

Attributes

- [Overview](#)
- [Attribute groups](#)
- [Attribute Management](#)
 - [Attribute code](#)
 - [Value type](#)
 - [Validation](#)
 - [Search options](#)

Overview

No matter how careful the system is designed to store data initially eventually there will be a time when extra pieces of information will need to be saved. Data enrichment is part of a natural system life cycle. The platform uses **Object-Attribute-Value** (OAV) model that allows to add unlimited number of additional data properties to extensible objects at runtime without any coding.

Typical example would be: a marketing team wishes to capture new customer preferences for contacting them by telephone. For this a new attribute definition would be created in CUSTOMER group. Then the code of this attribute definition will be added to registration form attributes configuration and profile attributes configuration for a given shop. As a result registration and profile management pages are updated with no coding, no deployment, it is ready to use as soon as the configurations are applied and marketing team can use the new customer preference to make more sales.

The platform identifies a number of extensible objects whose attributes serve various purposes. These objects are identified by attribute group. A rough depiction of how attributes are grouped and used can be seen in diagram below.

Admin App (6:18)

[Play](#)

Basics: Attribute Management



OAV mechanism works especially well when applied to product data modelling. Different product types can be modelled to have various attributes which can be shared or be unique to specific products. The mechanism is very generic and is fully controlled from Admin app providing generic no-coding required approach to business functions such as filtered navigation, search, product specification view and product comparison. For detailed discussion of the topic see [Product information management \(PIM\)](#).

Attribute groups

Attribute groups specify attributes that belong to specific extensible object in the system. This grouping is done for better management of attributes that belong to different functional parts of the system. Out of the box the platform provides the following groups:

| Attribute Group | Purpose | Where to configure |
|-----------------|---|---|
| SYSTEM | extends the platform itself to contain arbitrary parameters | System > Preferences section of the admin app |
| SHOP | extends specific shop instance usually to provide default configurations and adjust behaviour of business functions | Attributes/Images tab of the shop editor |

| | | |
|------------------|--|--|
| CATEGORY | extends master catalog categories and content objects in the system | Attributes/Images tab of the category/content editor |
| CUSTOMER | extends customer profile information to enhance quality of data about the customer | Attributes/Images tab of the customer editor, for registration and profile forms configuration see this cookbook |
| BRAND | extends brand data | Attributes/Images tab of the brand editor |
| PRODUCT | extends product and SKU information to provide attributes that can be used by product type definitions to create product specifications and facilitate filtered navigation | Attributes/Images tab of the product/SKU editor |
| ADDRESS | defines address capture forms | Configured in the attribute definitions see this cookbook |
| CONTACTFORM SaaS | defines contact forms (e.g. contact us, newsletter sign up) | Configured in the attribute definitions see this cookbook |
| WIDGET | define widgets available to select on dashboard, system managed and are used to provide information about the widgets | Configured in the attribute definitions |

Attribute groups API is generic, therefore custom implementation can define own groups for configuring specific settings

Attribute Management

Configuring attributes is a simple task and usually involves creating an attribute definition in **System > Attributes** section. Basic details required are declaration of a unique attribute code and type of the value for this attribute. After an attribute is created its code cannot be changed.

Usually attributes are created for a specific purpose such as additional attributes already supported by the platform, new attributes for custom implementations and meta data. When and how the attribute is created and used will largely depend on the business requirements.

In most cases business users will only need to create PRODUCT attributes to [configure product types](#). For example product type of "Notebook" may need attributes such as "RAM", "HDD specification" and "CPU speed", where as product of type "Dress" would need attributes such as "color", "size" and "collection".

The best way to think about attributes is as a description of properties of an object. For example when defining "color" attribute for PRODUCT we may specify it as a String property because it is just a colour name but that can be translated into various languages, so it has to be localisable. But we always search for specific colour so every value that we save e.g. "red", "white" and so on will be a specific match in terms of search. On the other had if we are talking about "RAM" we may specify this as a number (e.g. number of GB), which would allow us to search for notebooks that have RAM of more than 4GB but less than 8GB and so on. For advanced attributes we can even specify validation expression. For example when configuring "email" attribute for CUSTOMER we can set a regular expression rule to tell the system which emails are valid and thus prevent customers from entering bogus emails. There are many use cases and the platform provides a flexible mechanism to cover this.

Specifying attribute type will also enable some of the features in Admin app. Thus for specific value types a dedicated editor will be used. For example String would use a localisable component to capture the value, Integer will use a single input with validation for whole number values and Image will have an upload button and a preview for existing value.

DEMO ENVIRONMENT

Powered by
YC - pure Commerce

Logged in as Yes

Customer service

PIM & Master Catalog

CMS

Marketing & Pricing

Inventory

Reports

Shipping

Shops

Organisation

System

Attributes

Import

Export

Search Indexes

Payment Gateways

Unprocessed Callbacks

Clone Shop

Preferences

Query API

Cache Monitoring

Performance Sensors

Performance Samples

Tasks

Cluster

Help

Attributes / System settings.

Attribute Management Menu

Open group attributes

Selected Group

Selected Activity

Group Attributes

Attributes / System settings.

Edit attribute

| Code | Name | Type | Validation | Rank |
|--------------------------------------|---|---------|------------|------|
| SYSTEM_PREVIEW_URI_CSS | Admin\CMS preview CSS URI | String | ✓ | 500 |
| SYSTEM_PREVIEW_URL_TEMPLATE | Admin\CMS preview URL template | String | ✓ | 500 |
| SYSTEM_BACKDOOR_TIMEOUT_MS | Admin\Communication timeout | Integer | ✓ | 500 |
| SYSTEM_PANEL_HELP_COPYRIGHT | Admin\Copyright note | HTML | ✓ | 500 |
| SYSTEM_PANEL_HELP_DOCS | Admin\Help doc link | URL | ✓ | 500 |
| SYSTEM_BACKDOOR_IMAGE_TIMEOUT_MS | Admin\Import image operation timeout | Integer | ✓ | 500 |
| IMPORT_JOB_LOG_SIZE | Admin\Import import log file size | Integer | ✓ | 500 |
| IMPORT_JOB_TIMEOUT_MS | Admin\Import job timeout | Integer | ✓ | 500 |
| SYSTEM_BACKDOOR_PRODB_IDX_TIMEOUT_MS | Admin\Indexing Bulk product index timeout | Integer | ✓ | 500 |
| SYSTEM_BACKDOOR_PRODS_IDX_TIMEOUT_MS | Admin\Indexing Single product index timeout | Integer | ✓ | 500 |

Attributes / System settings. / Admin\CMS preview CSS URI

Main Localization Validation Options Search

Code

SYSTEM_PREVIEW_URI_CSS

Type

String

☐ Secure

Description

This URI points to preview CSS. For example:
SFW: "wicket/resource/org.yes.cart.web.page.HomePage/../../../../style/yc-preview.css"
SFG: "resources/style/yc-preview.css"

Rank

500

Type of attribute affects the editor used to enter values for this attribute

The screenshot shows the 'System Preferences' interface in a 'DEMO ENVIRONMENT'. The top bar indicates the user is 'Logged in as Yes'. A yellow callout box states: 'System attributes define settings that can be set in System > Preferences'. The main table lists attributes, with 'Admin\CMS preview CSS URI' selected. A green callout points to the 'Attribute value editing' icon. A modal window titled 'Attribute' is open, showing a description: 'This URI points to preview CSS. For example: SFW: "/>

Attribute

This URI points to preview CSS. For example:
SFW: "wicket/resource/org.yes.cart.web.page.HomePage/::/::/::/::/sty
SFG: "resources/style/yc-preview.css"

Admin\CMS preview CSS URI

default value: wicket/resource/org.yes.cart.web.page

Language: Value

Buttons: Cancel, Change

Attribute

Timeout in ms for cluster calls

Admin\Communication timeout: 60000

Buttons: Cancel, Change

Attribute Usage

String type displays localisable value editor, for example















Integer type displays single input field and validates data to be valid integer

Attribute code

Attribute code should only contain **latin letters**, **numbers** and **underscore** characters. It is advisable to set code as a meaningful name (e.g. CART_MAX_ITEM_SIZE rather than ATTR0001). This will make management of the system easier in the long run (e.g. when it comes to `import`).

Value type

The following product types are supported by the platform:

| Type | Data type | Admin editor | Example | Default validation |
|--------------------|-------------------------|--|---|---|
| String | java.lang.String | Localizable string | any random string |  |
| SecureString | java.lang.String | Localizable string + masking in table view | any random string |  |
| URI | java.lang.String | Text input | file://path/to/file |  |
| URL | java.lang.String | Text input | http://mydomain/path/to/file.html |  |
| Image | java.lang.String | Image upload + preview | image.png  Uploaded images will be converted using PNG or JPG encoder only | *.png, *.jpg, *.jpeg |
| File | java.lang.String | upload + preview | documentation.pdf |  |
| SystemFile | java.lang.String | upload + preview | documentation.pdf |  |
| CommaSeparatedList | java.lang.String | Text area | Value1,Value2,Value3 |  |
| Phone | java.lang.String | Text input | +3(096)5555555 555-55-55 | ^(\+)?(\(\)0-9\-)+\$ |
| Email | java.lang.String | Text input | bob.doe@domedomain.com | ^[_a-z0-9-]+\(_\[_a-z0-9-]+\)*@[a-z0-9-]+\(_\[_a-z0-9-]+\)*(\.[a-z0-9-]+)\$ |
| HTML | java.lang.String | Text area | <div>piece of html</div> |  |
| Float | java.lang.Float | Text input | 10.2 -10.2 10 | ^((\-[0-9]+) ([0-9]*))\.[0-9]+\$ |
| Integer | java.lang.Integer | Text input | 10 -10 | ^\-[0-9]+\$ |
| Boolean | java.lang.Boolean | Checkbox | true false  either true or false | (true false) |
| Date | java.time.LocalDate | Text input | 2014-04-29  For example 1st February 2010 would be 2010-02-01 . | ^[0-9]{4}\-([0][1-9] [1][0-2])\-([0][1-9] [1-2][0-9] [3][0-1])?\$ |
| DateTime | java.time.LocalDateTime | Text input | 2014-04-29 14:42:55  For example 1st February 2010 1:45pm would be 2010-02-01 13:45:00 . Time part is optional | ^[0-9]{4}\-([0][1-9] [1][0-2])\-([0][1-9] [1-2][0-9] [3][0-1]) (([0][0-9] [1][0-9] [2][0-3]):[0-5][0-9]:[0-5][0-9])?\$ |
| Timestamp | java.time.Instant | Text input | 1462372975000  Unix timestamp | ^\-[0-9]+\$ |
| Locked | java.lang.String | read only | |  |

Validation

Validation regular expression and message allow to give hints to Admin app editors so that more advanced validation can be performed as opposed default ones for given type.

Customer facing attributes e.g. CUSTOMER attributes that participate in registration/profile forms or ADDRESS attributes that participate in address capture forms will use the validation expression on storefront and corresponding error message will be displayed to customers.

Search options

| Option | Purpose |
|---------------------------------------|---|
| Store value in index | Saves value in search index as is. This makes the index larger in size but enables to use this value in CMS on category browsing and search pages |
| Value is searchable | Analyses the value and enables searching via search box by this value using normal, partial and fuzzy searches |
| Values is primary key (exact matches) | Strict version of "searchable" which forces exact matches only (e.g. useful for SKU or model numbers) |
| Value is navigatable | Instructs the system to create a facet index for this value and enable auto generation of filter navigation blocks |