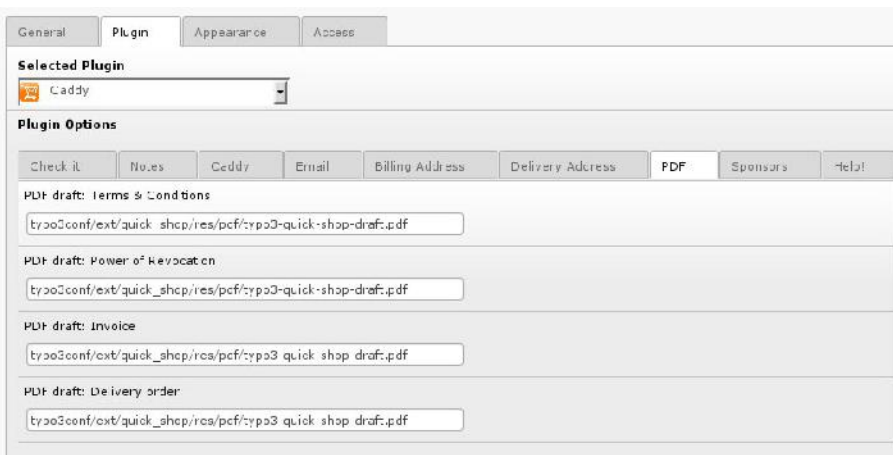


The screenshot shows the 'Delivery Address' tab of the Caddy plugin configuration. The 'Selected Plugin' is 'Caddy'. The 'Plugin Options' section includes tabs for 'Check it', 'Notes', 'Caddy', 'Email', 'Billing Address', 'Delivery Address' (active), 'PDF', 'Sponsors', and 'Help!'. The 'Delivery Address' section contains several fields, each with a 'Fields' button: 'Company (field from powermail)' with value 'D-Company'; 'First name (field from powermail)' with value 'D-first name'; 'Last name (field from powermail)' with value 'D-last name'; 'Address (field from powermail)' with value 'D-street and number'; 'Zip (field from powermail)' with value 'D-zip'; 'City (field from powermail)' with value 'D-location'; and 'Country (field from powermail)' with value 'D-Country'.

Illustration 37: Plugin [Delivery Address]

See "Billing Address" above.

PDF



The screenshot shows the 'PDF' tab of the Caddy plugin configuration. The 'Selected Plugin' is 'Caddy'. The 'Plugin Options' section includes tabs for 'Check it', 'Notes', 'Caddy', 'Email', 'Billing Address', 'Delivery Address', 'PDF' (active), 'Sponsors', and 'Help!'. The 'PDF' section contains four fields, each with a 'PDF draft' label and a text input field: 'PDF draft: Terms & Conditions' with value 'typo3conf/ext/quick_shop/res/pdf/typo3-quick-shop-draft.pdf'; 'PDF draft: Power of Revocation' with value 'typo3conf/ext/quick_shop/res/pdf/typo3-quick-shop-draft.pdf'; 'PDF draft: Invoice' with value 'typo3conf/ext/quick_shop/res/pdf/typo3-quick-shop-draft.pdf'; and 'PDF draft: Delivery order' with value 'typo3conf/ext/quick_shop/res/pdf/typo3-quick-shop-draft.pdf'.

Illustration 38: Plugin [Delivery Address]

Background PDF for your attachments.

Please upload a PDF with your corporate design to the fileadmin directory.

Update the values in the fields of this tab.

Further informations about attachments

See

- "Plugin / Flexform > Email" on page 28 above
- "Corporate Design > Attachments" on page 53 below

Sponsors

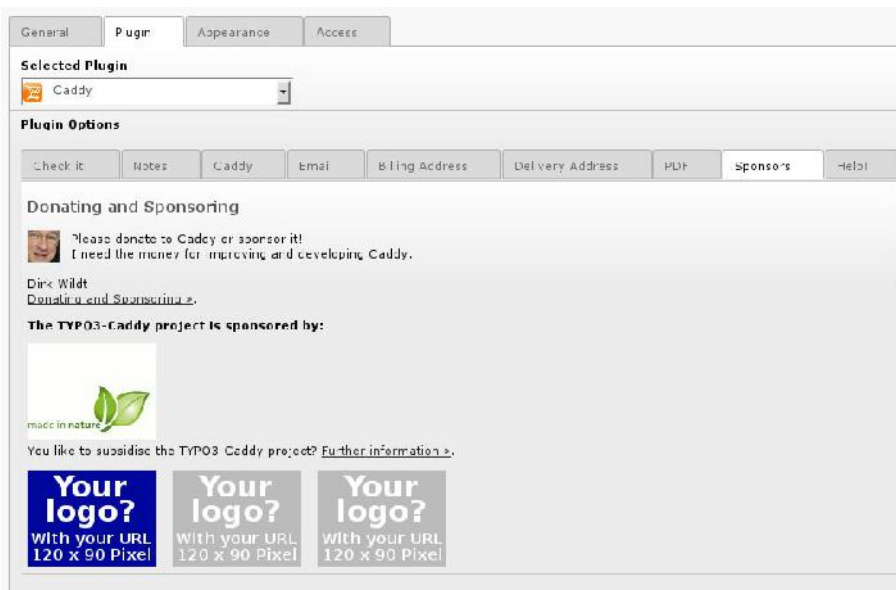


Illustration 39: Plugin [Sponsors]

List of sponsors.

Nothing to do but sponsoring. See

- "Packages for Sponsors" on page 93 below.

Help!



Illustration 40: Plugin [Help!]

Helpful links to

- the manual, the caddy forum and some other caddy websites,
- the one-click-installer and
- the developer.

Nothing to do.

Static Templates

Syntax – Order of the Templates

The labels of the static templates have order numbers.

For example: you should include Caddy [2] before Caddy [4].

Main (obligated)

If you like to use Caddy, you have to include the main template. This is obligated!

- Caddy [1] Basic <- obligated!

If you like to use Caddy together with a Bootstrap framework:

- Caddy [3] + Bootstrap
- Caddy [3.1] + Bootstrap CSS

If you like to use Caddy together with a Foundation framework:

- Caddy [3] + Foundation
- Caddy [3.1] + Foundation CSS

If you need Caddy with German rules like 1.200,50 EUR instead of \$ 1,200.50:

- Caddy [7] + Language German

TypoScript

- "Setup" from page 67 below.

Localization

Some formats are localized like an amount.

- 1,234.56 is the default notation of an amount.

If you like a German format like 1.234,56, please include this static template:

- Caddy [2] + German

CSS

Documentation for Caddy 4.x – maybe proper for version 6.x

CSS templates are needed, if you like to use the CSS of the Caddy extension.¹

CSS in general (blue):

- Caddy [2] CSS (blue)²

Colours

- Caddy [2.1] + CSS orange
- Caddy [2.1] + CSS green
- Caddy [2.1] + CSS red

Foundation

- Caddy [3.1] CSS Foundation 4.x

CSS for Powermail 2.x³

- Caddy [5.1] + Powermail 2.x CSS fancy

Example

- "Colours" on page 50 below.

¹ You are controlling a lot of CSS properties by the Constant Editor.

² Colour is blue

³ There isn't any CSS for Powermail 1.x

TypoScript

- "tx_caddy_pi1._CSS_DEFAULT_STYLE" on page 77 below.

Powermail (obligated)

One of the both Powermail templates is obligate. The proper template depends on your Powermail version.

- Caddy [5] Powermail 1.x
- Caddy [5] Powermail 2.x Main

TypoScript

- "tx_caddy_pi1.powermail-" on page 78 below.

E-Payment

If you like to use the e-payment interface with the e-payment provider Paymill, please include this static template:

- Caddy [8] + E-Payment: Paymill

Reset

If you like to reset all Caddy options, please include this static template at the top position:

- Caddy [99] Reset

Constant Editor

All TypoScript properties are controlled by the Constant Editor. See "Constant Editor" on page 34 below.

Constant Editor

All properties have the prefix plugin.caddy.

Example: main.pid is plugin.caddy.main.pid.

MAIN

Property	Type	Description	Default
main.pid	text	Uid of storage folder: Uid of the folder with the caddy records. Leave it empty (recommended!), to store records at the same page like the caddy plugin.	
main.dueDate	options	Due date for payment. Options <ul style="list-style-type: none"> • now • +1day • +2 days • +3 days • +4 days • +5 days • +6 days • +1 week • +2 weeks • +3 weeks • +1 month 	+2 weeks
main.dueDateFormat	string	Format of the due date. Examples: <ul style="list-style-type: none"> • %Y-%m-%d • %d. %b. %Y 	%Y-%m-%d

MAIN NUMBERFORMAT

For German rules there isn't any need to use the Constant Editor. See

- "Static Templates > Error: Reference source not found" on page Error: Reference source not found Error: Reference source not found.

Property	Type	Description	Default
main.decimal	text	Decimal. This setting will be used for formatting prices.	2
main.dec_point	text	Decimal symbol: Dot or comma. This setting will be used for formatting prices.	.
main.thousands_sep	text	Thousands separator. Dot or comma. This setting will be used for formatting prices.	,
main.currencySymbol	text	Currency symbol. Example: &euro, \$	\$
main.currencySymbolBeforePrice	boolean	Show currency symbol before price.	true
main.percentSymbol	string	Percent symbol	%

BANKACCOUNT

Property	Type	Description	Default
bankaccount.iban	text	IBAN: International Bank Account Number (IBAN). Is needed for the payment option cash in advance.	
bankaccount.bic	text	BIC: Bank identifier code (BIC). Is needed for the payment option cash in advance.	title

DATABASE

Property	Type	Description	Default
db.table	text	Table*: *Obligate! Name of the table with the items (products). Examples: <ul style="list-style-type: none"> tx_myextension_products tx_quickshop_products tx_cars_model 	
db.title	text	Title*: *Obligate! Field with the title of the item (product).	title
db.price	text	Price*: *Obligate! Field with the gross price of the item (product).	price
db.tax	text	Tax*: *Obligate! Field with the tax category of the item (product).	tax
db.sku	text	SKU: field with the sku (unique id) of the item (product).	
db.min	text	MIN: field with the minimum order quantity of the item (product).	
db.max	text	MAX: field with the minimum order quantity of the item (product).	
db.service_attribute_1	text	Service Attribute 1: the field with the service attribute 1 of the item (product).	
db.service_attribute_2	text	Service Attribute 2: the field with the service attribute 1 of the item (product).	
db.service_attribute_3	text	Service Attribute 3: the field with the service attribute 1 of the item (product).	

DEBUG

Property	Type	Description	Default
debug.paramsAndTs	boolean	Debug params/typoscript (frontend): Get a report in the frontend with received GET-/POST-parameters and typoscript configuration	false
debug.userfunc	boolean	Debug user functions (backend): Enable the DRS for user functions	false
debug.dontReplaceEmptyMarker	boolean	Don't replace empty marker: Enable it, if empty HTML markers should not removed.	false

E-PAYMENT

Property	Type	Description	Default
e-payment.provider	string	E-payment provider. Currently Paymill is possible only.	
e-payment.currency	string	I.e: EUR, USD. Currency has an effect for transactions! Currency have to accord with the currency of your Shop (see: CADDY - MAIN NUMBERFORMAT).	EUR

E-PAYMENT – PAYMILL

You have to include the static template "**Caddy [6] + E-Payment Paymill (caddy)**"

Property	Type	Description	Default
e-payment.paymill.mode	string	live <ul style="list-style-type: none"> Transaction are real transactions. Each transaction costs your money. test <ul style="list-style-type: none"> Transactions hasn't any effect, but Paymill will display each transaction on the dashboard. BE AWARE that customers can order items without a real payment! 	test
e-payment.paymill.key.live.private	string	Key (live) - private*:Your private key for the live mode.	
e-payment.paymill.key.live.public	string	Key (live) - public*:Your publickey for the live mode.	
e-payment.paymill.key.test.private	string	Key (live) - private*:Your private key for the test mode.	
e-payment.paymill.key.test.public	string	Key (live) - public*:Your publickey for the test mode.	

E-PAYMENT – PAYMILL FILES

You have to include the static template "**Caddy [6] + E-Payment Paymill (caddy)**"

Property	Type	Description	Default
e-payment.paymill.files.*	string	Needed Paymill files	

E-PAYMENT – PAYMILL PATHS

You have to include the static template "**Caddy [6] + E-Payment Paymill (caddy)**"

Property	Type	Description	Default
e-payment.paymill.paths.api	string	API:Path to the API Default: res/e-payment/paymill/api/php/	see description

E-PAYMENT – POWERMAIL FILES

You have to include the static template "**Caddy [6] + E-Payment Paymill (caddy)**"

Property	Type	Description	Default
e-payment.powermail.files.html.error	string	Error template:Path to the html error template Default: EXT:caddy/res/e-payment/powermail/foundation/4x/html/error.html	see description

E-PAYMENT – POWERMAIL PATHS

You have to include the static template "**Caddy [6] + E-Payment Paymill (caddy)**"

Property	Type	Description	Default
e-payment.powermail.paths.foundation	string	Path to foundation typo3conf/ext/t3foundation/res/foundation-4.3.2/	see description

GETPOST

Property	Type	Description	Default
getpost.uid	text	Uid*: *Obligate! Parameter name for a unique id (integer) of your item. caddy uses this id for SQL requests. It must correspond with [CADDY – URL]! Example <ul style="list-style-type: none">tx_browser_pi1 showUid	
getpost.quantity	text	Quantity*: *Obligate! Parameter name for the quantity/amount field of your order form (e.g. tx_trbooks_qty). The provided parameter must be equal to the name in your HTML template.	qty
getpost.title	text	Title: Parameter name for the title field in your order form.	
getpost.price	text	Parameter name for the gross price field in your order form.	
getpost.service_attribute_1	text	Service Attribute 1: Parameter name for the service attribute field 1 of your order form (e.g. weight).	
getpost.service_attribute_1	text	Service Attribute 2: Parameter name for the service attribute field 2 of your order form (e.g. volume).	
getpost.service_attribute_1	text	Service Attribute 3: Parameter name for the service attribute field 3 of your order form (e.g. length).	
getpost.sku	text	Parameter name for your SKU (stock keeping unit).	

HTML

HTML COLORS

There isn't any need to use the Constant Editor, if you like to use these colours:

- blue, green, orange, red

Please include the static template:

- +Caddy CSS +Colour (caddy)

where colour is one of the colours from above.

See

- "Static Templates > CSS" on page 32 above.
- "Colours" on page 50 below.

Property	Type	Description	Default
html.colour.border	colour	Border colour: border colour as CSS name or HTML notation like #FFF or #123456. It is used in the HTML table, which is placed in e-mails and pdf attachments.	#207CCA
html.colour.button.text	colour	Button text colour	#FFF
html.colour.button.gradient1	colour	Button gradient colour 1	#1E5799
html.colour.button.gradient2	colour	Button gradient colour 2	#2989D8
html.colour.button.gradient3	colour	Button gradient colour 3	#207CCA
html.colour.button.gradient4	colour	Button gradient colour 4	#7DB9E8
html.colour.icon.caddy	string	Icon caddy. Default <ul style="list-style-type: none"> • EXT:caddy/files/img/caddy_080_08.png Example <ul style="list-style-type: none"> • "Corporate Design > Icons" on page 51 below 	see description
html.colour.icon.delete	string	Icon delete. Default <ul style="list-style-type: none"> • EXT:caddy/files/img/delete_080_08.png Example <ul style="list-style-type: none"> • "Corporate Design > Icons" on page 51 below 	see description
html.colour.powermail.text	colour	Powermail text	#207CCA
html.colour.table.head.background	colour	Table header background	#207CCA
html.colour.table.head.text	colour	Table header text	#FFF

HTML MARKER

Property	Type	Description	Default
html.marker.caddy	string	Subpart Marker: The subpart marker for the caddy in your Caddy HTML template. Usually ###CADDY###.	CADDY

HTML WIDTH

Property	Type	Description	Default
html.width.table.td.qty	text	Width column Quantity: The width of the caddy column quantity in HTML tables for CSS. Units can be px, em or % among others.	10%
html.width.table.td.sku	text	Width column SKU: The width of the caddy column stock keep unit in HTML tables for CSS. Units can be px, em or % among others.	15%
html.width.table.td.item	text	Width column item: The width of the caddy column item in HTML tables for CSS. Units can be px, em or % among others.	35%
html.width.table.td.tax	text	Width column tax: The width of the caddy column tax in HTML tables for CSS. Units can be px, em or % among others.	10%
html.width.table.td.net	text	Width column net: The width of the caddy column net in HTML tables for CSS. Units can be px, em or % among others.	15%
html.width.table.td.sum	text	Width column sum: The width of the caddy column sum in HTML tables for CSS. Units can be px, em or % among others.	15%
html.width.table.td.skuitemtax	text	Width 3 columns: The total width of the caddy columns sku, item and tax in HTML tables for CSS. Units can be px, em or % among others.	60%
html.width.table.td.skuitemtaxnet	text	Width 4 columns: The total width of the caddy columns sku, item, tax and net in HTML tables for CSS. Units can be px, em or % among others.	75%

Options

OPTIONS PAYMENT

Property	Type	Description	Default
options.payment.default	integer	Default payment option:0: no preset. 1: 1st payment option. 2: 2nd payment option. 3rd payment option.	0
options.payment.1.lll	string	1. label:Label for 1st payment option. Example: cash in advance (German: Vorkasse) Default: EXT:caddy/pi1/locallang.xml:paymentoption_cashinadvance	see description
options.payment.1.enabled	boolean	1. enabled:Disable it, if you don't want to offer the 1st payment option.	1
options.payment.1.costs	double	1. costs:Extra costs for 1st payment option. Example: 9.99	0.00
options.payment.1.taxrate	string	1. tax rate:Tax rate for extra costs for 1st payment option. Values are: reduced, normal	normal
options.payment.[2-3]		Same options for options payment 2 to options.payment 3	

OPTIONS SHIPPING

Property	Type	Description	Default
options.shipping.default	integer	Default shipping option:0: no preset. 1: mail. 2: mail express.	0
TODO options.shipping.1.lll	TODO string	TODO 1. label:Label for 1st shipping option. Example: cash in advance (German: Vorkasse) Default: EXT:caddy/pi1/locallang.xml:shippingoption_cashinadvance	see description
options.shipping.1.enabled	boolean	1. enabled:Disable it, if you don't want to offer the 1st shipping option.	1
options.shipping.1.costs	double	1. costs:Extra costs for 1st shipping option. Example: 9.99	0.00
options.shipping.1.taxrate	string	1. tax rate:Tax rate for extra costs for 1st shipping option. Values are: reduced, normal	normal
options.shipping.[2]		Same options for option shipping 2	

OPTIONS SPECIALS

Property	Type	Description	Default
options.specials.default	integer	Default specials option:0: no preset. 1: neutral package. 2: fee for islands (German: Versandkosten Deutsche Inseln).	0
TODO options.specials.1.!!!	TODO string	TODO 1. label:Label for 1st specials option. Example: cash in advance (German: Vorkasse) Default: EXT:caddy/pi1/locallang.xml:specialsoption_cashinadvance	see description
options.specials.1.enabled	boolean	1. enabled:Disable it, if you don't want to offer the 1st specials option.	1
options.specials.1.costs	double	1. costs:Extra costs for 1st specials option. Example: 9.99	0.00
options.specials.1.taxrate	string	1. tax rate:Tax rate for extra costs for 1st specials option. Values are: reduced, normal	normal
options.specials.[2]		Same options for option specials 2	

PAGES

Property	Type	Description	Default
pages.caddy	integer	Caddy*:Obligate! Uid of the page with the caddy - your shopping cart	
pages.caddymini	integer	Caddy mini:Uid of the page with the caddy mini - the cart for marginal column	
pages.shop	integer	Shop*:Obligate! Uid of the page which contains the items. Usually the root page.	
pages.revocation	integer	Power of Revocation:Uid of the page with the revocation info (German: Widerrufsbelehrung)	
pages.terms	integer	Terms & Conditions:Uid of the page with the terms & conditions (German: AGB)	

PDF

Property	Type	Description	Default
pdfGeneral.fontfamily	options	Font family: The font family of your PDF attachments Options <ul style="list-style-type: none"> Courier Helvetica Symbol Times ZapfDingbats 	Helvetica
pdfGeneral.fontStretching	integer	Font Stretching: In percent. 100: no effect. 90: smaller. 110: expanded	90

PDF CADDY

Property	Type	Description	Default
pdf.caddy.x	int+	Caddy position left: Position of the caddy (items) from the left margin in millimeters	25
pdf.caddy.y	int+	Caddy position top: Position of the caddy (items) from the top margin in millimeters	130
pdf.caddy.width	int+	Caddy width: Width of the caddy (items) in millimeters	165
pdf.caddy.fontsize	int+	Caddy font-size: Font-size of the caddy (items) in points	10

PDF DATE

Property	Type	Description	Default
pdf.date.align	options	Date alignment: L: left. C: center. R: right. empty: left for LTR or right for RTL Options <ul style="list-style-type: none"> empty L C R 	L
pdf.date.x	int+	Date position left: Position of the date from the left margin in millimeters	175
pdf.date.y	int+	Date position top: Position of the date from the top margin in millimeters	104
pdf.date.width	int+	Date width: Width of the date in millimeters	25
pdf.date.fontsize	int+	Date font-size: Font-size of the date in points	10
pdf.date.strftime	string	Date format: Format of the date. Examples: %Y-m-%d, %d. %M %Y	%Y-%m-%d

PDF DELIVERY ORDER

Property	Type	Description	Default
pdf.deliveryorder.address.x	int+	Address position left: Position of the delivery order address from the left margin in millimeters	25
pdf.deliveryorder.address.y	int+	Address position top: Position of the delivery order address from the top margin in millimeters	55
pdf.deliveryorder.address.fontsize	int+	Address font-size: Font-size of the delivery order address in points	10
pdf.deliveryorder.number.x	int+	Address position left: Position of the delivery number from the left margin in millimeters	25
pdf.deliveryorder.number.y	int+	Address position top: Position of the delivery number from the top margin in millimeters	110
pdf.deliveryorder.number.fontsize	int+	Address font-size: Font-size of the delivery number in points	24
pdf.deliveryorder.number.prefix	string	Number prefix: Prefix for the delivery number. Example : D-	
pdf.deliveryorder.termOfCredit.x	int+	Address position left: Position of the term of credit from the left margin in millimeters	25
pdf.deliveryorder.termOfCredit.y	int+	Address position top: Position of the term of credit from the top margin in millimeters	250
pdf.deliveryorder.termOfCredit.fontsize	int+	Address font-size: Font-size of the term of credit in points	10

PDF INVOICE

Property	Type	Description	Default
pdf.invoice.address.x	int+	Address position left: Position of the invoice address from the left margin in millimeters	25
pdf.invoice.address.y	int+	Address position top: Position of the invoice address from the top margin in millimeters	55
pdf.invoice.address.fontsize	int+	Address font-size: Font-size of the invoice address in points	10
pdf.invoice.number.x	int+	Address position left: Position of the invoice number from the left margin in millimeters	25
pdf.invoice.number.y	int+	Address position top: Position of the invoice number from the top margin in millimeters	110
pdf.invoice.number.fontsize	int+	Address font-size: Font-size of the invoice number in points	24
pdf.invoice.number.prefix	string	Number prefix: Prefix for the invoice number. Example : D-	
pdf.invoice.termOfCredit.x	int+	Address position left: Position of the term of credit from the left margin in millimeters	25
pdf.invoice.termOfCredit.y	int+	Address position top: Position of the term of credit from the top margin in millimeters	250
pdf.invoice.termOfCredit.fontsize	int+	Address font-size: Font-size of the term of credit in points	10

PDF ORDER

Property	Type	Description	Default
pdf.order.number.prefix	string	Number prefix: Prefix for the order number. Example : O-	

PDF REVOCATION

Property	Type	Description	Default
pdf.revocation.align	options	Revocation alignment: L: left. C: center. R: right. empty: left for LTR or right for RTL Options <ul style="list-style-type: none"> • <i>empty</i> • L • C • R 	
pdf.revocation.x	int+	Revocation position left: Position of the date from the left margin in millimeters	25
pdf.revocation.y	int+	Revocation position top: Position of the date from the top margin in millimeters	100
pdf.revocation.width	int+	Revocation width: Width of the date in millimeters	165
pdf.revocation.fontsize	int+	Revocation font-size: Font-size of the date in points	10

PDF TERMS

Property	Type	Description	Default
pdf.terms.align	options	Terms alignment: L: left. C: center. R: right. empty: left for LTR or right for RTL Options <ul style="list-style-type: none"> • <i>empty</i> • L • C • R 	
pdf.terms.x	int+	Terms position left: Position of the date from the left margin in millimeters	25
pdf.terms.y	int+	Terms position top: Position of the date from the top margin in millimeters	100
pdf.terms.width	int+	Terms width: Width of the date in millimeters	165
pdf.terms.fontsize	int+	Terms font-size: Font-size of the date in points	10

PDF TEXTCOLOR

Property	Type	Description	Default
pdf.textColour.address	colour	Address: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.caddy	colour	Caddy: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.date	colour	Date: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.numberDeliveryorder	colour	Number delivery order: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.numberInvoice	colour	Number invoice: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.numberLine	colour	Number line: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.numberOrder	colour	Number order: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.revocation	colour	Revocation: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.termOfCredit	colour	Term of credit: font colour as CSS name or HTML notation like #FFF or #123456	navy
pdf.textColour.terms	colour	Terms: font colour as CSS name or HTML notation like #FFF or #123456	navy

SYMBOLS AND SERVICE ATTRIBUTES

Property	Type	Description	Default
symbol.quantitySymbol	text	Quantity symbol: for example: pc. (English), St. (German)	pc.
symbol.service_attribute_1_symbol	text	Service Attribute 1 symbol: e.g. Weight: g, kg, lbs	
symbol.service_attribute_2_symbol	text	Service Attribute 2 symbol: e.g. Volume: l, gal	
symbol.service_attribute_3_symbol	text	Service Attribute 3 symbol: e.g. Length: m, cm, inch	

TAX

Property	Type	Description	Default
tax.reduced	text	Tax rate for reduced items (e.g. 7). Value will be displayed.	7
tax.reducedCalc	text	Tax rate for reduced items (e.g. 0.07). Used for calculation.	0.07
tax.normal	text	Tax rate for normal items (e.g. 19). Value will be displayed.	19
tax.normalCalc	text	Tax rate for normal items (e.g. 0.19). Used for calculation.	0.19

TEMPLATES

Property	Type	Description	Default
templates.e-mail	string	E-mails: Path to the HTML template file, which is used in e-mails. Default <ul style="list-style-type: none">EXT:caddy/files/templates/caddy.html	see description
templates.html.caddy	string	HTML caddy: Path to the HTML template file, which is used in HTML pages for the caddy Default <ul style="list-style-type: none">EXT:caddy/files/templates/caddy.html	see description
templates.html.caddymini	string	HTML caddy mini: Path to the HTML template file, which is used in HTML pages for the mini caddy Default <ul style="list-style-type: none">EXT:caddy/files/templates/caddy.html	see description
templates.html.caddysum	string	HTML caddy sum: Path to the HTML template file, which is used in HTML pages for the caddy sum Default <ul style="list-style-type: none">EXT:caddy/files/templates/caddy.html	see description
templates.pdf.deliveryorder	string	PDF delivery order: Path to the HTML template file, which is used in the delivery order - a PDF attachment Default <ul style="list-style-type: none">EXT:caddy/files/templates/caddy.html	see description
templates.pdf.invoice	string	PDF invoice: Path to the HTML template file, which is used in the invoice - a PDF attachment Default <ul style="list-style-type: none">EXT:caddy/files/templates/caddy.html	see description

URL

Property	Type	Description	Default
url.showUid	string	showUid: Alias for the URL parameter showUid. It must correspond with [CADDY - GETPOST: uid] . I.e: calendarUid	itemUid
url.extension	String	Extension parameter: Parameter of your extension (sends the item uid). It must correspond with [CADDY - GETPOST: uid] . If you are using Browser - TYPO3 without PHP, it is tx_browser_pi1.	tx_browser_pi1

Calculating

Default code

Caddy is delivered with a default payment, shipping and specials configuration. See TypeScript Object Browser at

```
plugin.tx_caddy_pi1 {
    api {
        options {
            payment {
                ...
            }
            shipping {
                ...
            }
            specials {
                ...
            }
        }
    }
}
```

See detailed information at "tx_caddy_pi1.api.options.payment" on page 72 below.

Non documented properties

Each shipping and payment method have four properties:

- available_from
- available_until
- free_from
- free_until

and a general setting for shipping and payment

- show_all_disabled

Sorry, these properties aren't documented in this manual. The code snippet below contains an example with "available_until".

Sample with a modified default code

```
plugin.tx_caddy_pi1 {
    api {
        options {
            ...
            shipping {
                options {
                    1 {
                        title {
                            data >
                            value = Shipping by GLS
                            noTrimWrap >
                        }
                        extra = by_quantity
                        extra {
                            1 {
                                value = 1
                                extra = 5.00
                            }
                            2 {
                                value = 6
                                extra = 7.40
                            }
                        }
                    }
                    // option is only available if gross price is less than 200.00
                    available_until = 200.00
                }
            }
            ...
        }
    }
}
```

Based on price (by_price)

With this rule you can calculate one of your service costs depending on the total gross price for all products.

Setup

```
plugin.tx_caddy_pil {
    api {
        options {
            ...
            shipping {
                options {
                    1 {
                        title = Standard
                        extra = by_price
                        extra {
                            1 {
                                value = 0
                                extra = 10.00
                            }
                            2 {
                                value = 100
                                extra = 7.50
                            }
                            3 {
                                value = 200
                                extra = 5.00
                            }
                        }
                    }
                    tax = normal
                }
                ...
            }
        }
    }
}
```

In the example there is one shipping method called "Standard", the normal tax will be applied.

- If the total gross price of all products is lower than 100.00 the shipping costs will be 10.00.
- If the total gross price of all products is lower than 200.00 the shipping costs will be 7.50.
- If the total gross price of all products is equal to or higher than 200.00 the shipping costs will be 5.00.

Based on quantity (by_quantity)

With this rule you can calculate one of your service costs depending on the total quantity for products.

Constants

```
plugin.caddy {
  symbol {
    // Constant Editor [CADDY - SYMBOLS AND SERVICE ATTRIBUTES] > Quantity symbol
    quantitySymbol = Pcs.
  }
}
```

With the setting above you can configure the symbol or abbreviation which is displayed as suffix of the quantity figure.

Setup

```
plugin.tx_caddy_pil {
  options {
    shipping {
      options {
        1 {
          title {
            data >
            value = Shipping by GLS
            noTrimWrap >
          }
          extra = by_quantity
          extra {
            1 {
              value = 1
              extra = 5.00
            }
            2 {
              value = 6
              extra = 7.40
            }
          }
        }
        // method only available if gross price is less than 200.00
        available_until = 200.00
      }
    }
  }
}
```

In the example above we configure a special shipping rule based on the quantity. There is one shipping method called "Standard", the normal tax will be applied.

- If the total quantity of all products is lower than 4 the shipping costs will be 5.35.
- If the total quantity of all products is lower than 11 the shipping costs will be 7.10.
- If the total quantity of all products is lower than 21 the shipping costs will be 9.70.
- If the total quantity of all products is equal to or higher than 21 the shipping costs will be 11.60.

Based on each (each)

With this rule you can calculate one of your service costs depending on the total quantity of all products. The quantity will be multiplied by the price you set for each product.

Example setup

```
plugin.tx_caddy_pi1 {
    api {
        options {
            ...
            shipping {
                options {
                    1 {
                        title = Standard
                        extra = each
                        extra {
                            1 {
                                extra = 2.00
                            }
                        }
                    }
                    tax = normal
                }
                ...
            }
        }
    }
}
```

In the example above we configure a special shipping rule for each product. There is one shipping method called "Standard", the normal tax will be applied.

- The shipping costs will be 2.00 for each product in your cart.
- If you have got 10 products in your cart (total quantity) the shipping will be 20.00.

Individual sum-calculation (by_service_attribute_x_sum)

With this rule you can calculate one of your service costs depending on the total of an individual attribute for all products. You have to set either the correct database field.

Constants

```
plugin.caddy {
  db {
    // Constant Editor [CADDY - DATABASE] > Service Attribute 1
    service_attribute_1 = weight
  }
  symbol {
    // Constant Editor [CADDY - SYMBOLS AND SERVICE ATTRIBUTES] > Service Attribute 1 symbol
    service_attribute_1_symbol = lbs
  }
}
```

The constants set the database field and the abbreviation which is displayed as suffix of the quantity value.

Setup

```
plugin.tx_caddy_pil {
  api {
    options {
      ...
      shipping {
        options {
          1 {
            title = Standard
            extra = by_service_attribute_1_sum
            extra {
              1 {
                value = 600
                extra = 4.00
              }
              2 {
                value = 800
                extra = 6.00
              }
              3 {
                value = 1000
                extra = 12.00
              }
            }
          }
          tax = normal
        }
      }
      ...
    }
  }
}
```

In the example above we configure a special shipping rule based on the total of an individual attribute "weight" in pounds (lbs). The attribute is stored in your database.

- If the total weight of all products is lower than 600 the shipping costs will be 0.00.
- If the total weight of all products is lower than 800 the shipping costs will be 4.00.
- If the total weight of all products is lower than 1.000 the shipping costs will be 6.00.
- If the total weight of all products is equal to or higher than 1.000 the shipping costs will be 12.00.

Individual max-calculation (by_service_attribute_x_max)

With this rule you can calculate one of your service costs depending on the maximum value of an individual attribute. For example this rule could be used for sending posters. The longest poster determines the length of the packaging.

Constants

```
plugin.caddy {
  db {
    // Constant Editor [CADDY - DATABASE] > Service Attribute 2
    service_attribute_2 = size
  }
  symbol {
    // Constant Editor [CADDY - SYMBOLS AND SERVICE ATTRIBUTES] > Service Attribute 2 symbol
    service_attribute_2_symbol = cm
  }
}
```

Setup

```
plugin.tx_caddy_pil {
  api {
    options {
      ...
      shipping {
        options {
          1 {
            title = Standard
            extra = by_service_attribute_2_max
            extra {
              1 {
                value = 0
                extra = 4.00
              }
              2 {
                value = 80
                extra = 6.00
              }
              3 {
                value = 100
                extra = 8.00
              }
              4 {
                value = 200
                extra = 12.00
              }
            }
            tax = normal
          }
          ...
        }
      }
    }
  }
}
```

In the example above we configure a special shipping rule based on the maximum of an individual attribute "length" in centimeters (cm). The attribute is a POST parameter.

- If the maximum length is lower than 80 the shipping costs will be 4.00.
- If the maximum length is lower than 100 the shipping costs will be 6.00.
- If the maximum length is lower than 200 the shipping costs will be 8.00.
- If the maximum length is equal to or higher than 200 the shipping costs will be 12.00.

Please not that the quantity of a specific poster is not important. The same poster does not get longer if you buy 2 or 3 etc.

Corporate Design

Colours

Documentation for Caddy 4.x – maybe proper for version 6.x

Caddy is delivered with the four colour schemes:

- blue, green, orange, red.

This schemes has an effect on

- HTML content,
- E-mail and
- PDF attachments.

Screenshots



Illustration 41: Caddy blue



Illustration 42: Caddy green



Illustration 43: Caddy orange



Illustration 44: Caddy red

Select a colour scheme

You select a colour scheme by including a static template for this colour. See

- "Static Templates > CSS" on page 32 above.

The static template overwrites the colour properties in the Constant Editor. If you need another colour scheme, you have to configure the colours in the Constant Editor. See

- "Constant Editor > HTML" on page 37 above.

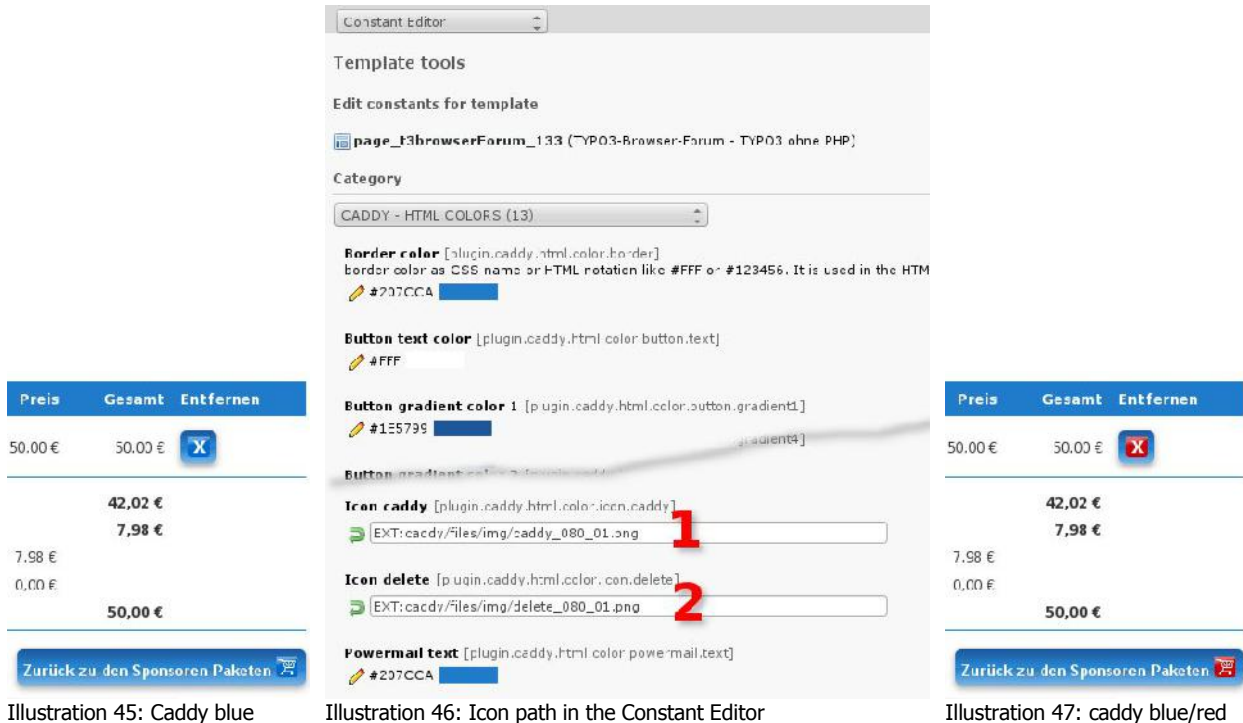
Send your scheme to the developer!

If you are creating your own scheme, please send the values of the Constant Editor to the developer. I will include this scheme as a static template. My email: <http://wildt.at.die-netzmacher.de/>

Icons

Documentation for Caddy 4.x – maybe proper for version 6.x

Controlling



If you like to change the icons, you have to configure two properties with the Constant Editor:

1. html.colour.icon.caddy
2. html.colour.icon.delete

See "Constant Editor > HTML" on page 37 above.

We replaced blue icons with red icons in the illustration above.

Ready-to-use icons



Illustration 48: Caddy icons

Caddy is delivered with two icons:

- the caddy icon
- the delete icon

Each icon is available in 13 colours.

Each colour is available in three intensities.

See the 82 icons in the illustration above.

The icons are stored at

- files/img and
- <http://forge.typo3.org/projects/extension-caddy/repository/show/trunk/files/img>

E-Mail

Coloured lines are used in e-mails.

You configure the colour by the Constant Editor. See

- "Colours" on page 50 above.
- "Constant Editor > HTML" on page 37 above.

HTML Template

- "Constant Editor > TEMPLATES" on page 43 above.
- "Setup > tx_caddy_pi1.templates" on page 76 below.

Attachments

Caddy attaches PDF files to an e-mail.

This PDF attachments are applicable:

- invoice
- delivery order
- power of revocation
- terms and conditions

You can add more attachments unlimited.

Corporate Design



Illustration 49: Your (1) draft will merged ... Illustration 50: ... with (2) Caddy content ... Illustration 51: ... to the (3) attachment.

You can use your own corporate design for each attachment:

1. Your draft (left hand) will merged with
2. the content generated by Caddy (middle) to
3. the attachment (right hand).

The illustration above displays an example for an invoice.

Your own draft (PDF version upto 1.4)

Your draft must be a PDF file in **PDF version 1.4 or lower**. You will have best quality.

Caddy is using the German DIN A/4 paper size by default:

- width: 210 millimeter
- height: 297 millimeters

Upload

Please upload your draft:

1. Modul File Filelist
2. Page tree Select a folder
3. Edit area Upload your PDF file

In our example we uploaded my.pdf into the directory fileadmin/user_upload/quick_shop.

Configuring Paths

Please configure the Caddy paths.

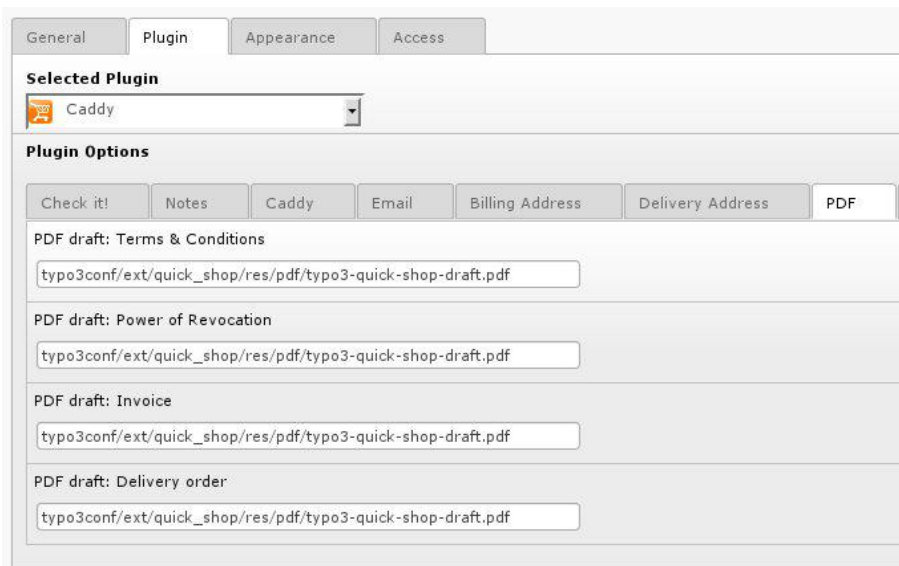


Illustration 52: [PDF] tab of the Caddy plugin

1. Modul Web List
2. Page tree The page with the caddy plugin
3. Edit area Select the caddy plugin
4. tab [Plugin] > tab [PDF]
(see illustration above)
5. Enter the path of your PDF file
(here: fileadmin/user_upload/quick_shop/my.pdf)

Colours

You configure the colour ready-to-use by the include static templates and for individual needs by the Constant Editor. See

- "Colours" on page 50 above.
- "Constant Editor > HTML" on page 37 above.
- "Constant Editor > PDF TEXTCOLOR" on page 42 above

Properties

You can control nearly everything within a PDF attachment:

- the invoice address (1)
- the delivery address (2)
- the date (3)
- the so called number line (4)
- the caddy (5), the power of revocation and the terms and conditions
- the term of credit (6)
- the background and
- the font.

You are controlling this properties overall:

- the date,
- the font family,
- the font stretching and
- the font colour.

You are controlling this properties per attachment¹:

- invoice addresses and delivery address
- invoice number and delivery number
- terms of credit

Background

You can use your own corporate design for the background in an attachment. See

- "Corporate Design" on page 53 above.

Font Family and Font Stretching

You are controlling

- the font family and
- the font stretching

in general by the Constant editor. See

- "Constant Editor > PDF" on page 40 above,
- "Constant Editor > PDF" on page 40 above.

You are controlling it for each element by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

Font Colour

You are controlling the font colour in general by the Constant editor. See

- "Constant Editor > PDF TEXTCOLOR" on page 42 above.

You are controlling it for each element by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

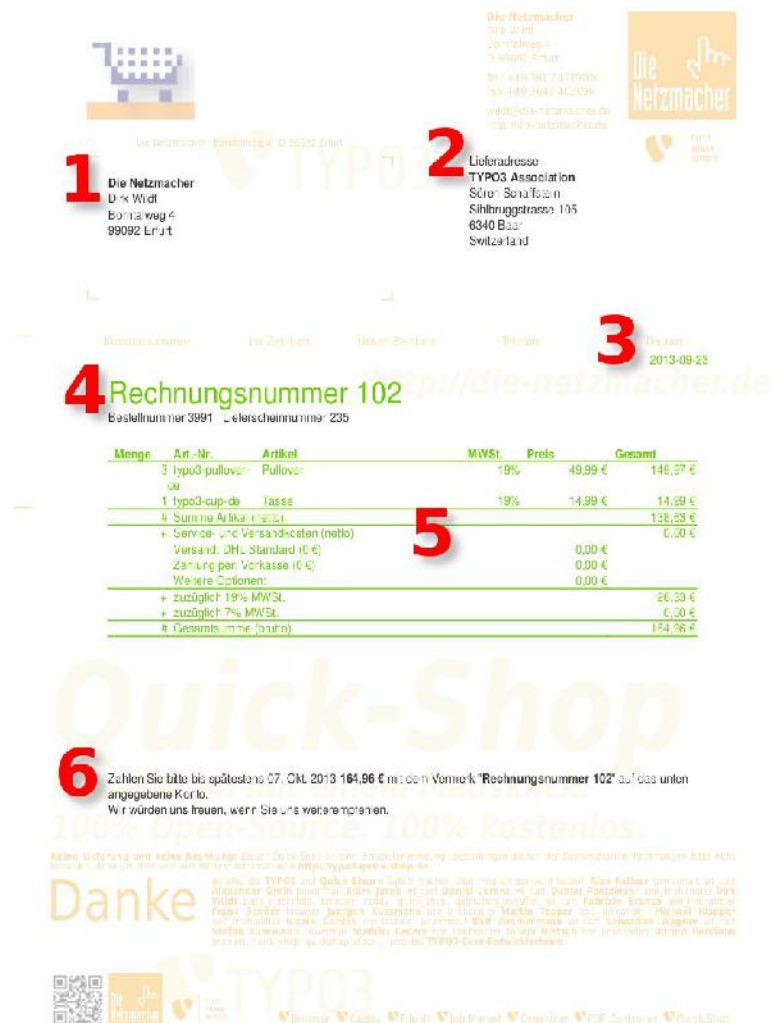


Illustration 53: Properties of a PDF attachment (here: invoice)

¹ Attachments are: invoice, delivery order, power of revocation and terms & conditions

Invoice Address

The invoice address is used in the attachments:

- invoice
- delivery order, if there isn't any different address for the delivery
- power of revocation
- terms and condition

You are controlling the invoice address in general by the Constant editor. See

- "Constant Editor > PDF INVOICE" on page 41 above.

You are controlling it for each element by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

Delivery Address

If there are different addresses for the invoice and the delivery, the delivery address is used in the attachments:

- invoice
- delivery order

You are controlling the invoice address in general by the Constant editor. See

- "Constant Editor > PDF DELIVERY ORDER" on page 40 above.

You are controlling it for each element by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

Date

The date is used in the attachments:

- invoice
- delivery order
- power of revocation
- terms and condition

You are controlling the date in general by the Constant editor. See

- "Constant Editor > PDF DATE" on page 40 above.

You are controlling it for each element by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

Numbers

You are controlling the numbers in general by the Constant editor:

- invoice
"Constant Editor > PDF INVOICE" on page 41 above.
- delivery order
"Constant Editor > PDF DELIVERY ORDER" on page 40 above.
- power of revocation
"Constant Editor > PDF REVOCATION" on page 41 above.
- terms and condition
"Constant Editor > PDF TERMS" on page 41 above.

You are controlling it for each element by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

Caddy

The caddy is used in the attachments:

- invoice

- delivery order

The caddy format depends on the attachment.

- The caddy of an invoice displays items and calculation.
- The caddy of a delivery order displays items only.

You are controlling

- position, width and font size

in general by the Constant editor:

- "Constant Editor > PDF CADDY" on page 40 above.

Term of Credit

The term of credit is used in the attachments:

- invoice
- delivery order

You are controlling it in general by the Constant editor:

- "Constant Editor > PDF INVOICE" on page 41 above.
- "Constant Editor > PDF DELIVERY ORDER" on page 40 above.

Terms and Conditions

The terms and conditions are taken from the page Terms & Conditions. See

- "Constant Editor > PAGES" on page 39 above.

The terms and conditions are used in the attachment:

- terms and conditions

You are controlling

- position,
- width and
- font size

in general by the Constant editor:

- "Constant Editor > PDF TERMS" on page 41 above.

You can control it by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

Power of Revocation

The power of revocation is taken from the page Power of Revocation. See

- "Constant Editor > PAGES" on page 39 above.

The power of revocation are used in the attachment:

- power of revocation

You are controlling

- position,
- width and
- font size

in general by the Constant editor:

- "Constant Editor > PDF REVOCATION" on page 41 above.

You can control it by TypoScript. See

- "Setup > tx_caddy_pi1.pdf" on page 75 below.

Work-flow

This section is for understanding only.

Add, update and delete an item

Add an item

A customer can add an item into the caddy.

- You have to create the form – or the link – with the GET parameters or POST parameters needed by Caddy.
- Caddy needs the uid of the item and the quantity selected by the customer.
- Caddy checks the price and the tax of the item in the table of the item.
- You have to configure the table label and some field labels by the TypoScript Constant Editor.
- Caddy stores all needed values in a session.
- You can use this data everywhere at your website i.e. with the mini caddy plugin.
- The DRS – Development Reporting System – prompts the session data to devlog.

Update an item

The customer can update the quantity of an item.

- Caddy updates the session data.

Delete an item

The customer can delete an item from the caddy.

- Caddy updates the session data.

Calculating

Caddy calculates the cost of the order.

Parts of this calculation are among others

- items x quantity x price gross
- delivery costs
- payment costs
- special costs
- sum net
- sum gross
- tax

Caddy stores all values in the session. You can use this data everywhere at your website i.e. with the mini caddy plugin.

The DRS – Development Reporting System – prompts the session data to devlog.

TypoScript

- "Setup > tx_caddy_pi1.api.marker.sum" on page 70 below.

Ordering

The customer can order the items of course.

The confirmation e-mail – the e-mail to the customer – and the order e-mail – the e-mail to the vendor – is generated and sent by Powermail.

Caddy controls Powermail:

- If the caddy is empty, it hides the Powermail form by CSS.

Caddy is listening on powermail parameters:

- GET parameters and POST parameters
- Some other "internal" parameters like
 - the Powermail confirmation mode and
 - the Powermail version (1.x or 2.x)

Caddy is listening on the Powermail session.

Caddy can detect – depending on the parameters above –, if the Powermail form is sent.

If the Powermail form is sent:

- Caddy updates all needed session data
- Caddy serves some methods (user functions), which are called by Powermail like
 - controlling attachments for the e-mails to the customer and the sender
 - serving data like the whole caddy and the numbers for the delivery order, the invoice and the order itself.
- Caddy cleans up the own session.

TypoScript

- "Setup > tx_caddy_pi1.powermail" on page 78 below.

Variants

Sorry, that variants isn't explained in detail. See below only a sample of the TYPO3 Organisier.

The TYPO3 Organisier is using the property variants for reserving tickets.

The sample is a variant of

- ticket and
- price

TypoScript Snippet

Easy, unsecure and NOT RECOMMENDED

```
plugin.tx_caddy_pil {
    db {
        sql >
        sql = TEXT
        sql {
            value (
                SELECT
                    CAST('{GP:tx_org_cal|uid}' AS SIGNED) AS puid,
                    CONCAT('<strong>', tx_org_cal.title, '</strong><br />',
                        '###GP:TX_ORG_CAL.DATETIME###<br />',
                        'Karte ', tx_org_calentrance.title, ' ', tx_org_calentrance.value, ' &euro;')
                    tx_org_calentrance.value AS title,
                    tx_org_tax.value AS gross,
                    tx_org_tax.value AS tax,
                    CAST('{GP:tx_org_cal|uid}' AS SIGNED) AS sku
                FROM `tx_org_cal`
                LEFT JOIN tx_org_cal_mm_tx_org_calentrance
                ON (
                    tx_org_cal_mm_tx_org_calentrance.uid_local = tx_org_cal.uid)
                LEFT JOIN tx_org_calentrance
                ON (
                    tx_org_cal_mm_tx_org_calentrance.uid_foreign = tx_org_calentrance.uid
                    AND tx_org_calentrance.uid = '###GP:TX_ORG_CALENTRANCE.UID###'
                    ###ENABLE_FIELDS:TX_ORG_CALENTRANCE###)
                LEFT JOIN tx_org_tax
                ON (
                    tx_org_calentrance.tx_org_tax = tx_org_tax.uid
                    ###ENABLE_FIELDS:TX_ORG_TAX###)
                WHERE tx_org_cal.uid = '###GP:TX_ORG_CAL.UID###'
                    ###ENABLE_FIELDS:TX_ORG_CAL###
            )
            insertData = 1
        }
    }
    settings {
        variant {
            10 = tx_org_calentrance.uid
        }
    }
}
```

Secure, not easy but RECOMMENDED

View:

https://forge.typo3.org/projects/extension-org/repository/entry/trunk/Configuration/TypoScript/calendar/201/tx_caddy/single/setup.ts

Download:

https://forge.typo3.org/projects/extension-org/repository/raw/trunk/Configuration/TypoScript/calendar/201/tx_caddy/single/setup.ts

```
plugin.tx_caddy_pil {
    db {
        sql >
        sql = COA
        sql {
            // SELECT
            10 = COA
            10 {
                wrap = SELECT |
                // puid: GP:tx_org_cal|uid
                10 = TEXT
                10 {
                    stdWrap {
                        data = GP:tx_org_cal|uid
                        prioriCalc = intval
                    }
                    noTrimWrap = | '|' AS puid, |
                }
                // title: CONCAT ...
                20 = COA
                20 {
                    wrap = CONCAT(|) AS title,
                    // tx_org_cal.title
                    10 = TEXT
                    10 {
                        value = tx_org_cal.title
                        noTrimWrap = | '<strong>', |, '</strong><br />', |
                    }
                    // GP:tx_org_cal|datetime
                    20 = TEXT
                    20 {
                        //value = Friday, 25. September " 2015, '01:38 Uhr
                        stdWrap {
                            data = GP:tx_org_cal|datetime
                            split {
                                // ASCII 39: quote
                                token.char = 39
                                // ASCII 32: space
                                //token.char = 32
                                cObjNum = 1
                                1.current = 1
                                1.wrap = |
                            }
                            htmlSpecialChars = 1
                        }
                        noTrimWrap = | '<br />', |
                    }
                }
                // Label for 'ticket'
                30 = TEXT
                30 {
                    value = ticket
                    lang {
                        de = Karte
                        en = ticket
                    }
                    noTrimWrap = | '|' ', |
                }
                // ... as title
                40 = TEXT
                40 {
                    value = tx_org_calentrance.title, ' ', tx_org_calentrance.value, ' &euro;'
                }
            }
        }
    }
}
```

```

    // gross: tx_org_calentrance.value
30 = TEXT
30 {
    value = tx_org_calentrance.value
    noTrimWrap = | | AS gross, |
}
    // tax: tx_org_calentrance.value
40 = TEXT
40 {
    value = tx_org_tax.value
    noTrimWrap = | | AS tax, |
}
    // sku: GP:tx_org_cal|uid
50 = TEXT
50 {
    stdWrap {
        data = GP:tx_org_cal|uid
        htmlSpecialChars = 1
    }
    noTrimWrap = | | AS sku |
    prioriCalc = intval
}
}
// FROM tx_org_cal
20 = TEXT
20 {
    value = tx_org_cal
    noTrimWrap = | FROM | |
}
// JOIN
30 = COA
30 {
    // LEFT JOIN tx_org_mm_all
10 = TEXT
10 {
    value (
        LEFT JOIN tx_org_mm_all
        ON (
            tx_org_mm_all.uid_local      = tx_org_cal.uid
            AND tx_org_mm_all.table_local = 'tx_org_cal')
    )
    noTrimWrap = | | |
}
// LEFT JOIN tx_org_calentrance
20 = COA
20 {
    wrap = LEFT JOIN tx_org_calentrance ON (|)
    // constants
10 = TEXT
10 {
    value = tx_org_mm_all.uid_foreign = tx_org_calentrance.uid AND
tx_org_mm_all.table_foreign = 'tx_org_calentrance'
    noTrimWrap = | | |
}
    // GP:tx_org_calentrance|uid
20 = TEXT
20 {
    stdWrap {
        data = GP:tx_org_calentrance|uid
        htmlSpecialChars = 1
    }
    prioriCalc = intval
    noTrimWrap = | AND tx_org_calentrance.uid = | |
}
    // ###ENABLE_FIELDS:TX_ORG_CALENTRANCE###
30 = TEXT
30 {
    value = ###ENABLE_FIELDS:TX_ORG_CALENTRANCE###
    noTrimWrap = | | |
}
}

```

```

    // LEFT JOIN tx_org_tax
30 = TEXT
30 {
    value (
        LEFT JOIN tx_org_tax
        ON (
            tx_org_calentrance.tx_org_tax = tx_org_tax.uid
            ###ENABLE_FIELDS:TX_ORG_TAX###)
    )
    noTrimWrap = | | |
}
// WHERE
40 = TEXT
40 {
    stdWrap {
        data = GP:tx_org_cal|uid
    }
    prioriCalc = intval
    noTrimWrap = | WHERE tx_org_cal.uid = | ###ENABLE_FIELDS:TX_ORG_CAL### |
}
}
}
}
settings {
    variant {
        10 = tx_org_calentrance.uid
    }
}
}

```

Non possible features

There isn't any limit for items, neither a minimum nor a maximum, in case of variants.

Archive audit proof

Caddy archives each order in a database.

Database is read-only

By default this database is read-only:

- You can't change any value in a record.
- You can't delete any record.

The database stores among others:

- All attachments, which are sent to
 - the customer
 - the vendor
- The e-mail address of the customer.

Both – read-only and the disabled delete property – is meant with audit proof¹.

Be aware:

If you need the logged data for ever, you have to take care off a backup!

Disable audit proof

You can disable the audit proof mode by the extension manager.

Possible reasons

There are some needs for disabling the audit proof:

- You have tested Caddy and you want to delete the logged data.
- Your database is growing an growing and you want to delete records and files.

Extension Manager

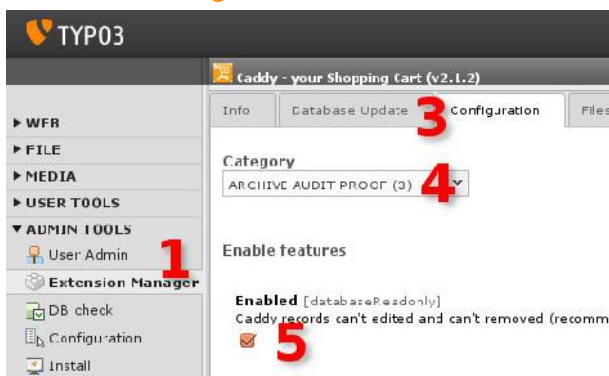


Illustration 54: Audit proof by the Extension Manager

- | | |
|------------------------|---------------------------------|
| 1. Module: Admin Tools | Extension Manager |
| 2. Edit Area | Select the Caddy extension |
| 3. | Tab [Configuration] |
| 4. | Category: [ARCHIVE AUDIT PROOF] |
| 5. | Enabled: [] |
| 6. | [Update] |

¹ German: revisionssicher

Reset numbers

Caddy is storing the numbers in the registry of TYPO3 – table sys_registry.

If you like to reset the numbers, you need a SQL command like in the examples below.

Get all keys

```
SELECT *
FROM sys_registry
WHERE entry_namespace = 'tx_caddy'
```

The result will look like this:

```
"uid","entry_namespace","entry_key","entry_value"
"561","tx_caddy","lastOrder_1279","i:1;"
"553","tx_caddy","lastOrder_15","i:4;"
"563","tx_caddy","lastOrder_7567","i:4;"
...
"626","tx_caddy","page_7843_deliveryorder","i:1;"
"627","tx_caddy","page_7843_invoice","i:1;"
"628","tx_caddy","page_7843_order","i:1;"
```

Syntax of the entry_key

The entry_key has the format:

- pageUid_KindOfNumber

where

- pageUid is
 - the uid of the page, which contains the Caddy plugin
 - format is: page_uid
- KindOfNumber is
 - deliveryorder
 - invoice
 - order

Reset (remove the keys)

All keys

```
DELETE
FROM sys_registry
WHERE entry_namespace = 'tx_caddy'
```

Keys of a single Caddy plugin

```
DELETE
FROM sys_registry
WHERE entry_namespace = 'tx_caddy'
AND entry_key LIKE 'page_7567_%'
```

where 7567 is the page uid of your Caddy plugin

Reference

It is recommended to configure Caddy by the Constant Editor and not by the TypoScript Object Browser directly.

Constant Editor

Please refer to the section "Controlling Caddy > Static Templates" on page 32 above.

Setup

It is recommended to configure Caddy by the Constant Editor and not by the TypoScript Object Browser directly.

All properties have the prefix plugin.tx_caddy_pi1.

Example: db.table is plugin.tx_caddy_pi1.db.table

tx_caddy_pi1 – Overview

Property:	Data type:	Description:
plugin.tx_caddy_pi1	array	<p>Configuration for the Caddy main plugin</p> <p>Code:</p> <pre>plugin.tx_caddy { api ... db ... debug ... pdf ... pid ... pluginCheck ... templates ... _HTMLMARKER ... _CSS_DEFAULT_STYLE ... powermail ... }</pre>

tx_caddy_pi1.api

Property:	Data type:	Description:
api	array	<p>Configuration for the Caddy API</p> <p>Code:</p> <pre>plugin.tx_caddy { api { conditions ... e-payment ... getpost ... marker ... options ... symbols ... tax ... } }</pre>

tx_caddy_pi1.api.conditions

Property:	Data type:	Description:
api.conditions	array	<p>Conditions for calculating. Array contains one condition only: the minimum limit for the gross sum over all items (without any other cost like for delivery or service)</p> <p>Code:</p> <pre>plugin.tx_caddy { api { conditions { ... <-please inspect it with the TypoScript Object Browser } } }</pre>

tx_caddy_pi1.api.e-payment

Property:	Data type:	Description:
api.e-payment	array	<p>E-Payment Interface</p> <p>Code:</p> <pre> plugin.tx_caddy { api { e-payment { ... <-please inspect it with the TypoScript Object Browser } } } </pre>

tx_caddy_pi1.api.getpost

Property:	Data type:	Description:
api.getpost	array	<p>Configuration for the Get-/Post-parameters, which are sent by GET or POST. Each item has the COA property.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > GETPOST" on page 36 above. <p>Code:</p> <pre> plugin.tx_caddy { api { getpost { gross ... min ... max ... qty ... sku ... tax ... title ... uid ... volume ... weight ... } } } </pre>

tx_caddy_pi1.api.marker.items

Property:	Data type:	Description:
api.marker.items	array	<p>Configuration of marker, which are used in HTML templates for an item.</p> <p>Example for an HTML marker:</p> <ul style="list-style-type: none"> ###SUMGROSS### <p>Please inspect the items below with the TypoScript Object Browser.</p> <p>Code:</p> <pre> plugin.tx_caddy { api { marker { items { delete = IMAGE // delete icon gross = COA // gross costs of the item min = TEXT // minimum quantity for ordering max = TEXT // maximum quantity for ordering net = COA // net costs of the item uid = TEXT // uid of the item qty = TEXT // quantity of the item service_attribute_1 = TEXT // service attribute 1 service_attribute_2 = TEXT // service attribute 2 service_attribute_3 = TEXT // service attribute 3 sku = TEXT // stock keeping unit sumgross = COA // gross costs sum (gross costs * qty) sumnet = COA // net costs sum (net costs * qty) sumtax = COA // tax costs sum (tax costs * qty) tax = TEXT // tax value of the item (0 , 1, 2) taxrate = COA // taxrate of the item title = TEXT // label of the item } } } } </pre>

tx_caddy_pi1.api.marker.sum

Property:	Data type:	Description:
api.marker.sum	array	<p>Configuration of marker, which are used in HTML templates for a sum. Example for an HTML marker:</p> <ul style="list-style-type: none"> ###SUMSUMTAXNORMAL### <p>Please inspect the items below with the TypoScript Object Browser.</p> <p>Code:</p> <pre> plugin.tx_caddy { api { marker { sum { labels { optionspaymentlabel = TEXT optionsshippinglabel = TEXT optionsspecialslabels = TEXT taxrate_reduced_string = COA taxrate_normal_string = COA } rates { optionspaymentsumrate = COA optionsshippingsumrate = COA optionsspecialssumrate = COA } values { optionspaymentsumgross = COA optionspaymentsumnet = COA optionspaymentsumtaxnormal = COA optionspaymentsumtaxreduced = COA optionsshippingsumgross = COA optionsshippingsumnet = COA optionsshippingsumtaxnormal = COA optionsshippingsumtaxreduced = COA optionsspecialssumgross = COA optionsspecialssumnet = COA optionsspecialssumtaxnormal = COA optionsspecialssumtaxreduced = COA sumitemsgross = COA sumitemsnet = COA sumitemstaxnormal = COA sumitemstaxreduced = COA sumoptionsgross = COA sumoptionsnet = COA sumoptionstaxnormal = COA sumoptionstaxreduced = COA sumsumgross = COA sumsumnet = COA sumsumtaxnormal = COA sumsumtaxreduced = COA sumsumtaxsum = COA } } } } } </pre>

tx_caddy_pi1.api.options

Property:	Data type:	Description:
api.options	array	<p>Configuration of the options</p> <ul style="list-style-type: none"> • payment, • shipping, • specials <p>You are controlling the calculating and the output in HTML, e-mail and PDF.</p> <p>Please inspect the items below with the TypoScript Object Browser.</p> <p>Constant Editor: See "Options" on page 38 above.</p> <p>Details and Examples: See</p> <ul style="list-style-type: none"> • "Calculating" on page 44 above • "tx_caddy_pi1.api.options.payment" on page 72 below <p>Code:</p> <pre> plugin.tx_caddy { caddy { options = // payment, shipping, special options { payment = // preset, options payment { preset = 1 // default method options = // cash in advance, invoice, cash on delivery options { ... } } shipping = // preset, options shipping { preset = 1 // default method options = // maildefault, mailexpress options { ... } } specials = // preset, options specials { divider = TEXT // divider for list of options // in e-mails und PDF attachments options = // neutralpackage, addfeeforislands options { ... } } } } } </pre>

tx_caddy_pi1.api.options.payment

Property:	Data type:	Description:
api.options.payment	array	<p>Constant Editor: See "Options" on page 38 above.</p> <p>Example:</p> <pre> plugin.tx_caddy { caddy { options = // payment, shipping, special options { payment = // preset, options payment { preset = 1 // default method options = // cash in advance, invoice, cash on delivery options { 1 = cash in advance (de: Vorkasse) 1 { enabled = {\$plugin.caddy.options.payment.1.enabled} // label, gross, cash discount title = COA title { ... // label, gross 10 = COA 10 { 10 = TEXT ... } // amount 20 = COA 20 { ... } } // cash discount 20 = COA 20 { ... } } } // on, percent cash-discount = cash-discount { // items, payment, service, shipping on = on { // [Boolean] true: get discount on items items = 1 // [Boolean] true: get discount on payment payment = 0 // [Boolean] true: get discount on shipping shipping = 0 // [Boolean] true: get discount on specials specials = 0 } // [Double] percent = {\$plugin.caddy.options.payment.1.cashdiscount} } // extra cost (gross price) of payment method extra = {\$plugin.caddy.options.payment.1.costs} // tax rate which will be applied (reduced or normal) tax = {\$plugin.caddy.options.payment.1.taxrate} } 2 ... } } } </pre>

tx_caddy_pi1.api.options.shipping

See "tx_caddy_pi1.api.options.payment" on page 72 above.

tx_caddy_pi1.api.options.specials

See "tx_caddy_pi1.api.options.payment" on page 72 above.

tx_caddy_pi1.api.symbols

Property:	Data type:	Description:
api.symbols	array	<p>Configuration of symbols.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > SYMBOLS AND SERVICE ATTRIBUTES" on page 42 above. <p>Example:</p> <pre> plugin.tx_caddy { api { symbols { decimal = 2 dec_point = , thousands_sep = . currencySymbol = &euro; currencySymbolBeforePrice = 0 quantitySymbol = St. service_attribute_1_symbol = service_attribute_2_symbol = service_attribute_3_symbol = } } } </pre>

tx_caddy_pi1.api.tax

Property:	Data type:	Description:
api.tax	array	<p>Configuration for tax calculating.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > TAX" on page 42 above. <p>Example:</p> <pre> plugin.tx_caddy { api { tax { reduced = 7 reducedCalc = 0.07 normal = 19 normalCalc = 0.19 } } } </pre>

tx_caddy_pi1.db

Property:	Data type:	Description:
db	array	<p>Interface for your database.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > DATABASE" on page 35 above. <p>Example:</p> <pre> plugin.tx_caddy { db { table = tx_quickshop_products title = title gross = price tax = tax sku = sku min = quantity_min max = quantity_max service_attribute_1 = service_attribute_2 = service_attribute_3 = sql { ... -> See tx_caddy_pi1.db.sql below } } } </pre>

tx_caddy_pi1.db.sql

Property:	Data type:	Description:
db.sql	array	<p>Extend the database interface for special needs – usually variants.</p> <p>Example:</p> <pre> plugin.tx_caddy { db { sql { ... -> See Error: Reference source not found on page Error: Reference source not found Error: Reference source not found } } settings { ... -> See Error: Reference source not found on page Error: Reference source not found Error: Reference source not found } } </pre>

tx_caddy_pi1.debug

Property:	Data type:	Description:
debug	boolean	<p>Prompts a debug report with informations about TypoScript to the frontend.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > DEBUG" on page 35 above. <p>Example:</p> <pre> plugin.tx_caddy { debug = 0 } </pre>

tx_caddy_pi1.pdf

Property:	Data type:	Description:
pdf	array	<p>Configuration for the PDF attachments.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > PDF" on page 40 above. <p>Please inspect the items below with the TypoScript Object Browser.</p> <p>Code:</p> <pre> plugin.tx_caddy { pdf { deliveryorder { filename = COA // label - delivery order number - date . pdf content { address = // deliveryorder, invoice caddy = // body date = // body numbers = // deliveryorder, invoice and order } } invoice { filename = COA // label - invoicenumber - date . pdf content { address = // deliveryorder, invoice caddy = // body date = // body numbers = // deliveryorder, invoice and order termOfCredit = // body } } revocation { filename = COA // label - invoicenumber - date . pdf content { address = // invoice date = // body additionaltextblocks = // revocation } } terms { filename = COA // label - invoicenumber - date . pdf content { address = // invoice date = // body additionaltextblocks = // terms } } } } </pre>

tx_caddy_pi1.pid

Property:	Data type:	Description:
pid	integer	<p>Id of the page with the CADDY plugin.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > PAGES" on page 39 above. <p>Example:</p> <pre> plugin.tx_caddy { pid = 8154 } </pre>

tx_caddy_pi1.pluginCheck

Property:	Data type:	Description:
pluginCheck	string	<p>For self evaluation only. Never touch it!</p> <p>Code:</p> <pre>plugin.tx_caddy { pluginCheck = dummy }</pre>

tx_caddy_pi1.templates

Property:	Data type:	Description:
templates	array	<p>Configuration of templating.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > TEMPLATES" on page 43 above. <p>Code:</p> <pre>plugin.tx_caddy { templates { e-mail { file = EXT:caddy/files/templates/caddy.html marker { // all, item } table { // border, cellpadding, cellspacing } } html { caddy { // pi1: file, marker, table } caddymini { // pi3: file, marker } caddysum { // pi2: file, marker } form { order { foundation { 5x { wiSelect = COA wiSelect { ... } woSelect = COA woSelect { ... } } } } } } pdf { deliveryorder { // file, marker, table } invoice { // file, marker, table } revocation { // file, marker, table } terms { // file, marker, table } } } }</pre>

tx_caddy_pi1._HTMLMARKER

Property:	Data type:	Description:
_HTMLMARKER	array	<p>Feel free to add any marker to the templates for HTML, the e-mail and the PDF attachments.</p> <p>The property from below will replace ###_HTMLMARKER_LINTOSHOP###.</p> <p>Please inspect the code from below with the TypoScript Object Browser.</p> <p>Example:</p> <pre> plugin.tx_caddy { _HTMLMARKER { linktoshop = COA linktoshop { // Label 10 = TEXT 10 { ... } // Icon 20 = IMAGE 20 { ... } } wrap = <div class="linktoshop caddylinktoshop"> </div> } } </pre>

tx_caddy_pi1._CSS_DEFAULT_STYLE

Property:	Data type:	Description:
_CSS_DEFAULT_STYLE	string	<p>Inline CSS. An inline CSS can configured while runtime.</p> <p>This property is configured only, if a CSS static template is included. See</p> <ul style="list-style-type: none"> "Static Templates > CSS" on page 32 above. <p>Please inspect the code from below with the TypoScript Object Browser.</p> <p>Code:</p> <pre> plugin.tx_caddy_pi1 { _CSS_DEFAULT_STYLE (form.caddy { text-align: right; } form.caddy button.submit img { position: relative; top: .1em; } ...) } </pre>

tx_caddy_pi1.powermail

Property:	Data type:	Description:
powermail	array	<p>Configuration for controlling powermail.</p> <p>This property is configured only, if a powermail static template is included. See</p> <ul style="list-style-type: none"> "Static Templates > Powermail (obligated)" on page 33 above. <p>The configuration depends on the powermail version 1.x and 2.x</p> <p>The example below corresponds with the fluid marker in the powermail form.</p> <p>Code (powermail 2.x):</p> <pre> plugin.tx_caddy { powermail { caddy = USER caddy { // The caddy content } caddydeliveryordernumber = USER caddydeliveryordernumber { // Delivery order number } caddyinvoicenummer = USER caddyinvoicenummer { // Invoice number } caddyordernumber = USER caddyordernumber { // Order number } clearcaddysession = USER clearcaddysession { // Clear the caddy session } } } </pre>

tx_caddy_pi2

Frontend and Plugin

- See "Plugins > Sum Gross" on page 24 above.

TypoScript

Property:	Data type:	Description:
plugin.tx_caddy_pi2	array	<p>Configuration for the Caddy Sum Gross plugin</p> <p>Code:</p> <pre> plugin.tx_caddy_pi2 { // item, marker content = content { // label, sumsumgross sum = COA sum { // label 20 = TEXT 20 { data = LLL:EXT:caddy/pi2/locallang.xml:phrasesum noTrimWrap = } // sumsumgross 30 = COA 30 { // value 10 = USER 10.userFunc = tx_caddy_userfunc->numberformat 10.userFunc { number = TEXT number.field = sumsumgross decimal = 2 dec_point = {\$plugin.caddy.main.dec_point} thousands_sep = {\$plugin.caddy.main.thousands_sep} drs = {\$plugin.caddy.debug.userfunc} } // currency 20 = TEXT 20 { value = {\$plugin.caddy.main.currencySymbol} noTrimWrap = } wrap = } } } // page id of the caddy pid = {\$plugin.caddy.pages.caddy} // e-mail, html, pdf templates = templates < plugin.tx_caddy_pi1.templates } </pre>

tx_caddy_pi3 – Overview

Frontend and Plugin

- See "Plugins > Mini Caddy" on page 22 above.

TypoScript

Property:	Data type:	Description:
plugin.tx_caddy_pi3	array	<p>Configuration for the Caddy Mini plugin</p> <p>Code:</p> <pre> plugin.tx_caddy_pi3 { // item, marker content = content { // quantity, label, gross item = COA item { ... } // quantity, label, gross sum = COA sum { ... } } // page id of the caddy pid = {\$plugin.caddy.pages.caddy} // e-mail, html, pdf templates = templates < plugin.tx_caddy_pi1.templates // linktocaddy, linktoshop _HTMLMARKER = _HTMLMARKER { ... } } </pre>

tx_caddy_pi3.content.item

Property:	Data type:	Description:
content.item	array	<p>Configuration for the items of the Caddy Mini plugin</p> <p>Code:</p> <pre> plugin.tx_caddy_pi3 { // item, marker content = content { // quantity, label, gross item = COA item { // quantity 10 = TEXT 10 { field = quantity wrap = <div class="quantity itemquantity"> </div> } // label 20 = TEXT 20 { field = label wrap = <div class="label itemlabel"> </div> } // gross 30 = COA 30 { // value 10 = USER 10.userFunc = tx_caddy_userfunc->numberformat 10.userFunc { number = TEXT number.field = gross decimal = 2 dec_point = {\$plugin.caddy.main.dec_point} thousands_sep = {\$plugin.caddy.main.thousands_sep} drs = {\$plugin.caddy.debug.userfunc} } // currency 20 = TEXT 20 { value = {\$plugin.caddy.main.currencySymbol} noTrimWrap = } wrap = <div class="gross itemgross"> </div> } wrap (<div class="item minicaddyitem"> </div> <div class="caddy_cleaner"></div>) } } } </pre>

tx_caddy_pi3.content.sum

Property:	Data type:	Description:
content.sum	array	<p>Configuration for the Sum of the Caddy Mini plugin</p> <p>Code:</p> <pre> plugin.tx_caddy_pi3 { // item, marker content = content { // quantity, label, gross sum = COA sum { // quantity 10 = TEXT 10 { field = quantity wrap = <div class="quantity sumquantity"> </div> } // label 20 = COA 20 { // in case of one item 10 = TEXT 10 { if { value = 2 isLessThan { field = quantity } } data = LLL:EXT:caddy/pi3/locallang.xml:item } // in case of more than one item 20 = TEXT 20 { if { value = 2 isLessThan { field = quantity } negate = 1 } data = LLL:EXT:caddy/pi3/locallang.xml:items } wrap = <div class="label sumlabel"> </div> } // gross 30 = COA 30 { 10 = USER 10.userFunc = tx_caddy_userfunc->numberformat 10.userFunc { number = TEXT number.field = gross decimal = 2 dec_point = {\$plugin.caddy.main.dec_point} thousands_sep = {\$plugin.caddy.main.thousands_sep} drs = {\$plugin.caddy.debug.userfunc} } 20 = TEXT 20 { value = {\$plugin.caddy.main.currencySymbol} noTrimWrap = } wrap (<div class="gross sumgross"> </div> <div class="caddy_cleaner"></div>) } wrap = <div class="sum minicaddysum"> </div> } } } </pre>

tx_caddy_pi3.pid

Property:	Data type:	Description:
pid	integer	<p>Id of the page with the Caddy plugin.</p> <p>Please configure it with the Constant Editor. See</p> <ul style="list-style-type: none"> "Constant Editor > PAGES" on page 39 above. <p>Code:</p> <pre>plugin.tx_caddy_pi3 { // page id of the caddy pid = {\$plugin.caddy.pages.caddy} }</pre>

tx_caddy_pi3.templates

Property:	Data type:	Description:
templates	array	<p>Configuration of the templates is taken from the Caddy plugin.</p> <p>Code:</p> <pre>plugin.tx_caddy_pi3 { // e-mail, html, pdf templates = templates < plugin.tx_caddy_pi1.templates }</pre>

tx_caddy_pi3._HTMLMARKER.linktocaddy

Property:	Data type:	Description:
_HTMLMARKER. linktocaddy	array	<p>Link to the page with the Caddy plugin. Link contains an image – the Caddy icon.</p> <p>Code:</p> <pre> plugin.tx_caddy_pi3 { // linktocaddy, linktoshop _HTMLMARKER = _HTMLMARKER { // label, icon. Replaces _HTMLMARKER_LINTOCADDY linktocaddy = COA linktocaddy { // Label 10 = TEXT 10 { data = LLL:EXT:caddy/pi3/locallang.xml:linktocaddy typolink { parameter = {\$plugin.caddy.pages.caddy} title { data = LLL:EXT:caddy/pi3/locallang.xml:linktocaddy } } noTrimWrap = } // Icon 20 = IMAGE 20 { file = {\$plugin.caddy.html.color.icon.caddy} altText { data = LLL:EXT:caddy/pi3/locallang.xml:caddy } titleText { data = LLL:EXT:caddy/pi3/locallang.xml:caddy } params = class="linktocaddy" imageLinkWrap = 1 imageLinkWrap { enable = 1 typolink { parameter = {\$plugin.caddy.pages.caddy} } } } } wrap = <div class="linktocaddy minicaddylinktocaddy"> </div> } } </pre>

tx_caddy_pi3._HTMLMARKER.linktoshop

Property:	Data type:	Description:
_HTMLMARKER. linktoshop	array	<p>Link to the page with the Shop plugin. Link contains an image – the Caddy icon.</p> <p>Code:</p> <pre> plugin.tx_caddy_pi3 { _HTMLMARKER { // label, icon. Replaces _HTMLMARKER_LINTOSHOP linktoshop = COA linktoshop { // Label 10 = TEXT 10 { data = LLL:EXT:caddy/pi3/locallang.xml:linktoshop typolink { parameter = {\$plugin.caddy.pages.shop} title { data = LLL:EXT:caddy/pi3/locallang.xml:linktoshop } } noTrimWrap = } // Icon 20 = IMAGE 20 { file = {\$plugin.caddy.html.color.icon.caddy} altText { data = LLL:EXT:caddy/pi3/locallang.xml:shop } titleText { data = LLL:EXT:caddy/pi3/locallang.xml:shop } params = class="linktoshop" imageLinkWrap = 1 imageLinkWrap { enable = 1 typolink { parameter = {\$plugin.caddy.pages.shop} } } } } wrap = <div class="linktoshop minicaddylinktoshop"> </div> } } </pre>

Debugging

If you have any unexpected result, you can inspect Caddy in the TYPO3 backend with the DRS – the Development Reporting System – and in the TYPO3 frontend with Firebug.

DRS – Development Reporting System

Inspect work-flow, session data and errors

The DRS – Development Reporting System – enables you

- to follow the work-flow of Caddy,
- to inspect alternate configuration possibilities,
- to inspect session data and their corresponding cObject fields,
- to inspect improper configuration and last but not least
- the DRS prompts warnings and errors

Enabling by the Extension Manager

Enable the DRS – Development Reporting System – for all caddy plugins by the Extension Manager. If you have more than one plugin on a page (like a caddy and a mini caddy), you will the report for each item. It is recommended to enable the DRS by the Plugin / Flexform (see below).

BE AWARE: This feature can't debug user functions. For a complete report, you have to enable this feature by the Constant Editor (see below).

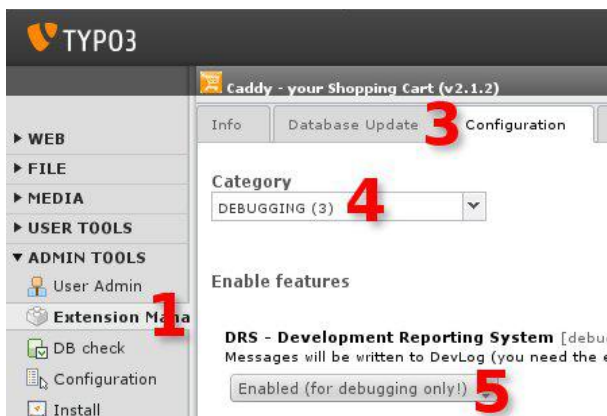


Illustration 55: Enable the DRS by the Extension Manager

1. Module: Admin Tools Extension Manager
2. Edit Area Select the Caddy extension
3. Tab [Configuration]
4. Category: [Debugging]
5. DRS – Development Reporting System: [Enabled]
6. [Update]

Enabling by the Plugin / Flexform (recommended)

Enable the DRS – Development Reporting System – for the current plugin only by the plugin / flexform.

BE AWARE: This feature can't debug user functions. For a complete report, you have to enable this feature by the Constant Editor (see below).

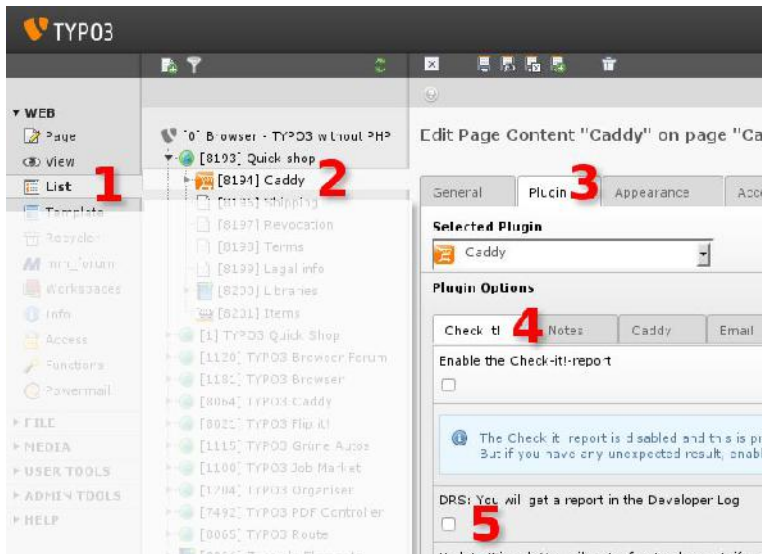


Illustration 56: Enable the DRS by the Plugin / Flexform

1. Module: Web List
2. Page tree Page with the Caddy plugin (here: [8194] Caddy)
3. Edit Area Caddy plugin: tab [Plugin]
4. Tab [Check it!]
5. DRS: You will get a report ... [X]

Enabling by the Constant Editor (recommended)

Enable the DRS – Development Reporting System – for user functions.

BE AWARE: This feature debugs user functions only. For a complete report, you have to enable the DRS by the plugin / flexform or the extension manager too.

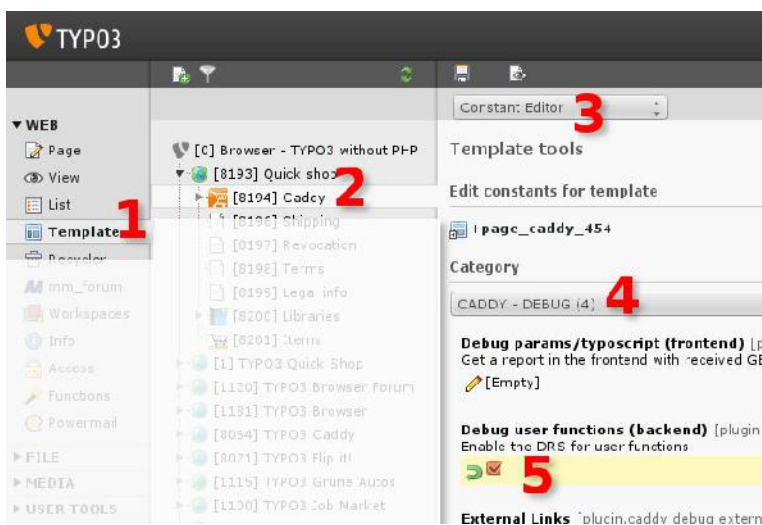


Illustration 57: Enable the DRS by the Plugin / Flexform

1. Module: Web Template
2. Page tree Page with the Caddy plugin (here: [8194] Caddy)
3. Edit Area [Constant Editor]
4. category [CADDY - DEBUG]
5. Debug user functions (backend) [X]

The DRS report

Precondition

You need the extension

- Developer Log (devlog)

Report

	Date	Severity	Extension	Message	Called from	Page	User	Extra data
WEB	24-09-13 00:10:20	INFO	caddy	[INFO/DRS] The DRS - Development Reporting System is enabled by the flexform (frontend mode).	class.tx_caddy_drs.php, line 180		Waren	crb
FILE	24-09-13 00:10:20	INFO	caddy	[INFO/DRS] "Warenkorb" (pid: 8148, uid: 2505)	class.tx_caddy_drs.php, line 165		Waren	crb
MEDIA	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] pid is taken from main.php and is set to 8148	class.tx_caddy_pid.php, line 767		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	WARN	caddy	[WARN/INIT] No access: current IP doesn't match the list of allowed IPs. Result won't prompt to the frontend.	class.tx_caddy_pid.php, line 113		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/POWERMAIL] Powermail version is 2.0.9 (internal: 2000003)	class.tx_caddy_powermail.php, line 878		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/SQL] SELECT * FROM tx_content WHERE pid = 8148 AND hidden = 0 AND deleted = 0 AND tx_type = 'powermail' ORDER BY sorting LIMIT 1	class.tx_caddy_powermail.php, line 943		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/POWERMAIL] powermailid: '2537'	class.tx_caddy_powermail.php, line 846		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/POWERMAIL] powermailtitle: "Powermail für Quickshop"	class.tx_caddy_powermail.php, line 848		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/POWERMAIL] powermailconfirm: "E"	class.tx_caddy_powermail.php, line 850		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] tx_quickshop_products is an element of the \$TCA.	class.tx_caddy_pid.php, line 516		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] Delivery order number from registry: 19	class.tx_caddy_pid.php, line 588		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] Order number from registry: 19	class.tx_caddy_pid.php, line 590		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] Invoice number from registry: 19	class.tx_caddy_pid.php, line 592		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] Current delivery order number: 236	class.tx_caddy_pid.php, line 594		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] Current invoice number: 203	class.tx_caddy_pid.php, line 596		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] Current order number: 3992	class.tx_caddy_pid.php, line 598		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/INIT] Current numbers are stored in the session.	class.tx_caddy_pid.php, line 602		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/VARIANTS] settings.variant is empty: Variants aren't handled.	class.tx_caddy_session.php, line 2113		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/CALC] Case: add an item	class.tx_caddy_session.php, line 1102		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/CALC] Current item (Geheime Bilder: 4) hasn't any maximum limit. Maximum limit won't be checked.	class.tx_caddy_session.php, line 1145		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/CALC] Current item (Geheime Bilder: 4) hasn't any minimum limit. Minimum limit won't be checked.	class.tx_caddy_session.php, line 1211		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/CALC] Quantity of current product:	class.tx_caddy_session.php, line 1499		Waren	crb
ADMIN TOOLS	24-09-13 00:10:20	INFO	caddy	[INFO/CALC] Quantity: a1 #0 - add #1 = sum #1	class.tx_caddy_session.php, line 1499		Waren	crb

Illustration 58: The DRS Report

Get the report:

- Module > Admin Tools: Development Log

Upgrades/Updates

From Caddy 6.x to 6.3

Foundation Template must included again

[6.3.0]

Path to Powermail static template is moved.

You have to include the static template again once: Caddy [6] + Powermail 2.x Foundation

From Caddy 4.x to 6.x

Check Constants

There are new constants [CADDY – URL]. Please check, if your configuration is proper. See "URL" on page 43 above.

You have to update your static templates before. See next section.

Include static templates again

Static templates are modular now. You have to include the static templates again.

See samples in the section "Caddy" on page 14 above

From Caddy 2.x/3.x to 4.x

HTML templates are moved from files/templates to files/templates/x.0.0

If you are using the CADDY templates "without options" or "without options and without fee", you have to update the path from

- files/templates

to

- files/templates/2.0.0 or
- files/templates/4.0.0¹ or

SEE TypoScript Constant Editor > CADDY - TEMPLATES

From Powermail 1.x to Powermail 2.x

Caddy supports both versions: Powermail 1.x and Powermail 2.x.²

If you have made an update from 1.x to 2.x, please evaluate your Caddy configuration with the tab [Check it!]. See

- "Plugin / Flexform > Check it!" on page 26 above.
- "Plugin / Flexform > Billing Address" on page 29 above.
- "Plugin / Flexform > Delivery Address" on page 30 above.

From wt_cart 1.x to Caddy 2.x

- Please install Caddy with one of the both one-click-installer – Organiser or Quick Shop. See
 - "One-Click-Installation (recommended)" on page 10 above.
- Than move the caddy page to the page with the wt_cart plugin.
- Adapt the Caddy configuration to your needs.
- Remove wt_cart.

¹ Currently these templates are available in 2.0.0 only

² If you are using TYPO3 6.x, powermail version must be 2.x because 1.x version doesn't work with different languages with 6.1. Powermail 2.x seems to be compulsory if the site is not in English.

What's new?

Version 6

Bootstrap

[6.3.0]

Caddy provides ready-to-use templates for Bootstrap.

Button template

[6.0.0]

The Caddy Button "Buy now" is now a template. Should simplify to integrate Caddy in the HTML code of any extension. See "tx_caddy_pi1.templates" on page 76 above.

Foundation 5.x

[6.0.0]

Upgrade for Caddy, Paymill and Powermail. New static templates:

- Caddy [3] + Foundation 5.x
- Caddy [3.1] + Foundation 5.x CSS

Mini Caddy

[6.1.0]

Mini Caddy icon. Include static templates for dropdown button and reveal mode (dialog box).

Paymill upgrade

[6.0.0]

- Upgrade for Foundation 5.x
- New test data since Aug. 2014

New static template:

- Caddy [8] + E-Payment: Paymill Foundation 5.x

Powermail upgrade

[6.0.0]

Upgrade of templates and partials for Foundation 5.x. New static template:

- Caddy [6] + Powermail 2.x Foundation 5.x

TYPO3 6.2

[6.0.0]

Upgrade to TYPO3 6.2.

TypoScript file structure

[6.0.0]

TypoScript file is modular. New file structure should simplify maintenance.

Version 4

E-Payment

[4.0.5]

Interface for e-payment. See "E-Payment" on page 16 above.

Options are configurable

[4.0.5]

If you like to change the default options for payment, shipping and specials, you don't need to configure TypoScript directly any longer.

See "Options" on page 38 above.

Pid for Caddy

[4.0.3]

You can use the Caddy plugin on different pages with the same content. This is useful for development among others: You can use Caddy with different configurations on page A and page B. Both plugins must have the same page id at

```
plugin.caddy {
    pages.caddy =
}
```

See "PAGES" on page 39 above.

Responsive

[4.0.5]

Caddy has a responsive design. See "Responsive Design" on page 13 above.

Inventory Control

[4.0.7]

Caddy can control your inventory. If your items have the properties "quantity in stock" and "stock management".

If inventory control is enabled:

- Caddy will decrease the quantity of your item after an order.
- The quantity in stock is the maximum order quantity of an item.
- Items can't ordered, if they aren't in stock.

[CADDY – INVENTORY CONTROL]

See **XXX**

TypoScript

[4.0.5]

###TABLE.FIELD### replaced by fields

All so called table.field-markers are removed from the static template. They are no longer needed.

Former TypoScript

```
my_table {
    my_field = TEXT
    my_field {
        value = ###MY_TABLE.MY_FIELD###
    }
}
```

Current TypoScript

```
my_table {
    my_field = TEXT
    my_field {
        field = my_table.my_field
    }
}
```

Structure cleaned up

The TypoScript structure is cleaned up:

```

plugin.tx_caddy_pi1
  • conditions -> moved to: api.conditions
  • getpost    -> moved to: api.getpost
  • marker     -> moved to: api.marker
  • options    -> moved to: api.options
  • symbols    -> moved to: api.symbols
  • tax        -> moved to: api.tax

```

See "tx_caddy_pi1.api" on page 67 above.

Packages for Sponsors

Your donation and sponsoring is welcome.

Please order a package for sponsors at

- English: <http://typo3-caddy.de/en/sponsoring/>
- German: <http://typo3-caddy.de/sponsoring/>



Fun – 50 EUR



You like to treat the TYPO3 Caddy project.

You don't expect any equivalent. No logo. No service.

Throw 50 EUR or any other old sum into the cash box.

You will get a receive. You can set the amount off against tax liability.

Logo – 150 EUR



You are treating the TYPO3 Caddy project and you like to be a guide for others.

Your logo will appear in the manual, at the website, in the plugin and in the extension manager of the Caddy.

The period of validity is limited to one year.

You will get a receive. You can set the amount off against tax liability.

You can pay more than 150 EUR.

Patron – 500 EUR



You are the patron. You are treating TYPO3 Caddy project and you like to pay the developer more than best for two hours service.

Your logo will appear in the manual, at the website, in the plugin and in the extension manager of Caddy.

You can book the developer of Caddy for two hours.

The period of validity for both – the logo and the two hours – is limited to one year. If you don't use the two hours in this period, you will lost it.

You will get a receive. You can set the amount off against tax liability.

You can pay more than 500 EUR.

Credits

Made in Nature

The German TYPO3 agency Made in Nature has contracted

- a part of the upgrade to 6.0.0
- the E-Payment interface
- PDF attachments for the delivery order and the invoice¹
- Responsive Design by the Foundation Framework
- a part of the fork.

Website

- <http://made-in-nature.de/>

wt_cart Developer Team

Thanks to the wt_cart developer team for the great wt_cart extension, which was the base for Caddy:

- extco.de UG, Daniel Lorenz (Leader) – www.extco.de
- TRITUM GmbH, Björn Jacob (Co-Leader), Sebastian Wagner, Ralf Zimmermann – www.tritum.de
- in2code, Alexander Kellner – www.in2code.de

¹ It is realised with wt_cart 1.4.x in 2011 but never published :(

Known Bugs

PDF Attachments without any address data

Event

PDF Attachments are without any address data.

Reason and Solution

Probably the powermail plugin was never saved. Please save it once. PDF attachments should contain address data from now.

If error occurs again, please investigate the DRS report. See

- DRS – Development Reporting System on page 86 above.

Required argument "mail" is not set

Prompt

TYPO3 system prompt:

Oops, an error occurred!

Required argument "mail" is not set.

More information regarding this error might be available online.

Reason and solution

Caddy tried to send an e-mail without an e-mail-address or with an improper e-mail-address.

You can check this by investigate the latest saved caddy record: probably e-mail is missing in the form of the record.

Cause should be a mismatch in the configuration of the powermail plugin. Check it!

TCPDF-Error

Prompt

TCPDF ERROR: This document (fileadmin/files/my.pdf) probably uses a compression technique which is not supported by the free parser shipped with FPDF.

Reason and Solution

Probably the version of your PDF file is the reason: Version must be lower than 1.5.

More Bugs

Please refer to

- <http://forge.typo3.org/projects/extension-caddy/issues>

Further Information

Forum

Caddy has a website and a forum. Please visit:

- <http://typo3-caddy.de/en/>
- <http://forum.typo3-caddy.de/>

TYPO3 forge

- <http://forge.typo3.org/projects/extension-caddy>

History of Caddy

Fork of wt_cart

Caddy was a fork of wt_cart version 1.4.6.

Copyright wt_cart 1.4.6: 2010-2012, wt_cart Development Team. See "wt_cart Developer Team" on page 94 above.

Why do I forked wt_cart in 2013?

A generic shopping cart for TYPO3 is a good idea.

And an improved idea is

- to fix bugs speedy,
- to develop the cart permanently, and
- to let profit the TYPO3 community of contracted features.

Caddy is the attempt, to keep the promise from above.

Caddy 2.x versus wt_cart 1.x

Having regard to plugin 1

skill	caddy 2.x	wt_cart 1.4
Features		
Items from your database	yes	yes
Min/Max limit for each item	yes	yes
Min/Max limit for all items	yes	no
Delivery costs depending on <ul style="list-style-type: none"> amount size weight 	yes yes yes	yes yes yes
PDF attachments ¹	yes	no
Order-/Invoice-numbers without gap	yes	no
Variants ²	yes	buggy
Interface for online payment	no	no
Configuration properties		
Plugin / Flexform ³ , Constant Editor	153 properties	42 properties
Database		
Orders are stored in an own database	yes	no
Archive audit proof ⁴	yes	no
Debugging and Development		
Developer versions	http://forge.typo3.org/.../caddy	https://github.com/extcode/wt_cart
GET/POST parameter	yes	yes
TypoScript	yes	yes
User functions	yes	no
Work flow	nearby 300 prompts ⁵	0 prompts
Evaluation		
by the Extension Manager	yes	no
by the plugin / flexform	yes	no
Icons		
Cart	1 icon / 40 colours	no icon
Delete	1 icon / 40 colours	1 icon / 1 colour
Installation		
Out of the box / one click installation	yes	stopped with Quick Shop installer 2.x
Manual		
Illustrations	42	8
Pages	77	30
Powermail		
Compatible with 1.x	yes	yes
Compatible with 2.x	yes	no
Independent on Powermail hooks	yes	no
Support		
Website	http://typo3-caddy.de/	http://wt-cart.com/
Forum / Support	http://forum.typo3-caddy.de/	http://snippets.wt-cart.com/

¹ Delivery order, invoice, terms & conditions








² One dimension

³ Plugin 1

⁴ Audit proof in German: revisionssicher

⁵ supported by the DRS – Development Reporting System

Other extensions published by Die Netzmacher

-  +AOE Linkhandler Configurator: Configure the AOE linkhandler supported by userinterfaces. Out-of-the-box templates for cal, org, tt_news and tt_products. Don't edit page TSconfig any longer.
<http://typo3.org/extensions/repository/view/linkhandlerconf/>
-  autositemap: A smart site-map optimised for the footer. It groups menus in columns. Great menus will get two columns. Configuration is based on TypoScript HMENU.
<http://typo3.org/extensions/repository/view/autositemap/>
-  Browser – TYPO3 without PHP. Develop your TYPO3 extension 8 times faster! You need one line TypoScript for a result list with a search form, a record browser and an index browser. Images are wrapped self-acting. SEO, Search Engine Optimization, is integrated. <http://typo3.org/extensions/repository/view/browser/>
-  Caddy – the TYPO3 shopping cart. You can use it for your own needs. You need a database with products or any other items only. You can install Caddy out of the box with one mouse click – see Quick Shop below.
<http://typo3.org/extensions/repository/view/caddy/>
-  Deal! – TYPO3 Quick Shop interface for the marketplace ebay. Manage all items with TYPO3 and publish it on ebay. Amazon is under construction.
<http://typo3.org/extensions/repository/view/deal/>
-  Flip it! offers lovely and smooth page flip transitions. It enables you to run over pages in PDF documents like in a real magazine. It is based on flash. Flipt it! can convert PDF documents to swf files automatically.
<http://typo3.org/extensions/repository/view/flipit/>
-  Green Cars (Grüne Autos) - Database optimized for ecological cars. It is a case study and demonstrates, how to get a complex database with the browser (see above) in three hours only.
http://typo3.org/extensions/repository/view/green_cars/
-  Jobmarket is a catalogue with job offers. Views, the a-z-browser, the page-browser, the search, social bookmarks and a lot of other stuff can configured by the Browser plugin with the mouse. Job Market hasn't any PHP code, it should be easy to adapt it to your needs.
http://typo3.org/extensions/repository/view/job_market/
-  Majordomo: For subscribing to and unsubscribing from a majordomo mailing list.
<http://typo3.org/extensions/repository/view/majordomo/>
-  Organiser – TYPO3 for the lobby and the organisers. Handle news, events, staff, headquarters, locations, workshops and a calendar with one extension. Sell online tickets! Install the Organiser with one mouse click!
<http://typo3-organiser.de/>
-  Quick Shop – the fastest shop in the history of TYPO3. Install it with one mouse click! Quick Shop is based on the browser (see above) and powermail.
<http://typo3-quick-shop.de/>
-  PDF Controller: Easy to install. Add to your HTML page the PDF-controller-button. Link from the button to the controller. Adjust the controller by mouseclicks. The PDF Controller supports CSS 3.
<http://typo3-pdfcontroller.de/>
-  Radial Search (Umkreissuche) – Enable the browsing through your data by geo location. 'Your data must have a latitude and a longitude. Address data can geocoded automatically by the Browser - TYPO3 without PHP.
<http://typo3.org/extensions/repository/view/radialsearch/>
-  Route – Publish your routes with GoogleMaps or OpenStreetMap. Routes have points of interest (POI). You can categorise and filter both: routes and POI. Address data can geocoded automatically.
<http://typo3.org/extensions/repository/view/route/>
-  seo_dynamic_tag: Search Engine Optimization for the title tag, the canonical tag and the meta tags author, description and keywords. Configuration by a user interface.
http://typo3.org/extensions/repository/view/seo_dynamic_tag/
-  TSconfig Pages and Users by extManager (extkey: tsconf): Configure the the eight most commonly used TSconfig porperties with the mouse - like page tree uids, activated extended view, activated clipboard, ...
<http://typo3.org/extensions/repository/view/tsconf/>

Change Log

6.3.1 Improvement

- * #i0103: If you like a simple Caddy, please include the static template:
"Caddy [3] + Bootstrap without Caddy options and costs"
This is working in context with Bootstrap only

6.3.0 Feature

- * #i0101: JavaScript for responsive powermail can select by Constant Editor:
See [CADDY - POWERMAIL – JAVASCRIPT]
- * #i0099: Items are saved (TCA inline)
- * #i0093: Bootstrap
Effected files among others:
 - * Configuration/TypoScript/Bootstrap/* (new)
 - * Resources/Public/Html/Bootstrap/* (new)

Improvement

- * #i0102: caddy.js: \$ is moved to jQuery.
- * #i0098: tca.php is moved to Configuration/TCA
 - * proper TYPO3 6.x architecture
- * #i0097: Flexforms are moved
 - * caddy/pi1 -> Configuration/FlexForms/Pi1
- * #i0096: Templates are moved
from Resources/Private/Powermail* to Resources/Private/Templates/Powermail*
- * #i0095: Templates are moved
from Resources/Public/Html to Resources/Private/Templates
- * #i0094: Path to Powermail static template moved. You have to include again once:
 - * Caddy [6] + Powermail 2.x Foundation

Bugfix

- * #i0100: DRS prompt in case of a disabled DRS

6.2.2 Improvement

- * #i0092: Powermail update from 2.17.0 > 2.18.0
Effected files:
 - * Resources/Private/Powermail_2.18.0/Foundation/

6.2.1 Improvement

- * #i0091: Powermail update from 2.1.0 > 2.17.0

6.2.0 Feature

- * #i0090: PDF: Different templates per page
 - * New setup:
 - * plugin.tx_caddy_pi1.pdf
 - * deliveryorder.page
 - * invoice.page
 - * revocation.page
 - * terms.page

Improvement

- * #i0089: Update fpdi from 1.4.2 to 1.5.4

6.1.1 Improvement

- * #i0087: content of folder res is moved to Resources/Public/JavaScript/*
 - * removed: res/javascript/jquery.fancybox-1.3.4/index.html
- * #i0086: content of folder files is moved to Resources/Public/*
- * #i0085: Update the manual

6.1.0 Feature

- * #66534: Mini-Caddy: Foundation Reveal Modal
New static templates
 - * Caddy [3.2] + Foundation Minicaddy Dropdown
 - * Caddy [3.2] + Foundation Minicaddy Reveal
- * #i0084: Delete icon is an icon of the foundation font. Extension start is needed.
- * #i0083: HTML/CSS: optimise for small

6.0.14 Improvement

- * #i0082: HTML/CSS: optimise for medium

6.0.13 Bugfix

- * #i0081: Remove conflict mode from paymill api
- 6.0.12 **Improvement**
 - * #i0080: state is set from alpha to beta
 - * #i0079: absolute path to the PDF document
- 6.0.11 **Improvement**
 - * #i0078: PHP Warning: Invalid argument supplied for foreach() in lib/class.tx_caddy_dynamicmarkers.php
- 6.0.10 **Improvement**
 - * #65458: Shipping Option Pick Up
- 6.0.9 **Manual**
 - * #i0076: New section calculation
 - * #i0075: Cash discount is documented
- 6.0.8 **New Feature**
 - * #i0074: VAT Reg.No. (USt-IdNr.)**Improvement**
 - * #i0073/#156: Accordion with caddy items must be open while the confirmation the order.
It is a legal need in Germany.
 - * #i0072/#156: Powermail confirmation form got a second button line at the top
 - * #i0071/#161: Labelling: formButton is moved from "Buy now" to "Into the caddy"
- 6.0.7 **Feature**
 - * #i0069: Different caddy icons are possible depending on an empty or full caddy
 - * #i0068: Foundation classes for caddy mini
 - * #i0067: Classes for powermail by TypoScript**Improvement**
 - * #i0070: moved: files/img to Resources/Public/Images
- 6.0.4 **Bugfix**
 - * #i0066: Missing argument 2 for tx_caddy_session::zz_escapeStrForLike()
called in lib/class.tx_caddy_session.php on line 2491 and others
- 6.0.3 **Improvement**
 - * #i0065: sumsumgrossepayment: amount for e-payment
 - * #i0064: Paymill response code 50000
 - * #i0063: some foundation files are integrated
 - * #i0061: DRS**Bugfix**
 - * #i0062: Paymill transaction (conflict with #i0058)
- 6.0.2 **Improvement**
 - * #i0060: Update manual: Template structure without Foundation
- 6.0.1 **Improvement**
 - * #i0060: Update manual
- 6.0.0 **Major Feature**
 - * #i0057: Paymill
 - * Upgrade for Foundation 5.x
 - * New test data since Aug. 2014
 - * #62044: TypoScript file structure
 - * #62043: Button template
 - * New constant plugin.caddy.url.extension
 - * Context with plugin.caddy.getpost.uid !
 - See:
 - * plugin.tx_caddy_pi1.templates.html.form.order.foundation.5x
 - * #61879: Foundation 5.x for Powermail 2.x
 - * #61865: Foundation 5.x
 - * #61624: Upgrade TYPO3 6.2
 - * #61631: Replace the static method t3lib_div::debug
 - * #61632: Replace the call to mysql_escape_string
 - * #61633: Remove constant path PATH_t3lib
 - * #61634: Remove require_once (piBase, PATH_t3lib)**Workflow**
 - * #61877: Min and Max failed in case of variants
 - * Not solved but workflow:
 - * Min and Max won't checked in case of variants.

Cause is another param in the url.
See more details at [ChangeLog](#)

Bugfix

- * #i0059: A second item is added without any quantity
- * #i0058: Paymill error in case of a non-e-payment method
- * #i0056: Moved code in contest with #61892
- * #61893: PDF attachments without any address data
- * #61892: Database record without value for customer e-mail
 - * powermail 2.0 to 2.1 moved the POST params from ids to labels.
- * #61889: Variants: Bug in the SQL query for powermail
- * #61870: Delete an item fails in case of variants
- * #58645: Empty powermail mapping causes fatal exception
- * #47370: t3lib_div::getIndpEnv('TYPO3_DOCUMENT_ROOT')

4.0.12 Bugfix

- * #58646: Currency symbol in email is crippled

4.0.11 Feature

- * #i0055: Tax info in the simple caddy HTML

4.0.10 Feature

- * #i0054: Bank account in case of cash in advance for email and pdf
- Bugfix
- * #i0053: Fatal error: Class 'Paymill\Request' not found in lib/e-payment/paymill/class.tx_caddy_paymill_transaction.php on line 765
 - * #i0052: Paymill exception while transaction: '50647.056' must contain only digits

4.0.9 Bugfix

- * #i0051: condition is placed outside the label
- * #i0050: Non e-payment-methods are enable only, if an e-payment-method is enabled before.

4.0.8 Feature

- * #i0049: Tax is removed from HTML template
- * #i0048: Billing-, service- and shipping costs are displayed only, if sum is greater than 0.00
 - * Rule is set for HTML template only
 - * The seller and the client need the information for shipping, payment and specials in the e-mail template in every case.
 - * new fields sumcashdiscountsumnet and sumcashdiscountsumgross among others
- * #i0047: cash discount
 - * new fields: sumcashdiscountnet, sumcashdiscountgross
- * #i0046: new payment option cash on pick up (german: Bar bei Abholung)
- * #i0045: e-payment
 - * enhance payment options from 3 to 6

4.0.7 Feature

- * #55726: Stock management: decrease item quantity after ordering
 - * New db.fields: stockquantity and stockmanagement
 - * New api.getpost: stockquantity and stockmanagement
 - * New api.marker.item: stockquantity and stockmanagement

4.0.6 Bugfix

- * #i0044: Adding a product removes the session
- * #i0043: e-payment
 - * payment id is removed after transaction
 - * payment options are missing in e-mail and pdf

4.0.5 Major Feature

- * #53678: e-payment step 2
- * #53742: Fancybox is removed

Feature

- * #55333: Configurable options
- * #54999: Link to an item (to the single view)

Improvement

- * #55322: Clean up TypoScript structure
 - * plugin.tx_caddy_pi1
 - * conditions -> moved to: api.conditions
 - * getpost -> moved to: api.getpost
 - * marker -> moved to: api.marker
 - * options -> moved to: api.options

- * symbols -> moved to: api.symbols
- * tax -> moved to: api.tax

Bugfixing

- * #i0047: Unproper form in the extension manager (TYPO3 6.x)

4.0.4 Features

- * #54968: Powermail improvement
- * #54967: Responsive framework foundation
- * #54858: Split the form in different parts
- * #54832: Another label wrap for radiobuttons and checkboxes
- * #i0047: Foundation
- * #i0046: Validator

4.0.3 Features

- * #i0042: ###UID_POWERMAIL_FORM###
- * #i0040: jQuery for powermail
- * #i0039: Sum for shipping and specials (without payment)
- * #54628: Calculated Caddy available for dynamicMarkers
- * #54634: Enable controlling of the caddy (pi1) by pid

Improvement

- * #i0038: Caddy HTML

Bugfix

- * #i0045: unproper replace of an empty pid in session -> getPid()
- * #i0044: Unsupported operand types in lib/caddy/class.tx_caddy.php on line 898
- * #i0043: getPaymentOptionLabelBySessionId seem's to be unproper
- * #i0041: ###TARGET### wasn't handled

4.0.2 Major Feature

- * #53678: e-payment step 1

Features

- * #i0037: Replace marker by cObjData
- * #i0036: Debugging: Don't replace empty marker

Improvement

- * #i0034: Disable jquery.caddy-3.0.0 / fancybox
- * #i0033: Move version from 4.0.1 to 3.0.1

Bugfix

- * #i0039: Caddy included TypoScript template for Caddy 4.x. Proper is template for Caddy 2.x

4.0.1 Major Feature

- * #53742: Fancybox

Improvement

- * #i0032: plugin.tx_powermail.settings.misc.showOnlyFilledValues = true

4.0.0 Major Feature

- * #53679: Workflow in AJAX

3.0.0 Major Feature

- * #53360: Fit it into TYPO3 6.x

2.1.5 Improvement

- * #i0032: Searchfields tca.php
- * #i0031: Update the manual. Here: wt_cart

2.1.4 Improvement

- * #52313: Try to externalise tcpdf

2.1.3 Manual

2.1.2 Improvement

- * #i0029: Setup: minimum groos sum for items
- * #i0028: CSS
- * #i0027: pdf filename %H%M -> %H%M%S
- * #52165: Update tcpdf: 5.9.149 -> 6.0.031

2.1.1 Improvement

- * #i0026: additional template caddy_wo_options.html
- * #i0025: DRS

2.1.0 Features

- * #51916: Plugin for amount only

- 2.0.11 **Improvement**
 - * #i0024: db.sql**Bugfixing**
 - * #i0023: \$this->extKey -> \$_EXTKEY
- 2.0.10 **Bugfixing**
 - * #50045: rounding error
- 2.0.9 **Bugfixing**
 - * #i0019: UTF8-Bug in Powermail 2
- 2.0.8 **Features**
 - * #i0022: Unique file names
 - * #i0021: Term of Credit**Improvement**
 - * #i0020: Constant Editor Labelling**Bugfixing**
 - * #49431: misspelling nachnahme
 - * #49427: misspelling constants
 - * #49428: Localisation dependency (isn't tested in not localised context)
 - * #49430: Check TypoScript Case
- 2.0.7 **Features**
 - * #i0018: * New user functions
 - * calcDueDate
 - * calcMultiply
 - * #i0016: * Improved caddy table structur**Improvement**
 - * #i0017: * Shipping tax
 - * Moved from normal to reduced
- 2.0.6 **Improvement**
 - * #i0015: * Overwrite existing files!
- 2.0.5 **Bugfix**
 - * #i0014: * Unproper TypoScript path to HTML table properties for PDF files
 - * #i0013: * If customer doesn't get any invoice and any deliveryorder, HTML template for caddy isn't loaded.
This causes a bug, while sending the confirmation e-mail.
Workflow:
 - * Respect HTML template in case of all PDF files.
- 2.0.4 **Bugfix**
 - * #i0012: * foreach() : Invalid argument in line 607
 - * foreach() : Invalid argument in line 171
- 2.0.3 **Improvements**
 - * #i0006: HTML mini caddy
 - * #i0005: TypoScript is cleaned up
- 2.0.2 **Improvement**
 - * #i0004: Rendering of option lists
 - * #i0003: Calculation of options costs
- 2.0.1 **Feature**
 - * #i0002: Power of Revocation**Bugfix**
 - * #i0001: unproper locallang.xml value in userfunc
- 2.0.0 **Features**
 - * #45954: Database
 - * #45915: Powermail Controller
 - * #45863: Icons
 - * #45808: PDF attachments: Bill and Shipping
 - * #45967: Attachment Invoice
 - * #45968: Attachment Delivery Order
 - * #45969: Attachment Terms & Conditions**FORK**
 - * #45792: DRS - Development Reporting System
 - * #45775: Fork from wt_cart 1.4.6
 - * #45797: Consolidating code

- * #45783: Proper Flexforms
- * #45781: Proper extension manager
- * #45776: Move all labels from wt_cart to caddy

Illustration Index

Illustration 1: Caddy on a desktop and on a smartphone	3	Illustration 29: Plugin Caddy Mini (frontend)	24
Illustration 2: Caddy with e-payment interface	3	Illustration 30: Plugin Caddy Mini (backend)	24
Illustration 3: Confirmation e-mail	4	Illustration 31: Sample with Quick Shop	25
Illustration 4: The caddy e-mail attachments (here: Quick Shop) ..	5	Illustration 32: Plugin [Check it!]	26
Illustration 5: The caddy plugin.	6	Illustration 33: Plugin [Notes]	26
Illustration 6: A caddy order record	7	Illustration 34: Plugin [Caddy]	27
Illustration 7: The Quick Shop Installer enables you, to install Caddy ready-to-use (2+3) with one mouse click (1).	7	Illustration 35: Plugin [Email]	28
Illustration 8: Quick Shop Installer - Installation with one mouse click	11	Illustration 36: Plugin [Billing Address]	29
Illustration 9: Caddy on a desktop and on a smartphone	13	Illustration 37: Plugin [Delivery Address]	30
Illustration 10: Flush System Cache	14	Illustration 38: Plugin [Delivery Address]	30
Illustration 11: Caddy e-payment (Paymill) in live mode	16	Illustration 39: Plugin [Sponsors]	31
Illustration 12: Caddy e-payment (Paymill) in test mode	16	Illustration 40: Plugin [Help!]	31
Illustration 13: Transactions (Paymill test mode)	17	Illustration 41: Caddy blue	50
Illustration 14: Transaction (Paymill test mode)	17	Illustration 42: Caddy green	50
Illustration 15: Caddy with accordion items	20	Illustration 43: Caddy orange	50
Illustration 16: Caddy with accordion shipping and service	20	Illustration 44: Caddy red	50
Illustration 17: Caddy with accordion payment	20	Illustration 45: Caddy blue	51
Illustration 18: Caddy with accordion amount	20	Illustration 46: Icon path in the Constant Editor	51
Illustration 19: Caddy thanks	20	Illustration 47: caddy blue/red	51
Illustration 20: Plugin Caddy (backend)	21	Illustration 48: Caddy icons	52
Illustration 21: Desktop: the icon, if caddy contains items.	22	Illustration 49: Your (1) draft will merged	53
Illustration 22: Click on the mini caddy icon opens a dialog box. .	22	Illustration 50: ... with (2) Caddy content	53
Illustration 23: The position of the icon is fixed.	22	Illustration 51: ... to the (3) attachment.	53
Illustration 24: Smartphone: Caddy without mini caddy	22	Illustration 52: [PDF] tab of the Caddy plugin	54
Illustration 25: The icon, if caddy contains items.	22	Illustration 53: Properties of a PDF attachment (here: invoice) ...	55
Illustration 26: The position of the icon is fixed	22	Illustration 54: Audit proof by the Extension Manager	65
Illustration 27: The dialog box takes the full screen.	22	Illustration 55: Enable the DRS by the Extension Manager	86
Illustration 28: Interface of the mini caddy	23	Illustration 56: Enable the DRS by the Plugin / Flexform	87
		Illustration 57: Enable the DRS by the Plugin / Flexform	87
		Illustration 58: The DRS Report	88

Alphabetical Index

Attachments	Individual sum	48	Live mode	16	Missing	95
Corporate design	Show all disabled	44	Paymill	18	Attachments	53
Audit Proof	Colour Schemes		PCI-DSS	18	Background	55
Database	Attachments	50	Powermail	36	Caddy	40, 56
Disabling	E-mail	50	Provider	35	Colour	42
Bank account	HTML	50	Responsive	17	Date	40, 56
BIC	Icons	51	Test mode	16	Delivery	
Constant Editor	PDF	50	TypoScript	17	Address	56
IBAN	Constant Editor		Error		Font colour	55
Billing	Bank account	34	E-mail	95	Font family	40, 55
Address	Button	37	PDF without address	95	Font stretching	55
Caddy	Colours	37	TCPDF	95	Invoice	41
Invoice address	Currency symbol	34	Evaluation	26	Address	56
Powermail	Database	35	Extension Manager		Numbers	56
Bootstrap	Debugging	35	Debugging	86	Order number	41
Responsive Design	Decimal	34	Form		Paths	54
Static Templates	Due date	34	Button		Power of revocation	41
Button	E-PAYMENT	35	With select box	76	Power of Revocation	57
Templates	GET-/POST-Parameter	36	Without select box	76	Term of credit	57
Caddy	Icons	37	Templates	76	Terms & Conditions	57
Address	Service attribute	35	Foundation		TypoScript	75
Delivery	ShowUid	43	CSS	32	Plugin	
Invoice	Storage folder	34	Powermail	36	Address	
Audit Proof	Table width	38	Responsive Design	13	Delivery	30
Calculating	Tax	42	Static Templates	32	Invoice	29
Check-it!	Templates	43	Get-/Post-parameter	68	Caddy	27
Condition	Thousands separator	34	Installation		Caddy: Mini	23
Database	User functions	35	Manual	12	Caddy: sum gross	24
Debug	Corporate Design		One-Click	10	Debugging	87
Delivery	Button	37	Organiser Installer1	11	E-mail	28
Address	Colour schemes	50	Quick Shop Installer2	11	PDF	30
Order number	Colours	37	Invoice		Powermail	
Discount	Icons	37, 51	Address	56	Address	
Evaluation	Ready-to-use	52	Attachments	56	Billing	29
Invoice	Upload	53	Billing address	29	Delivery	30
Billing address	CSS		Caddy	29	Invoice	29
Invoice number	Foundation	32	Plugin	29	E-Payment	36
Maximum order quantity ...	Powermail	32	Powermail	29	Evaluation	26
Minimum order quantity	Static Templates	32	Templates	43	Foundation	36
Numbers	Database	65, 74	Localization		Static Templates	33
Reset	Debugging		Amount	32	TypoScript	78
Order number	DRS	86	Template	32	Update	89
Pid	E-Payment	18	Marker	69	Work-flow	58
Plugin	Extension Manager	86	Item	69	Responsive Design	13
Sql	GET-/POST-Parameter	35	Sum	70	Bootstrap	13
Static Templates	Paymill	18	Mini Caddy	22	Foundation	13
Sys_registry	Plugin	87	Plugin	23	Screenshots	13
Templates	Report	88	Screenshots	22	TYPO3 Foundation	14
Variants	Session data	86	TypoScript	23	TypoScript	14
Work-Flow	User functions	35, 87	Numbers		Session data	
Caddy Mini	Delivery		Reset	66	Debugging	86
Content	Address	56	Sys_registry	66	ShowUid	43
Item	Caddy	30	Pages		Sponsoring	93
Sum	Plugin	30	Caddy	39	Package	93
Templates	Powermail	30	Caddymini	39	Static Templates	
_HTMLMARKER	Templates	43	Revocation	39	Bootstrap	32
Caddy Sum Gross	Discount		Shop	39	Caddy	32
Content	TypoScript	72	Terms	39	CSS	32
Plugin	DRS		Paymill		Foundation	32
Sum	Debugging	86	Configuration	18	Ordering	32
Templates	E-mail		Constant Editor	35	Powermail	33
Calculating	Delivery Order	28	Debugging	18	Symbols	
Available from	Error	95	Javascript Bridge	18	Quantity	42
Available until	Invoice	28	Keys	35	Service Attribute	42
Based on each	Plugin	28	Live-mode	35	TCPDF	
Based on price	Power of Revocation	28	PCI-DSS	18	Error	95
Based on quantity	Templates	43	Transaction	17	Templates	
Default code	Terms & Conditions	28	TypoScript	33	Button	76
Discount	E-Payment	16	PDF	75	With select field	76
Free from	Constant Editor	35	Address		Without select field	76
Free until	Currency	35	Delivery	56	Caddy	43
Individual max	Debugging	18	Invoice	29, 56	Caddymini	43

Caddysum	43	Payment	71p.	Caddysum	76	Wt_cart	89
Delivery order	43	Paymill	33	Deliveryorder	76	User functions	
E-mail	43	Powermail	78	E-mail	76	Debugging	87
Form	76	Rates	70	Html	76	Variants	
Invoice	43	Reset	33	Invoice	76	Sql	74
Paths	43	Revocation	75	Revocation	76	Work-flow	
TypoScript		Shipping	71	Terms	76	Caddy	58
Decimal	73	Specials	71	Terms	75	Powermail	58
Deliveryorder	75	Sql	74	Values	70	Wt_cart	
E-Payment	33	Symbols	73	Variants	74	Update	89
Euro	73	Tax	73	_CSS_DEFAULT_STYLE	77	Versus Caddy	97
Invoice	75	Templates		_HTMLMARKER	77	_CSS_DEFAULT_STYLE	77
Labels	70	Caddy	76	Update	89	_HTMLMARKER	77
Options	71p.	Caddymini	76	Powermail	89	Caddy Mini	84