

Cookbook - Forms Management

- Overview
 - Registration
 - Registration attributes cheat sheet
 - B2E registration SaaS
 - Manager assisted registration SaaS 3.7.0+
 - Profile
 - Guest Checkout
 - Email 3.5.0+
 - Address
 - Address attributes check sheet
 - Value mapping
 - Declaring address form attribute
 - Declaring address formatting

Overview

Any shop has a number of different types of form that customers can interact with. One of the basic examples are registration form - every commerce shop has it and it is different for almost every shop. Reason behind this is that each business wishes to collect different types of information.

In order to approach this problem in a dynamic way the platform offers a number of tools for configuring typical forms.

Registration

Each shop can define several supported customer types, which essentially means the types of customers that can register on the site. This is defined by shop attribute **SHOP_CUSTOMER_TYPES** which accepts a localised free form list of values. Typical definitions could be B2C, B2B, B2E but it is at business user discretion what customer types to create. Note that there is a special type **B2G** which is reserved for guest checkout customer type.

Typical configuration may look like this:

Attribute

Supported shop customer types CSV
E.g. value=B2B,B2C, display value=Private,Company

Customer: supported customer types

default value

de

en

↗
✖
✖
+

For each of these types it is now possible to define list of **CUSTOMER** group attributes that will represent this form. This list is set in shop attributes **SHOP_CREGATTRS_XXX** where XXX is the customer types defined above.

For example:

Attributes/Images Payment Gateways B2B Address book

CREGATTRS ✖ ?

🔒 ✎ ✖ ↺ ↻ 💾

Attribute	Type	Value
Customer (B2B): registration form attributes (CSV)	CommaSeparatedList	salutation,firstname,lastname,FUNCTION,DEPARTMENT,CUSTOMER_PHONE,CUSTOMER_MOBILE,CUSTOMER_FAX,COMPANY,TAX,MARKETING_OPT_IN
Customer (B2C): registration form attributes (CSV)	CommaSeparatedList	salutation,firstname,lastname,CUSTOMER_PHONE,MARKETING_OPT_IN
Customer (B2E): registration form attributes (CSV)	CommaSeparatedList	
Customer (B2G): registration form attributes (CSV)	CommaSeparatedList	salutation,firstname,lastname

From the screenshot it can be observed that for each type of customer there is a different list of attributes.

E.g. B2B type has the following fields:

salutation,firstname,lastname,FUNCTION,DEPARTMENT,CUSTOMER_PHONE,CUSTOMER_MOBILE,CUSTOMER_FAX,COMPANY,TAX,MARKETING_OPT_IN

These attributes can be viewed in the admin application System > Attributes > CUSTOMER group

Attributes / Customer settings.

← CUSTOMER x ?
⌵
📄
+
✎
✖
↺
↻
📄

Code	Name	Type	Validation	Rank
CUSTOMER_FAX	Fax	Phone	✖	500
MARKETING_OPT_IN	Marketing Opt in	Boolean	✖	500
CUSTOMER_MOBILE	Mobile	Phone	✖	500
CUSTOMER_PHONE	Phone	Phone	✓	500

Therefore each property can have a full definition of the labels, validation properties and type of values that allows to automatically create registration forms specific to customer type in each shop. The fields are ordered in the same sequence as defined in the **SHOP_CREGATTRS_XX** attribute, so sorting field as simple as rearranging the list. Changing the type of the attribute definition will result in specific form field type. For example Boolean value will be displayed as checkbox, CommaSeparatedList - select box etc

There are special system values that are associated with each customer account such as **email** and **password**. Prior to version 3.5.0 these are build into templates. However in versions 3.5.0+ there are no more fixed fields and everything is configurable. To identify these special attributes the attribute definition's value has to be set with: **email**, **password** or **confirmPassword**.

This gives flexibility to have multiple configurations. E.g. create one attribute with [CODE=email, VAL=email] with one regex and then [CODE=another_email, VAL=email] with different regex. Thus you can define different set of the email attributes per shop or even per customer type.

Adding password and confirmPassword attributes with enable customers to specify password of their choosing during registration (see [password managemen](#) cookbook for more details).

Adding email attribute is mandatory for versions 3.5.0+ for registration form to work.

After set up the shop registration form will look something like this:

1 Log In
2 Your Address
3 Shipping Method
4 Payment

Login

E-mail

Password

[Sign In](#) [Forgot password?](#)

Guest checkout

E-mail

First name

Last name

[Checkout](#)

New customer

B2C
 B2B

E-mail

Salutation

First name

Middle name

Last name

Phone

Marketing Opt in

[Register](#)

Registration attributes cheat sheet

Below are mandatory/special attributes for registration forms that are defined in the **CUSTOMER** attribute group

Attribute	Value	Mandatory	Since	Description
salutation			2.x.x	Salutation is an optional attribute that "proxies" salutation property on the Customer object
firstname			2.x.x	First name is mandatory attribute that "proxies" firstname property on the Customer object v Attempt to create Customer object without it results in SQL error.
middlename			2.x.x	Middle name (second name) is an optional attribute that "proxies" middlename property on
lastname			2.x.x	Last name is mandatory attribute that "proxies" lastname property on the Customer object w Attempt to create Customer object without it results in SQL error.
email	email		3.5.0	Email property allows to define email regex. Since shops can have own specific view on wha attribute is driven by value which must be email . Attempt to create Customer object without i
password	password		3.5.0	Marks a password field to be used in the form. The input is captured only inside the system t The attribute is not saved on Customer object
confirmPassword	confirmPassword		3.5.0	Marks a password field to be used in the form. The input is captured only inside the system t The attribute is not saved on Customer object
regAddressForm	regAddressForm		3.5.0	Marker attribute for registration for to include shipping address form for given customer type. such as salutation, firstname, middlename, lastname, companyname1, companyname2 hone1 (CUSTOMER_PHONE) exist on address form there is a fallback mechanism to preve example CREG shop attribute with our address form: email,salutation,firstname,middlename NE,MARKETING_OPT_IN,password,confirmPassword will be replaced to email,regAddress password,confirmPassword if your shipping address form had salutation,firstname,middlene
companyname1			3.5.0	B2B company name "proxies" companyName1 property on the Customer object
companyname2			3.5.0	B2B additional company name (e.g. suffix) "proxies" companyName2 property on the Custor
companydepartment			3.5.0	B2B department within company "proxies" companyDepartment property on the Customer c

B2E registration SaaS

B2E is a feature whereby a shop represents a market place or a collection of subshops where each registered customer could belong a specific sub shop. In such cases the customer account needs to be associated with specific sub shop. For that purpose a special CUSTOMER group attribute **b2bsubshop** is used. It is rendered as a text field which requires customer to enter the name of the sub shop they are registering in.

If this attribute is not present in the list of registration form attributes for that type the customer is registered in master shop

This configuration provides flexible mechanism for categorising customers. For example by departments in large customers or to provide privileged access to products and special prices.

Manager assisted registration SaaS 3.7.0+

When manager logs in into storefront one of the options available is to register a new customer. This is usability improvement feature which allows manager to have a per customer type focused form with all fields required to register that customer including default address with one click.

In order to enable this feature "manager log in" must be enabled for shop and manager has to have **ROLE_SMCALLCENTERREGISTERCUSTOMER** assigned to them.

Customer types available for registration will be comprised of supported types through **SHOP_CUSTOMER_TYPES** as well as all declared types using **SHOP_CREGATTRS_XXX** attributes. This allows to create "private" customer types that are only available through manager assisted registration process.

If "registration requires approval" is enabled for the customer type the customer account will be auto-approved and activated during manager assisted registration. If this is not desired your theme must override the RegisterForm template and set hidden attribute activateAccount=0.

For example consider the following use case: an organisation whereby B2B customers work through a signed contract and need to be verified to be eligible for special prices. In this instance B2C customers can be allowed to register (i.e. **SHOP_CUSTOMER_TYPES=B2C**) but B2B customers can only be registered by manager, thus **SHOP_CREGATTRS_B2B** is specified in shop attributes but not specified in **SHOP_CUSTOMER_TYPES**.

Profile

Similarly to registration form attributes the profile form can be configured as list of **CUSTOMER** group attributes. This configuration is separate to registration form thus allowing flexibility to specify additional properties or hide some which were captured during registration.

Profile forms are controlled by shop attributes of the form **SHOP_CPROFATTRS_VISIBLE_XXX** and **SHOP_CPROFATTRS_READONLY_XXX** or list of customer type specific profile form and list of attributes which cannot be edited on that form respectively. To specify common definition of the profile form **SHOP_CPROFATTRS_VISIBLE** and **SHOP_CPROFATTRS_READONLY** (without the customer type suffixes) can be used.

Your profile

E-mail

E-mail bob.doe@test.com

Additional information

Salutation

First name

Middle name

Last name

Phone

Marketing Opt
in

Save

Guest Checkout

Guest checkout is viewed as a special case of registration form. In fact it can be seen from Registration form overview above that there is **SHOP_CREGATTRS_B2G** is used to define the guest checkout form. Therefore the configuration of the form itself is exactly the same as for normal registration.

One subtle difference is that this type of checkout has to be enabled by setting shop attribute **SHOP_CHECKOUT_ENABLE_GUEST**. When this attribute is set to true the form will automatically appear as an option on the checkout page for non-logged in customers .

Login

E-mail

Password

[Forgot password?](#)

Guest checkout

E-mail

First name

Last name

New customer

B2C
 B2B

E-mail

Salutation

First name

Middle name

Last name

Phone

Marketing Opt in

Email 3.5.0+

Email is a very common control around the shop which can be used in login, registration, contact form, newsletter form and effectively in any form that requires email communication with customer. As such it is sometimes important to provide restrictions on which kind of emails can be used.

Platform versions 3.5.0+ now have capability to specify shop specific email regular expressions to validate all email fields used in shop.

New special customer type is defined as **EMAIL** in the platform thus allowing to create a registration attribute form for **SHOP_CREGATTRS_EMAIL**. This configuration must have a single **CUSTOMER** group attribute definition that lists the email field to use.

This definition provides validation configuration for login, contact forms and newsletter forms across the whole shop.

Address

Address form takes on the same principle discussed for registration and profile, however it deals with attribute definitions defined in **ADDRESS** group.

The linking of attributes is different as all configurations for address forms are done via **ADDRESS** group attributes. The rationale for this is that whereas registration and profile forms are shop specific (and hence are forms are defined in shop attribute), address forms are country specific in most cases and thus can be shared by many shops.

The basic form definition for address form is defined in **default_addressform** attribute definition of the **ADDRESS** group. This is the configuration for the globally shared address form (effectively a master fallback for address form) and looks something like this:

Attributes / Customer address settings. / Customer: "default_" address form (CSV)

As depicted on the screenshot of the Options tab, the default value lists all **ADDRESS** group attribute codes that define this form. Each attribute listed can be found in the same attribute group listing.

Attributes / Customer address settings.

Code	Name	Type	Validation	Rank
default_custom4	Additional Info	String	✘	500
default_addrline1	Address	String	✔	500
default_addrline2	Address 2	String	✘	500
default_city	City	String	✔	500
default_mobile2	Contact phone number	String	✘ ((+){0,1}((00-9- /){5,})	500
default_countrycode	Country	String	✔	500
default_addressform	Customer: "default_" address form (CSV)	CommaSeparatedList	✘	500
default_email1	Email	String	✘	500
default_firstname	First name	String	✔	500
default_lastname	Last name	String	✔	500

Each of these attributes definitions provides the label, validation and type of field configurations in similar fashion as registration and profile forms are configured. One important difference to note is that all field specific attributes must specify field where values are written to in the value of attribute definition. This is similar concept to **email** and **password** fields attributes described previously.

Inspecting for example default_city attribute will show that its value is set to write to the **city** property on the address object.

Attributes / Customer address settings. / City



Main Localization Validation **Options** Search

Default value / Choice options

default value

city



Language

Value



Address attributes check sheet

Value mapping

Table below depicts value options are available for the address object.

Note that values marked mandatory must be present in the form as these are defined as mandatory on Address object.

Another property to point out is the formatter placeholder, which is how the value is specified in the SHOP_ADDRESS_FORMATTER_XXX and it is different to the value (especially note the letter case).

Value	Formatter placeholder	Mandatory	Since	Description
name	name		3.5.0	Address name e.g. "Home", "Work"
city	city	✓	2.x.x	City
postcode	postcode		2.x.x	Post code / Zip code
addrline1	addrline1	✓	2.x.x	Usually street
addrline2	addrline2		2.x.x	Usually house number or apartment details
countryCode	countrycode	✓	2.x.x	Special field which automatically lists countries available for this type of address in a select box
stateCode	statecode		2.x.x	Special field used when states are defined for countries in locations section of admin
phone1	phone1		2.x.x	Phone
phone2	phone2		2.x.x	Another phone
mobile1	mobile1		2.x.x	Mobile
mobile2	mobile2		2.x.x	Another mobile
salutation	salutation		2.x.x	Salutation
firstname	firstname	✓	2.x.x	First name
lastname	lastname	✓	2.x.x	Last name / Surname
middlename	middlename		2.x.x	Middle name
email1	email1		2.x.x	Email (this is not customer email but rather an email address to use with this address e.g. like company email if delivered to warehouse)

email2	email2		2.x.x	Another Email
companyName1	company1		3.5.0	B2B company name
companyName2	company2		3.5.0	B2B additional company name e.g. suffix
companyDepartment	department		3.5.0	B2B department within company
custom0	custom0		2.x.x	Custom field 0
custom1	custom1		2.x.x	Custom field 1
custom2	custom2		2.x.x	Custom field 2
custom3	custom3		2.x.x	Custom field 3
custom4	custom4		2.x.x	Custom field 4
custom5	custom5		2.x.x	Custom field 5
custom6	custom6		2.x.x	Custom field 6
custom7	custom7		2.x.x	Custom field 7
custom8	custom8		2.x.x	Custom field 8
custom9	custom9		2.x.x	Custom field 9

Each address object allows to capture some core fields that have specific names but also has 10 custom fields that can be used for anything. Therefore depending on the attribute definition these fields can have own labels, validation rules etc.

Declaring address form attribute

In order to create forms which are address type specific or shop specific or both we can use prefixes. Consider the following example where we have two shops with shop code SHOP10 and SHOP20, which are ordered in the sequence of rank (i.e. default lookup sequence)

Form definition attribute code	Eligible for
SHOP10__B2B__addressform_S	SHOP10 definition for B2B customer type definition for shipping addresses
SHOP10__B2B__addressform_B	SHOP10 definition for B2B customer type definition for billing type addresses
SHOP10__B2B__addressform	SHOP10 definition for B2B customer type definition for shipping and billing type addresses
SHOP10__default__addressform_S	SHOP10 definition for shipping type addresses
SHOP10__default__addressform_B	SHOP10 definition for shipping type addresses
SHOP10__default__addressform	SHOP10 definition for shipping and billing type addresses
B2B__addressform_S	Global definition for B2B customer type definition for shipping addresses available to SHOP10 and SHOP20
B2B__addressform_B	Global definition for B2B customer type definition for billing type addresses available to SHOP10 and SHOP20
B2B__addressform	Global definition for B2B customer type definition for shipping and billing type addresses available to SHOP10 and SHOP20
default__addressform_S	Global definition for shipping type addresses available to SHOP10 and SHOP20
default__addressform_B	Global definition for shipping type addresses available to SHOP10 and SHOP20
default__addressform	Global definition for shipping and billing type addresses available to SHOP10 and SHOP20

Declaring address formatting

Displaying address information is accomplished by address formatter, which is set via attribute at the shop level. The pattern for the attribute code is **SHOP_ADDRESS_FORMATTER[CLASSIFIER][COUNTRY][LOCALE][CUSTOMERTYPE]**. Therefore the attribute code can be set to represent the granularity of the formatter at very fine level. The attributes can be defined in attribute management and then shop owners can set the format they need at the appropriate level of granularity.

The fallback system of the attribute is shown below:

Shop formatter attribute code	Eligible for
SHOP_ADDRESS_FORMATTER_S_GB_en_B2C	Definition of the format of the address for Shipping address type (S) for GB country in English (en) eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_B_GB_en_B2C	Definition of the format of the address for Billing address type (B) for GB country in English (en) eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_GB_en_B2C	Definition of the format of the address for GB country in English (en) eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_S_GB_B2C	Definition of the format of the address for Shipping address type (S) for GB country in all languages eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_B_GB_B2C	Definition of the format of the address for Billing address type (B) for GB country in all languages eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_GB_B2C	Definition of the format of the address for GB country in all languages eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_S_en_B2C	Definition of the format of the address for Shipping address type (S) for any country in English (en) eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_B_en_B2C	Definition of the format of the address for Billing address type (B) for any country in English (en) eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_en_B2C	Definition of the format of the address for any country in English (en) eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_S_B2C	Definition of the format of the address for Shipping address type (S) for any country in any language eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_B_B2C	Definition of the format of the address for Billing address type (B) for any country in any language eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_B2C	Definition of the format of the address for any country in any language eligible for B2C customer type
SHOP_ADDRESS_FORMATTER_S_GB_en	Definition of the format of the address for Shipping address type (S) for GB country in English (en) eligible for any customer type
SHOP_ADDRESS_FORMATTER_B_GB_en	Definition of the format of the address for Billing address type (B) for GB country in English (en) eligible for any customer type
SHOP_ADDRESS_FORMATTER_GB_en	Definition of the format of the address for GB country in English (en) eligible for any customer type
SHOP_ADDRESS_FORMATTER_S_GB	Definition of the format of the address for Shipping address type (S) for GB country in all languages eligible for any customer type
SHOP_ADDRESS_FORMATTER_B_GB	Definition of the format of the address for Billing address type (B) for GB country in all languages eligible for any customer type
SHOP_ADDRESS_FORMATTER_GB	Definition of the format of the address for GB country in all languages eligible for any customer type
SHOP_ADDRESS_FORMATTER_S_en	Definition of the format of the address for Shipping address type (S) for any country in English (en) eligible for any customer type
SHOP_ADDRESS_FORMATTER_B_en	Definition of the format of the address for Billing address type (B) for any country in English (en) eligible for any customer type
SHOP_ADDRESS_FORMATTER_en	Definition of the format of the address for any country in English (en) eligible for any customer type
SHOP_ADDRESS_FORMATTER_S	Definition of the format of the address for Shipping address type (S) for any country in any language eligible for any customer type

SHOP_ADDRESS_FORMATTER_B	Definition of the format of the address for Billing address type (B) for any country in any language eligible for any customer type
SHOP_ADDRESS_FORMATTER	Definition of the format of the address for any country in any language eligible for any customer type

Default SHOP_ADDRESS_FORMATTER value if not specified anywhere is:

```

{{addrline1}} {{addrline2}} {{postcode}} {{city}} {{countrycode}}
{{statecode}}, {{salutation}} {{firstname}} {{middlename}} {{lastname}},
{{phone1}} {{mobile1}} {{email1}}

```

New lines in the pattern will be converted to
 tags in HTML automatically. You can also add any text in between place holders (see example below)

```

{{salutation}} {{firstname}} {{lastname}}
{{addrline1}} {{addrline2}}
P/O: {{postcode}}
{{city}} {{statecode}}
{{countrycode}}
Tel: {{phone1}}
Mob: {{mobile1}}
Email: {{email1}}

```

Contact SaaS

Configurable contact forms is a feature that allows to generate any type of contact form by creating a form definition and then calling the form by ID.

The configuration process is virtually the same as address forms, however for contact forms a separate attribute group **CONTACTFORM**.

Once the forms are configured they can be rendered on pages using a url include using two parameters **formId** which is the **attribute code of the CONTACTFORM attribute that has the form declaration** and view ("ContactsForm" is default but your theme may have own special template):

```

ctx.includeurl
"/internal/contactform?formId=default_contactform&view=ContactsForm";

```

Because the contact form submission is done via AJAX you can also place any arbitrary form in any part of the CMS content using script include

```

<script
type='yc-include'>/internal/contactform?formId=default_contactform&view=Co
ntactsForm</script>

```