

5. Further Configuration

- [Extend menu and maintain contact reasons](#)
- [Activate Order template for eCom](#)
- [Maintaining Client Values for eCom](#)
- [Maintain Settings](#)

Extend menu and maintain contact reasons

Summary

Two small configurations that concern the footer and the contact form of the shop.

Create custom menu items

Via *eCom \ Menu extension* you create additional menu items for the shop. They appear in the **second column of the footer — separately for each language**.

1. Open *eCom \ Menu extension*.
2. Select language.
3. Create and fill menu item.
4. Save and provide data.

Typical examples from a running system are menu items like *eCom user manual* or *Newsletter*.

Maintain contact reasons

The selection list in the shop's contact form comes from a code table.

1. Open *Basic configuration \ Code table*.
2. Select type **eCom contact reason**.
3. Maintain the desired contact reasons.
4. Save and provide data.

Without maintained contact reasons, the customer has no selection in the contact form.

To ensure that enquiries reach you, the e-mail subscription **eCom - contact form** must also be set up.

Activate Order template for eCom

Summary

Order templates only appear in the shop if you activate them there. This is done under *Order support \ Order template* in the **eCom** section of the respective template.

The fields of this section and their meaning are described in the service description.

Procedure

1. Open *Order support \ Order template* and select the template.
2. In the **eCom** section, set the **Display**:

Value	Effect
None	does not appear in the shop (default)
Start menu	appears in the product groups menu
Highlight menu	appears in the Start menu and in the Highlight menu

3. If necessary, set **Valid from** and **Valid until**. This controls the shop period independently of the client.
4. If necessary, set **Prefix** and **Name** for the display in the shop.
5. Save and provide data.

Why separate appointments for the shop make sense

The field force usually sees promotions a few days before the start to prepare and plan appointments. For the shop, you usually do not want this — the promotion should only appear on

the start day.

This is exactly what **Valid from** and **Valid until** in the eCom section are for: they override the period of the client.

Why a separate name makes sense

If the order template has a name intended for the field force — for example with internal notes or condition information — you can override it customer-optimised for the shop via the **Name** field.

Only for the shop, not for the field force

The checkbox **Not for field force** makes the template available exclusively in the eCom system.

Limit assortment to an item list

The checkbox *For the selected customers of this customer list, display only the items of the item list in the eCom system* acts like a closed listing: the customers of this template may only access, search for, and order the items of the item list.

“ **Warning:** If a customer appears in **multiple** order templates with this checkbox, the eCom system makes a **random selection** of which one applies. A mixture of several item lists is not possible. Ensure that each customer appears in only one such template.

What does not arrive in the shop

No modules are available in the eCom system. Modules assigned to the order template are not displayed in the shop — regardless of which they are.

Maintaining Client Values for eCom

Summary

Some values that control the order completion in the shop are not located in eCom, but under *Basic configuration \ Clients*.

The basic rule: **Client values apply to all channels**, that is, for both Client and Shop together — unless eCom deviates via a subsequent setting.

Two value lists are the exception and carry their own **eCom flag**.

The two value lists with eCom flag

Affected are **Free carriage limit** and **Freight costs**. Both lists have the columns `Value`, `Currency` and `eCom`.

“ If active, the value applies only to eCom. If not active, only to the DeDeSales Client.

“ **Attention:** The flag is a **switch, not an addition**. A row applies either to the Shop or to the Client, never to both. Anyone who wants to supply both channels needs **two rows** — one with the flag and one without. If the eCom row is missing, the Shop has no value.

Both lists include a currency. With multiple currencies, you therefore need one row per currency — and that again per channel.

Procedure

1. Open *Basic configuration \ Clients* and select the client.
2. Go to the value list **Free carriage limit** or **Freight costs**.
3. Create a row for the Client — **without** the eCom flag.
4. Create a row for the Shop — **with** the eCom flag.
5. Repeat for each currency if there are multiple currencies.
6. Save and provide data.

Interaction of Free carriage limit and Freight costs

Below the Free carriage limit, freight costs are calculated; above it, they are not.

The calculated freight costs appear in the Cart as a separate line — above the item you have stored under *eCom \ Settings* as the **Freight costs Item**. This setting therefore contains the **Item number**, not the amount.

Values without eCom flag

These fields of the client apply to all channels:

Field	Known effect
Minimum order value	Does not apply in the Shop — this general value is not read during order completion, but only used for export. For the Shop, the value in the eCom tab applies. See below.
Minimum delivery period in days	Controls the earliest possible date of delivery.
Maximum item value	Is calculated, but the message does not reach the customer — it is an enquiry, and the Shop skips enquiries. See below.
Warning at quantity	Similarly: is calculated, but the enquiry does not reach the customer. See below.
Holiday ID	Applies in the Shop . Determines the public holiday calendar for calculating the earliest possible date of delivery — one day is added for each public holiday encountered.

Field	Known effect
Minimum order value (warning)	Does not apply in the Shop — the value is not filled anywhere and would remain invisible as an enquiry anyway.
Minimum order quantity	Does not apply in the Shop — only in the sales client. See below.
Minimum order unit	The unit to which the minimum order quantity refers — for example <i>pcs</i> or <i>crt</i> . Not a numeric value. Also applies only in the Client.

The minimum order value appears twice

There are *two* minimum order values at the client: one general and one in the **eCom tab**. For the Shop, only the value in the eCom tab counts — the general one is not read during order completion.

Value	Effect in Shop
Minimum order value, eCom tab	applies
Minimum order value, general	none — only for export
Minimum order value of the debtor	only in a special case, see Order completion
Minimum order value (warning)	none

“ **Attention:** This is the most common confusion on this page. Anyone who maintains the value limit for the Shop in the general field wonders why nothing happens — and finds the error hard to detect because the value is correctly saved.

Why two fields are calculated but nothing appears

Maximum item value and *Warning at quantity* are actually evaluated server-side. However, their message is an **enquiry** — a message that someone in the sales client confirms to proceed.

In the Shop, there is no one who could do this. Therefore, the Shop **always skips enquiries**. This is a deliberate decision, not an error.

Field	Sales client	old Shop	new Shop
-------	--------------	----------	----------

Maximum item value	applies	message was visible	invisible
Warning at quantity	applies	message was visible	invisible

The fields are therefore **not non-functional** — they just no longer have effect where the customer would see them. Anyone maintaining them for the Shop should know that they remain without consequences there.

Minimum order quantity: no effect in the Shop

The two fields belong together — a check only takes place if **both** are filled — and they can be combined with the debtor value. However, none of this reaches the Shop:

- When loading the Cart, the undercutting is **calculated** and a notice text is generated (*"The minimum order quantity has not been reached ..."*).
- The interface through which the Cart retrieves its notices does **not output** this text. It is therefore lost before it reaches the Shop.
- The order completion only checks the minimum order **value** in euros, not quantity. Therefore, an order is never blocked by this.

This is not a peculiarity of the new Shop — the gap already exists in the existing web interface and thus applied equally to the old Shop.

For the setup, this means: The two fields may be maintained, but they then only apply in the order entry of the sales client.

Maintain Settings

Summary

Under *eCom \ Settings* you configure the behaviour of the shop.

Which settings exist, what values they take, and what they do is described in the specification on the page *Settings at a Glance*. This page only describes the procedure and the places where mistakes are commonly made.

Procedure

1. Open *eCom \ Settings*.
2. Change setting.
3. Save.
4. Provide data.

A change takes effect at the earliest after the next shop retrieval interval, i.e. after 15 or 30 minutes.

Settings that only take effect together with another place

This is the most common source of errors: the setting is set, but the second place is missing.

Setting	Also requires
Voucher input	a created voucher under <i>eCom \ Voucher management</i>
Freight costs item	Freight costs and Free carriage limit as client values with eCom flag
Apply minimum order value from debtor	delivered debtor data; otherwise the client value applies
Default language	Item texts and product group texts in the selected culture

Setting	Also requires
Not a customer yet?, checkbox removed	the text block <i>Access request – General Information</i> under <i>eCom Texts</i>

Change shop colours via CSS

The colouring is stored as free text in the field **CSS**. The colours are given as CSS variables in HSL notation.

For switching to your corporate colour, usually two values are sufficient:

1. `--primary` set to your corporate colour — dark shade.
2. `--primary-soft` set to the same hue angle, but very light.

The HSL notation is `hsl(hue saturation lightness)`. To make it lighter or darker in the same colour, you only change the **last** value. Variables with `var(--primary)` automatically take the value of the main colour.

“ **Warning:** Each colour has a `-foreground` colour, i.e. the text colour on it. Ensure sufficient contrast, otherwise text in the shop becomes unreadable.

The page background

Previously, the shop generated a **background gradient** from `--primary`. This could only be disabled by manually setting its two endpoints `--verlauf-warm` and `--verlauf-kalt` to the same colour.

There is now a dedicated variable. Both endpoints point to `--shop-hintergrund`, which in turn points to `--primary-soft`:

```
--shop-hintergrund: var(--primary-soft); --verlauf-warm: var(--shop-hintergrund); --verlauf-kalt: var(--shop-hintergrund);
```

Because both endpoints have the **same** value, in the default state **no gradient is visible** — the background is solid. And because the value is linked to `--primary-soft`, it automatically adapts to the corporate colour: Whoever changes `--primary-soft` gets the matching background.

Set a different background. Now only **one** line is needed instead of two:

```
:root { --shop-hintergrund: hsl(0 0% 100%); }
```

Want a gradient. Set `--verlauf-warm` and `--verlauf-kalt` individually to two **different** colours and leave `--shop-hintergrund` untouched.

“ **Note on colour choice:** A strong main colour appears much stronger over large areas than on a button. Because the background is linked to `--primary-soft` — the **light** shade — this is less noticeable than before. Still, check the result on the finished shop before fixing the colour.

The default state and a counterexample

The shop is delivered in **orange**. Next to it is a blue palette as an example for a change — both are based on **one** hue angle:

Variable	Orange (default)	Blue (example)
<code>--primary</code>	<code>hsl(27 87% 49%)</code>	<code>hsl(215 100% 25%)</code>
<code>--primary-soft</code>	<code>hsl(27 87% 94%)</code>	<code>hsl(215 100% 94%)</code>
<code>--secondary</code>	<code>hsl(36 82% 50%)</code>	<code>hsl(211 100% 40%)</code>
<code>--accent</code>	like <code>--primary</code>	like <code>--primary</code>
<code>--ring</code>	like <code>--primary</code>	like <code>--primary</code>
<code>--shop-hintergrund</code>	<code>var(--primary-soft)</code>	freely selectable
<code>--verlauf-warm</code> , <code>--verlauf-kalt</code>	both <code>var(--shop-hintergrund)</code>	both <code>hsl(0 0% 100%)</code> — white, so no visible gradient

From `--primary` to `--primary-soft` the hue and saturation remain **the same**; only the **lightness** jumps — for orange from 49 to 94 per cent, for blue from 25 to 94. This is exactly how the notation is intended: one colour, two lightness levels.

`--secondary` differs slightly in both cases — a few degrees in hue and a lighter level. This is enough to set off secondary elements without introducing a second colour.

“ **Note:** `--accent` and `--ring` take the main colour. If you do not explicitly want them different, you do not have to set them. In practice, there are thus **three** values: `--primary`, `--primary-soft` and `--secondary`.

The complete list of variables is in the specification.