Version

3.4

CMDBuild®

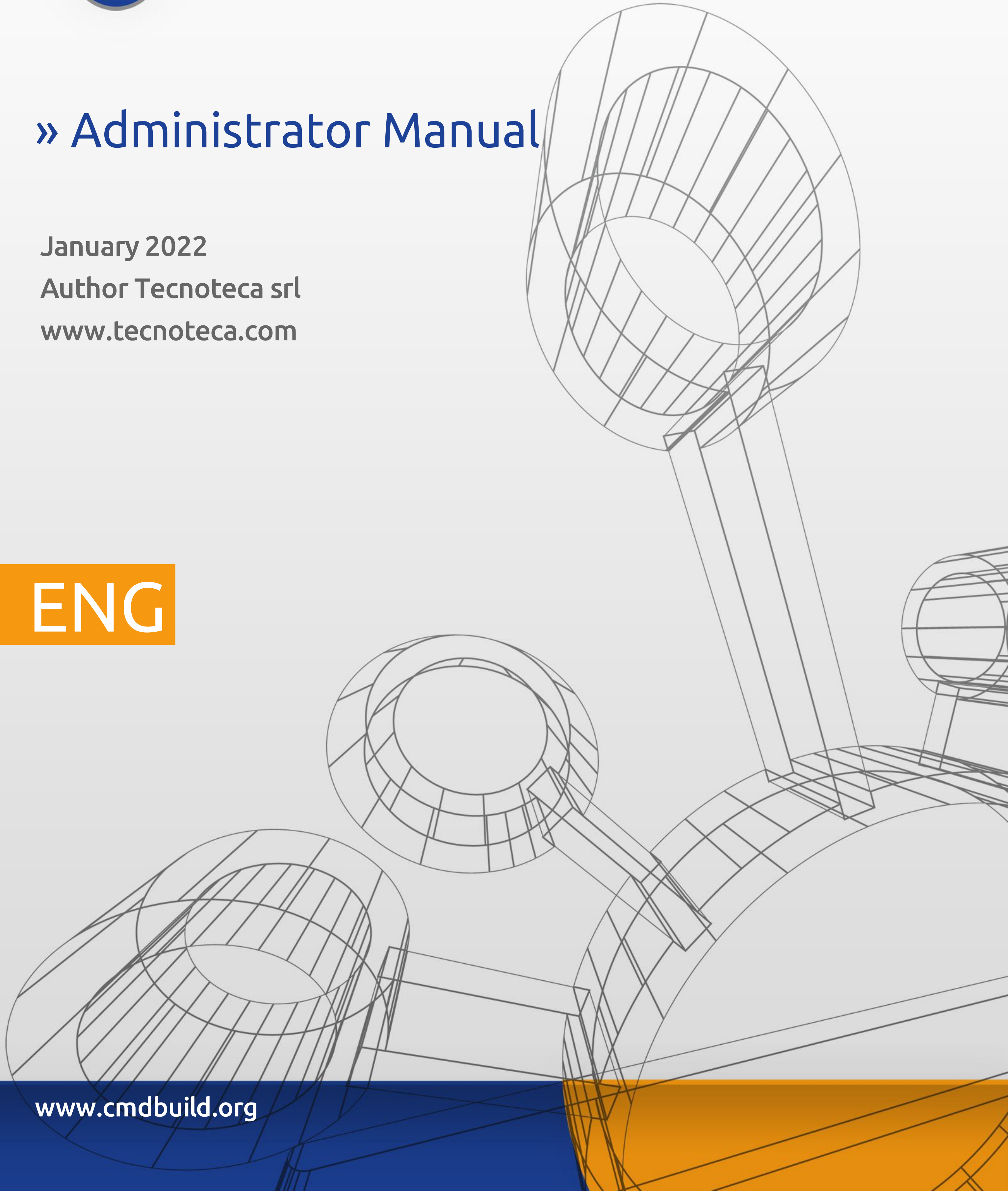
» Administrator Manual

January 2022

Author Tecnoteca srl

www.tecnoteca.com

ENG



www.cmdbuild.org

CMDBuild ® uses many great technologies from the open source community:
PostgreSQL, Apache, Tomcat, Eclipse, Ext JS, JasperStudio, Enhydra Shark, TWE, OCS Inventory, Liferay, Alfresco, GeoServer, OpenLayers, Quartz, BiMserver, Xeokit.
We are thankful for the great contributions that led to the creation of these products.

CMDBuild ® is a product of Tecnoteca S.r.l. which is responsible of software design and development, it's the official maintainer and has registered the CMDBuild logo.

The official website is http://www.cmdbuild.org

Contents

1. Introduction

1.1. The application

CMDBuild is an open source web environment for the configuration of custom applications for the Asset Management.

On the one hand, it provides native mechanisms for the administrator, implemented in a "core" code which has been kept separated from the business logic, so that the system can be configured with all its features.

On the other hand, it generates dynamically a web interface for the operators, so that they can keep the asset situation under control and always know their composition, detachment, functional relations and how they update, in order to manage their life-cycle in a comprehensive way.

The system administrator can build and extend his/her own CMDB (hence the name of the project), modelling the CMDB according to the company needs; a proper interface allows you to progressively add new classes of items, new attributes and new relations. You can also define filters, "views" and access permissions limited to rows and columns of every class.

Using external visual editors, the administrator can design workflows, import them into CMDBuild and put them at operators' disposal, so that they can execute them according to the configured automatisms.

Similarly, using external visual editors, the administrator can design various reports on CMDB data (printouts, graphs, barcode labels, etc.), import them into the system and put them at operators' disposal.

The administrator can also configure some dashboards made up of charts which immediately show the situation of some indicators in the current system (KPI).

A task manager included in the user interface of the Administration Module allows you to schedule various operations (process starts, e-mail receiving and sending, connector executions) and various controls on the CMDB data (synchronous and asynchronous events). Based on their findings, it sends notifications, starts workflows and executes scripts.

The interoperability with other systems is managed through the CMDBuild BUS Service, called WaterWAY.

Thanks to document management systems that support the CMIS standard (Content Management Interoperability Services) - among which there is also the open source solution Alfresco - you will be able to attach documents, pictures, videos and other files. It is possible to define document categories differentiated by class and sets of metadata differentiated by category.

There is also a Scheduling, which can be supplied both automatically when filling in a data card and manually. This Scheduling will manage single or recurring deadlines related, for example, to certifications, warranties, contracts with customers and suppliers, administrative procedures, etc.

Moreover, you can use GIS features to georeference and display assets on a geographical map (external map services) and / or on vector maps (local GeoServer and spatial database PostGIS) and BIM features to view 3D models (IFC format).

The system also includes a REST webservice, so that CMDBuild users can implement custom interoperability solutions with external systems.

Furthermore, CMDBuild includes two external frameworks:

- the CMDBuild Advanced Connector, which is written in Java and can be configured in Groovy: it helps the implementation of connectors with external data sources, i.e automatic inventory systems, virtualization or monitoring ones (supplied with non-open source license to the users that subscribe the annual Subscription with Tecnoteca)
- the CMDBuild GUI Framework, which helps the implementation of additional graphical interfaces, i.e. web pages (simplified for non technicians) that have to be published on external portals and that are able to interact with the CMDB through the REST webservice

CMDBuild includes a mobile interface (for smartphone and tablet). It is implemented as multi-platform app (iOS, Android) and is able to interact with the CMDB through the REST webservice (supplied with non-open source license to the users that subscribe the annual Subscription with Tecnoteca).

CMDBuild is an enterprise system: server-side Java, web Ajax GUI, SOA architecture (Service Oriented Architecture), based on webservice and implemented by using the best open source technologies and following the sector standards.

CMDBuild is an ever-evolving system, which has been released for the first time in 2006 and updated several times a year in order to offer more features and to support new technologies.

1.2. Official website

CMDBuild has a dedicated website: http://www.cmdbuild.org

The website gathers a lot of documents on technical and functional features of the project: brochures, slides, manuals (see next paragraph), testimonials, case histories, newsletters, forums.

1.3. CMDBuild modules

The CMDBuild application includes two main modules:

- the Administration Module for the initial definition and the next changes of the data model and the base configuration (relation classes and typologies, users and authorization, dashboards, upload report and workflows, options and parameters)
- the Management Module, used to manage cards and relations, add attachments, run workflow processes, visualize dashboards and execute reports

The Administration Module is available only to the users with the "administrator" role; the Management Module is used by all the users who view and edit data.

1.4. Available manuals

This manual is dedicated to the Administration Module, through which the administrator can configure data, define users and permissions, and perform other tasks.

You can find all the manuals on the official website (http://www.cmdbuild.org):

- system overview ("Overview Manual")
- system usage for operators (“User Manual”)
- installation and system management ("Technical Manual")
- workflow configuration (“Workflow Manual”)
- webservice details and configuration (“Webservice Manual”)

1.5. Applications based on CMDBuild

Tecnoteca has used the CMDBuild environment in order to implement two different pre-configured solutions:

- CMDBuild READY2USE, for the management of assets and IT services, oriented to internal IT infrastructures or services for external clients (www.cmdbuildready2use.org) according to the ITIL best practice (Information Technology Infrastructure Library)
- openMAINT, for the inventory management of assets, properties and related maintenance activities (www.openmaint.org)

Both applications are released with open source license, except for certain external components (data sync connectors, Self-Service portal, mobile APP, etc.), that are reserved to the users that subscribe the annual Subscription with Tecnoteca.

2. Getting Started

2.1. What is a CMDB

A CMDB (Configuration Management Data Base) is a storage and consultation system that handles the asset information of a company.

The idea of CMDB was born in an IT environment (it is a fundamental component of ITIL best practice - Information Technology Infrastructure Library). In this manual it is extended and applied in a general context where you have to know, manage and control assets (Configuration Items or CIs).

It is the official central repository and provides a consistent view of the items that have to be managed.

It's a dynamic system that represents the current situation and knowledge of the inventory of assets and the related connections.

2.2. What's CMDBuild

CMDBuild is a robust, customizable and extensible CMDB solution.

Providing an extensible solution means providing an open and dynamic system that can be easily designed, configured and extended by the system administrator in different phases in terms of types of objects, attributes and relations (domains), of workflows, reports, dashboards, interoperability solutions with external system, etc.

Since there are not two organizations that manage their assets in the same way, we decided to set, as the primary CMDBuild feature, the system flexibility, developing features to configure the whole system (see the CMDBuild Overview Manual, chapters 4, 5 and 6) and to optimally design the application according to your needs.

From a technical point of view, the configurability is based on a wide use of "metadata".

The CMDBuild "core" code interprets metadata creating also the web interface for operators, who have the possibility of updating the CMDB, start and advance processes, execute reports, view dashboards, etc.

Despite of its name, CMDBuild is not only a Modeling Environment for CMDB applications, i.e. limited within the management of a database of items (Configuration Item).

CMDBuild mechanisms allow you to manage Configuration Items within the entire life cycle by using workflows, business rules, documents, reports, georeferences, interoperability with other systems, etc.

Thanks to CMDBuild you can implement complete Asset Management applications in the various application contexts.

A system based on the configurability has its clear advantages:

- it can adapt to your organization and work procedures
- possibility of a gradual carrying out of the system, reducing the organizational impact
- easy response to organizational and technological editing
- more independence for users
- same “core” code (managed “per product”) used in different vertical solutions

2.3. Design criteria

At first, it's important to:

- choose a detail level proportional to the organization needs, human resources, financial, information and technology
- identify and involve the staff who will set up (Administrator) and update (Operator) information; an outdated system produces costs and no results
- introduce the new application in an organizational system based on procedures, roles and responsibilities that help the IT structure to correctly manage the information

A successful Asset Management project must consider impact and changes introduced by the system and must acquire the explicit approval of organization's managers.

2.4. Where to start

Implementing an Asset Management application through CMDBuild requires a preliminary design to manage the initial schema, ie identify main interests and options; then, the system can be easily extended as needed at a later date.

2.4.1. Data model

Regarding the data model, we recommend you to start by managing a complete and accurate set of objects and relationships, and then extend the system once you've become more familiar with CMDBuild rules and usage.

In particular, you should identify:

- the types of items to manage (classes): IT assets (computers, peripherals, network systems, phone devices, software, etc.), assets related to real estates (buildings, plants, technical devices, furniture), assets related to production plants (factories, plants, machines), other types of assets (motor vehicles, electro-medical instruments, etc.)
- "attributes": useful to define each class (e.g. for an asset there will be a code, a description, the supplier, the purchase date, etc.) and the related type of datum (string, long text, integer, decimal, floating point, date, "lookup" list, reference, geographical attribute, or open or closed polygon)
- relations between classes
- "attributes" useful to describe each "domain" (e.g. the role of each person in charge of a service, the type of dependency between two assets, etc.) and the related type of data (string, long text, integer, decimal, floating point, date, "lookup")
- user accounts for every class

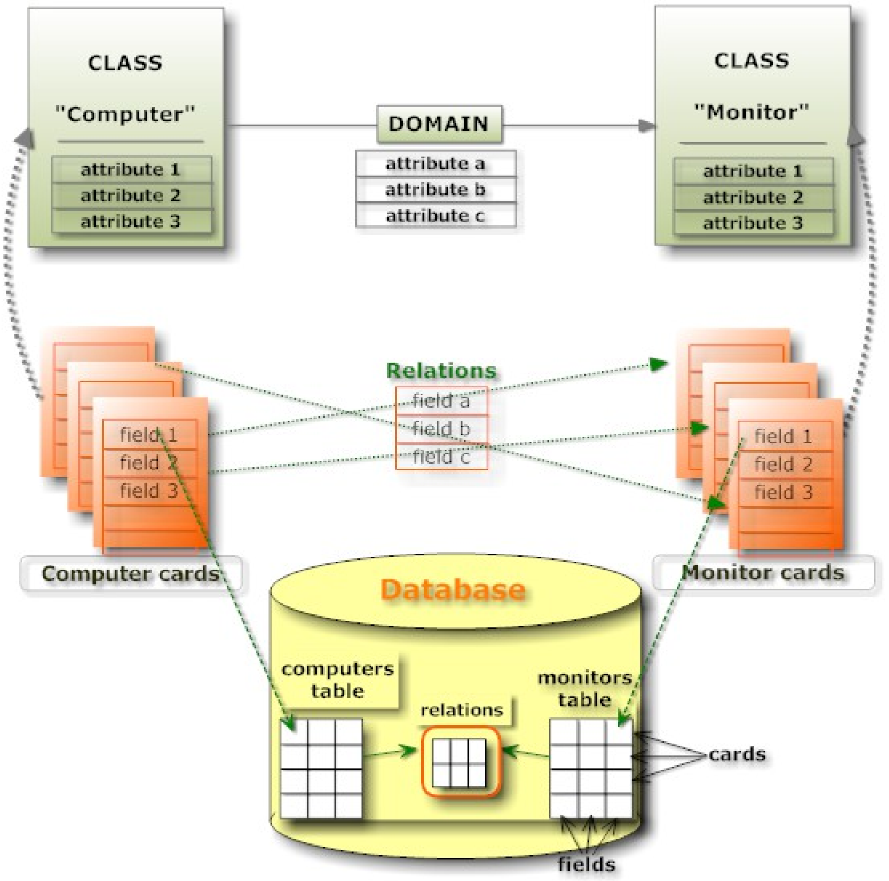
Another thing to consider is the hierarchy of classes since, in CMDBuild, it's possible to define abstract classes (Superclasses) which can be used as templates (for example "Computer") and then derive subclasses (for example, "Desktop", "Laptop", "Server") which will include the real data and both the shared attributes (specified in the superclass) and the ones specified in the subclass, in addition to the domain relations of the superclass and the specific domains.

It's important to identify a hierarchy that meets the current and future needs of the organization since a class can not be automatically converted into a superclass.

Once the entity-relationship model has been defined, you have to define classes and related attributes / data types.

At the end of this operation you should:

- use the Administration Module to model the system you've designed using E-R editor
- use the Management Module to insert, update and display cards



2.4.2. Processes

One important added value of CMDBuild is the possibility of defining processes (workflows) for operators to execute the management activities.

As for the data model, CMDBuild doesn't supply standard processes implemented in a static way, but provides a generic configuration system used according to the specific needs of every single company.

Workflows are designed using the open source external visual editor TWE (Together Workflow Editor), and then imported into CMDBuild (standard XPDL files) in order to be executed with the Together Workflow Server engine.

In the IT environment, these basic mechanisms allow the configuration of all processes provided by ITIL "best practices", included Incident Management, Change Management, Request Fulfillment, Service Catalog, etc.

In the Facility Management environment, all programmed and breakdown maintenance processes can be configured.

For a correct work planning, you have to start from the users' needs and define the specifications of the new process through a careful detail analysis.

In particular, the analysis step will detect:

- the actors of the process (users' groups or roles)
- the workflow, with:
  - the user's activities (interactive)
  - the automatic activities (scripts carried out by the system)
  - the transactions among activities (obliged or contingent)
- for every user's activity (data entry form):
  - the list of information with the related types of data and presentation modes (read-only, reading and writing, possible compulsoriness)
  - widgets to configure in order to execute additional operations (cards display, creation of relations, management of e-mail sending, attachment upload, etc.)
- for every automatic activity, the types of operations to implement (editing of cards or relations, report process, e-mail sending, external webservice calls, etc.)

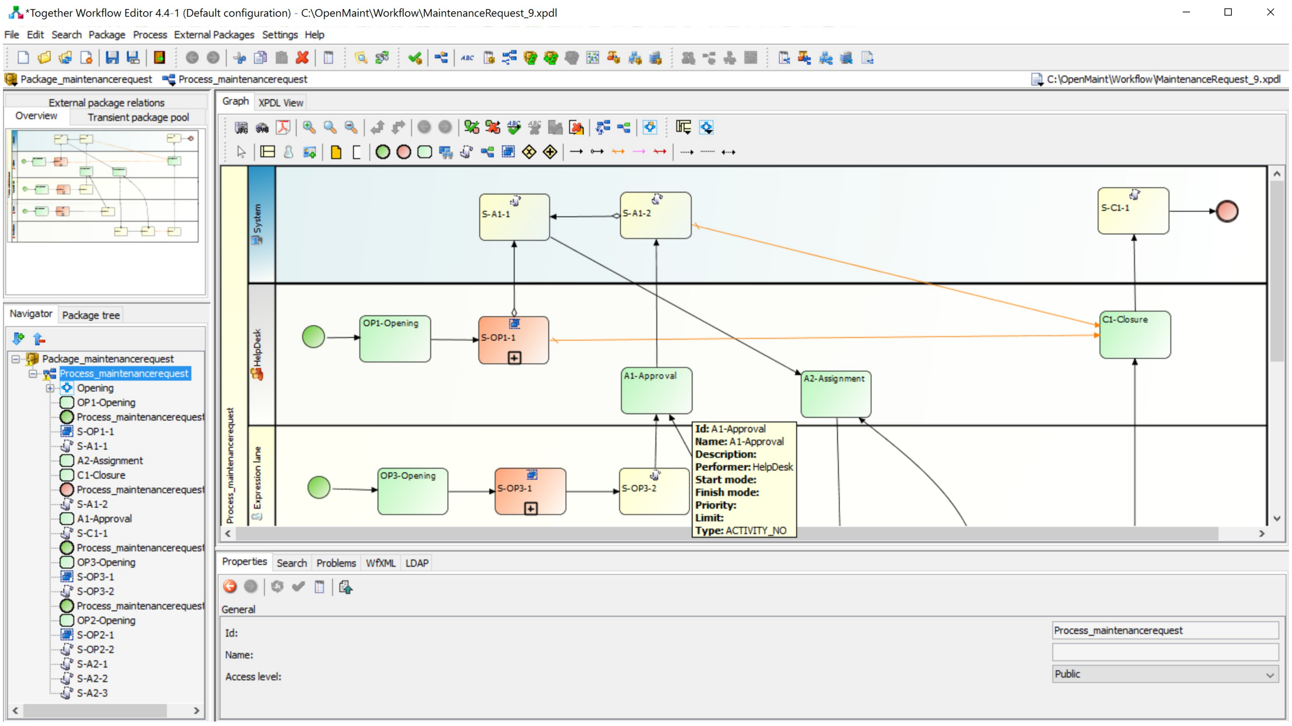
The creation step will:

- configure through the Administration Module the persistence elements of the CMDB data (create the new process "class", create the attributes detected during the analysis, create "domains" among the process and other classes or processes)
- create through the Administration Module the groups of users involved in the process
- export through the Administration Module the structure of the new process, in order to provide the TWE visual editor with users' groups (roles) and process attributes
- design through the TWE visual editor the process flow, configuring in particular:
  - the "lanes" with the involved users' roles
  - the user's activities where attributes and widgets are located
  - the automatic activities, in order to write the scripts that implement the expected logics
- save the XPDL file from the TWE visual editor and import it into CMDBuild through the Administration Module

Once concluded the operations described above, the new process can be used through the Management Module of CMDBuild, which will automatically interpret and execute its flow through the workflow engine (the Tecnoteca River or the Together Workflow Server).

The above mentioned operations can be carried out when you need to edit an imported process, but the changes must be received only through the new process instances which will be started.

Here's a sample of the user interface of the TWE visual editor.



*Together Workflow Editor 4.4-1 (Default configuration) - C:\OpenMaint\Workflow\MaintenanceRequest_9.xpdl
File Edit Search Package Process External Packages Settings Help
Package_maintenancerequest
Process_maintenancerequest
C:\OpenMaint\Workflow\MaintenanceRequest_9.xpdl
External package relations
Overview
Transient package pool
Navigator
Package tree
Package_maintenancerequest
Process_maintenancerequest
Opening
OP1-Opening
Process_maintenancerequest
S-OP1-1
S-A1-1
A2-Assignment
C1-Closure
Process_maintenancerequest
S-A1-2
A1-Approval
S-C1-1
Process_maintenancerequest
OP3-Opening
S-OP3-1
S-OP3-2
Process_maintenancerequest
OP2-Opening
S-OP2-1
S-OP2-2
S-A2-1
S-A2-2
S-A2-3
Graph
XPDL View
Process_maintenancerequest
System
HelpDesk
Expression lane
S-A1-1
S-A1-2
S-C1-1
OP1-Opening
S-OP1-1
C1-Closure
A1-Approval
A2-Assignment
OP3-Opening
S-OP3-1
S-OP3-2
Properties
Search
Problems
WfXML
LDAP
General

Id: A1-Approval
Name: A1-Approval
Description:
Performer: HelpDesk
Start mode:
Finish mode:
Priority:
Limit:
Type: ACTIVITY_NO

Id: Process_maintenancerequest
Name:
Access level: Public

Please see the CMDBuild Workflow Manual for further details.

2.4.3. Reports and dashboards

As for all other activities, you have to start from the users' needs and define the specifications of each report and dashboard through a detail analysis.

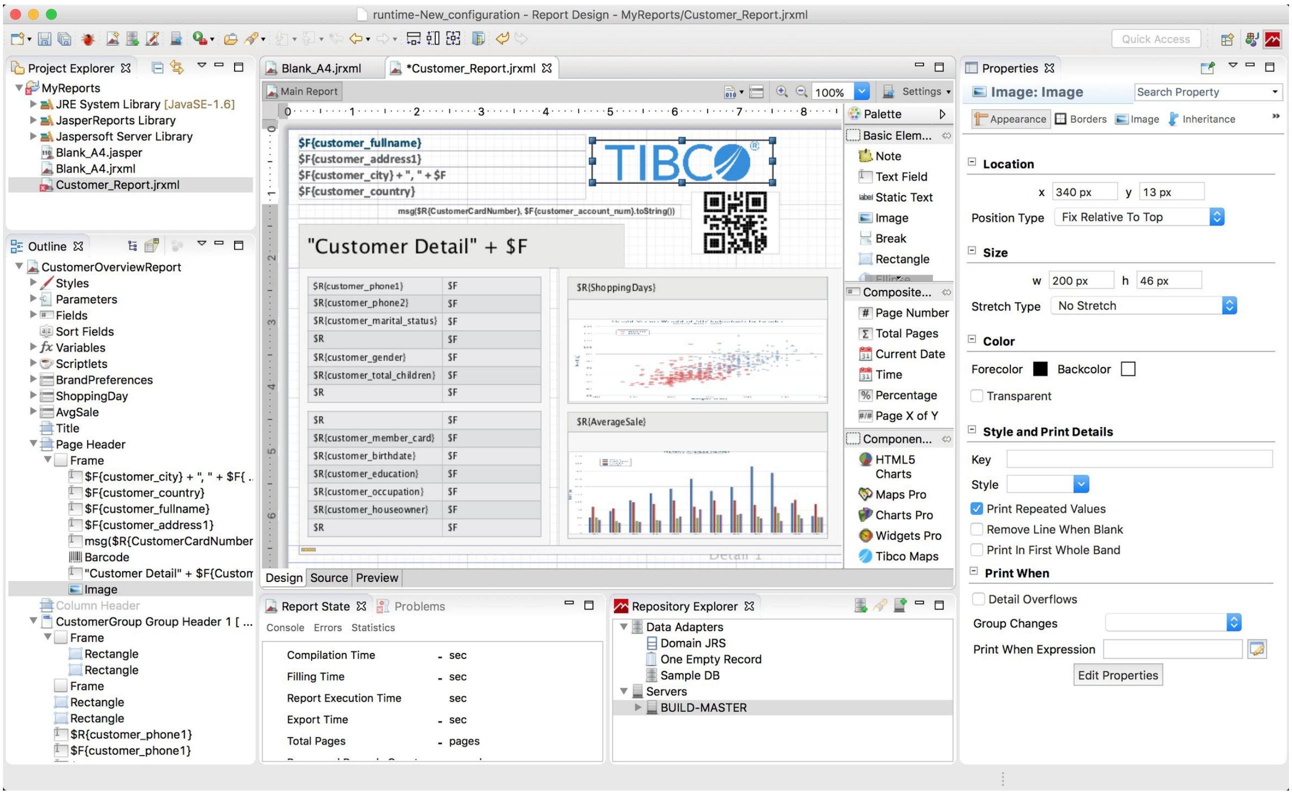
For each report it is important to know:

- the general page layout (orientation, header, footer, date, page number, widgets like pictures or geometric shapes, etc.)
- the types of information you have to include when printing and the ways to extract the database (SQL query with column list, join, filters, etc.), to organize, to group, to totalize, etc.
- the organization of extracted information (in a card, in a table, with subreports, formatting, calculations, etc.)
- any particular needs (integration of charts, barcode, pivot tables, etc.)
- any launch parameters and related default values

In order to design custom reports, CMDBuild uses the external visual editor JasperStudio from the open source suite JasperReports, which allows you to manage intuitively the layout and the formatting of the information extracted from the database.

Once the report is designed in JasperStudio, it is saved as XML file, imported in CMDBuild from the Administration Module and made available in operators' menus.

Here's a sample of the user interface of the JasperStudio visual editor.



x 340 px y 13 px

Position Type: Fix Relative To Top

w 200 px h 46 px

Stretch Type: No Stretch

Forecolor ■ Backcolor □

□ Transparent

Key

Style

☑ Print Repeated Values

□ Remove Line When Blank

□ Print In First Whole Band

□ Detail Overflows

Group Changes

Print When Expression

Compilation Time - sec

Filling Time - sec

Report Execution Time sec

Export Time - sec

Total Pages - pages

Regarding the dashboards you have to know, for each chart:

- the number and location of the charts that have to be included in the dashboard
- for each chart:
  - the information to represent and the ways to extract it from the database
  - the type of chart (pie, bar, line, gauge)

The configuration of custom dashboards can be done from the Administration Module of CMDBuild, after the creation - through the tools of the PostgreSQL database - of a specific function that extract the important information.

2.4.4. User interface and custom business logics

The CMDBuild Data Management Module interprets metadata defined in the Administration Module in order to create automatically the user interface to consult and update data, start and execute processes, print reports, display dashboards, etc.

It can happen that it is useful or necessary that operators have at their disposal custom pages, such as provided with views on data, several data tables on the same page, calculated fields or custom graphs, etc.

In this case, CMDBuild provides the Custom page mechanism to design user interface pages which are completely customized.

In CMDBuild 3 the custom pages can be implemented in javascript languages within the ExtJs mechanisms, instead of using the CMDBuild GUI Framework based on JQuery like in CMDBuild 2, in order to reuse already available CMDBuild GUI components and get a greater total integration.

It is also possible to program in javascript some features with user interface which can be started from the contextual menu, custom widgets, form triggers.

And, finally, it is possible to program in javascript dynamic custom behaviours when editing a form: hiding or making not editable an attribute according to other attributes, enhancing an attribute according to other attributes, defining custom validators for an attribute.

2.4.5. Interoperability solutions

CMDBuild implements a Service Bus that manages internal communication and facilitates the interoperability with applications and external systems, through standard services for coordination, security, messaging, routing and transformations.

Tasks can be configured through descriptor files.

The CMDBuild Service BUS is designed with a plugin architecture, where standard basic components available for everybody, standard advanced components supplied with the subscription and custom components can coexist.

2.4.6. Other activities

All other configurations can be done working directly on the Administration Module of CMDBuild.

Please refer to the following chapters for detail information on the intervention modalities.

3. Role and features

3.1. The role of administrator

The role of the Administrator is fundamental for a correct management of an Asset Management project based on CMDBuild.

The administrator is the DBA (Data Base Administrator) of the CMDB.

The administrator is also the contact person for the Business, in order to manage the analysis of data model, processes, reports and dashboards.

He/She is responsible for data security, which can be managed by creating users' groups, defining permissions for each role and assigning every user to the appropriate role or by applying Multitenant mechanisms.

He/She configures also menus, recurring tasks, filters and views, and sets technical parameters of the system according to the used infrastructure.

He/She or a collaborator can configure the mechanisms of the client interface of CMDBuild in order to implement the required business logics: custom pages, form trigger, contextual menus, rules for data validating, etc.

The Administrator should have good programming skills (relational database and SQL language, Javascript language to work on "client" logics, etc.) and system skills. Furthermore he/she should be also a good analyst and understand the explicit and implicit requirements expressed by the company's business contact people.

3.2. The features of the Administration Module

The administrator's interface, called Administration Module, allows to process all configuration operations of the system.

In general, the Administrator's skills are:

- configuring the data model of the CMDB, in terms of classes, attributes, relations
- configuring the workflows through the definition of the data persistence and import of the XPDL descriptor of the flow
- configuring user interfaces and custom business logics (custom pages, contextual menus, form trigger widgets, dynamic custom behaviours when editing a card, etc.)
- configuring the reports through the import of the XML descriptor that describes its layout and dashboards
- configuring the other user interface components: views, filters, menus, navigation trees
- configuring the security aspects (roles, users, permissions, tenants)
- configuring technical component of the system: template for interactive operations or import / export batches through file, task manager, timetable rules template, GIS and BIM, localization, system management and other parameters

In more detail, the features in Administration Module (which will described in the following chapters) include:

- summary dashboard for the CMDB functioning and contents

- configuration of the CMDB data model and of certain behaviours of the related user interface:
  - creation of new classes, i.e. new object typologies
  - creation and change of (common and geographical) attributes in a class and possible definition of presentation layout
  - definition of behaviours on the user interface: categories of attachments, data organisation, attributes grouping in fieldsets, form trigger for the implementation of business logics, contextual menus, widgets
- workflow configuration related to the collaborative CMDBuild processes:
  - definition of persistence information in workflows (classes of a particular type)
  - import of the flow externally designed with TWE tool (XPDL file)
  - definition of the behaviours on the user interface, which are similar to the class ones
- creation of domains, i.e. types of relations and any related attributes among couples of classes
- creation of tablelists (lookup) to manage attributes with closed values
- definition of filters and views:
  - creation of predefined filters used during the data reference on the data cards
  - creation of views, both through filters on simple and multitable data and SQL queries
- definition of navigation trees (used in the GIS and in the Relation Graph)
- creation and change of dashboards, configured from user interface
- report creation and change, by importing the XML layout designed externally with the JasperStudio tool
- definition of custom menus for different users' groups
- configuration of custom elements for the user interface: custom pages and components
- definition of the way to manage attachments: document categories and document models (groups of metadata)
- safety management:
  - creation of users and groups of users (roles)
  - definition of permissions on classes and processes (also restricted to subsets of rows and columns) and on other items managed in the system (views, search filters, reports, dashboards)
- e-mail management (accounts and templates)
- template configuration for import / export (CSV / XLS, XLSX, DB tables, DWG, IFC)
- configuration of the Task Manager and of the operations to execute in the background
- configuration of Bus Service features and components (WaterWAY)
- configuration of the timetable feeding modes
- configurations of GIS and BIM
- localization into several languages
- system management
- definition of the system parameters

4. User Interface

4.1. Technology

The web desktop user interface of CMDBuild, both in the Administration Module and in the Data Management Module, is implemented through the framework Sencha ExtJS, a JavaScript library for the building of web interactive applications with AJAX technology.

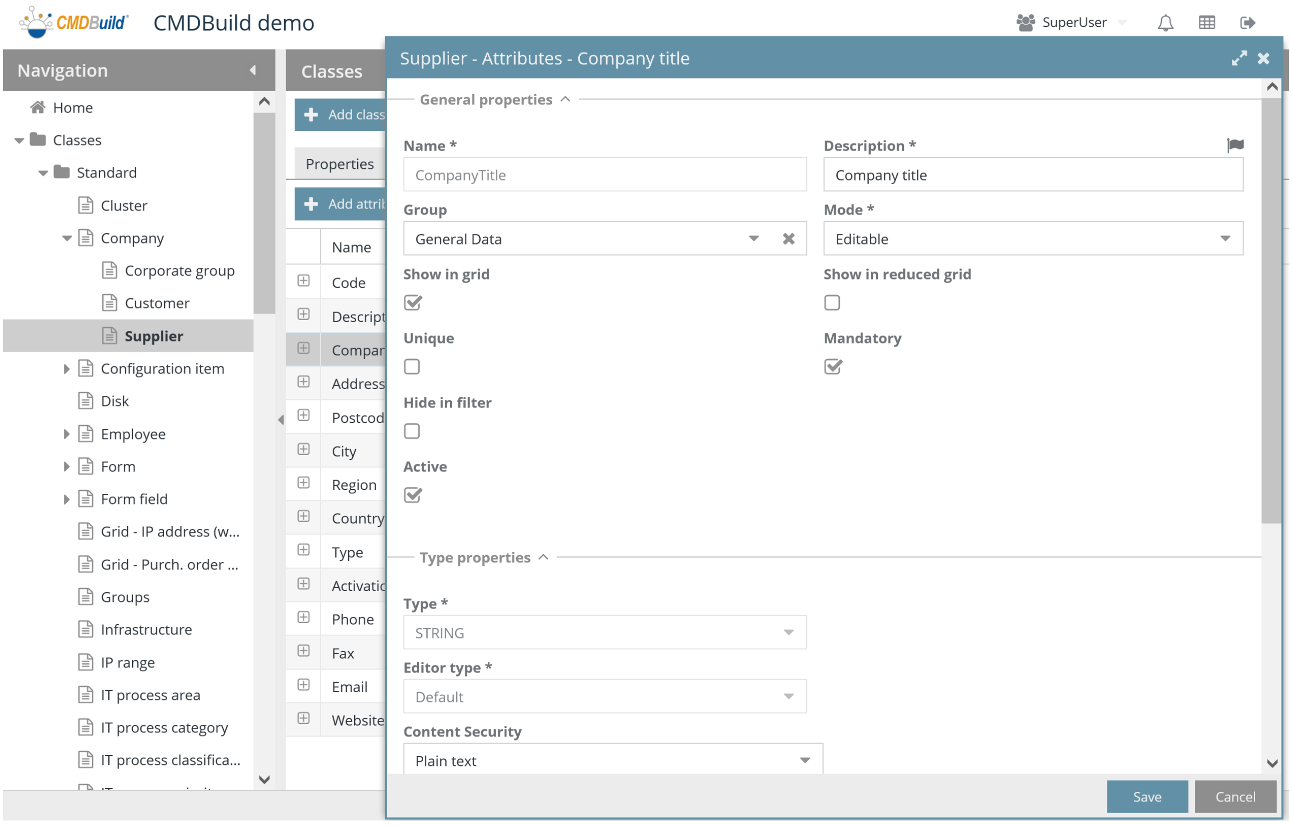
Every operation the user carries out on the web interface is transmitted to the server code, written in Java language, through calls to the standard REST webservice methods of CMDBuild.

After that, the server carries out the user's requests converting them in operations on the PostgreSQL database.

4.2. General design criteria

The user interface of the Administration Module includes the following main elements:

- header
- left side menu, to access the single features of the Administration Module
- central work area, which includes the information in cards or tables
- popup window, to manage in a completely mode the information of a single row belonging to a grid presented in the central area
- footer



Supplier - Attributes - Company title

Name * CompanyTitle

Description * Company title

Group General Data

Mode * Editable

Show in grid ☑

Show in reduced grid □

Unique □

Mandatory ☑

Hide in filter □

Active ☑

Type * STRING

Editor type * Default

Content Security Plain text

The header includes:

- on the left, the application logo (which has to be kept according to the distributing license of CMDBuild), the user's logo and the instance name
- on the right the user's name, the access to the preference menu, the transition to the Data Management Module, the notification management and the exit from the application

The menu on the left includes a folder for each function of the Administration Module, with the item lists (classes, processes, domains, lookup lists, views, search filters, dashboards, reports, menus, navigation trees, users' groups) or function lists.

The central working area includes:

- on the top, the controls:
    - buttons for actions on the type of item to manage, for example adding elements, printing, etc.
    - other controls for operations on the type of item to manage, for example the search
    - TABs to access other pages to manage the type of selected item, in turn including buttons and controls
- after that, the proper information organized in cards or grids (tables):
    - in case of cards, you can execute the expected operations from the available icons, for example enter in editing mode, fill in the form fields divided in "fieldsets", confirm or cancel the changes
    - in case of grids (tables) you can:
        - in the header: organize in one column, select the columns
        - on rows: highlight, enter in editing mode, expand the detail information, select
        - on the detail information: execute operations from the available icons

The popup window, if any, is a card for the filling in of the form fields, which can be divided in "fieldsets", confirming or canceling of changes.

The footer includes:

- the official website URL
- a page with useful links and credits
- the copyright notice

The details above are part of a standard setup but may differ in case of specific installations.

4.3. Localization

In CMDBuild there are two localization typologies, i.e. translation into the user's language:

- first-level localization, related to the only base CMDBuild user interface (menu texts and standard buttons, headers, footers, etc.)
- second-level localization, related to all elements configured for each verticalization of the base environment (class names, attributes, domains, lookups, etc.)

The first-level localization is managed through external system files (json format).

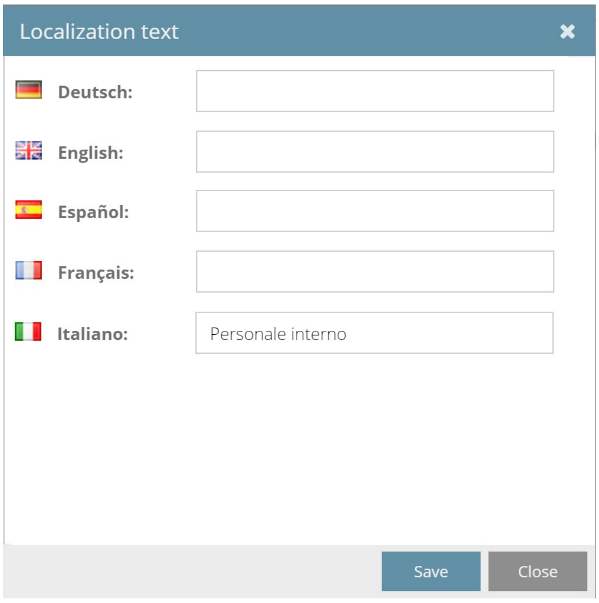
The second-level localization is managed in two ways:

- locally in each page where there are texts to localize, using an appropriate button that opens a popup window for the translation in the enabled languages
- globally, in the specific menu entry "Localizations", which allows to work on all texts to localize available in the system

For the local translation there is this icon, which you can see above the field to be localized:

Translation into other languages	⚑

which opens the following popup window:



Localization text

Deutsch:

English:

Español:

Français:

Italiano: Personale interno

From the Management Module, by selecting your language, all localizable texts will be translated into the chosen language.

5. Accessing the application

5.1. General Information

The Administration Module is available only to the administrator and can be used to configure the current CMDBuild instance according to the logic model and to make changes and useful extensions at a later stage.

The changes made will be immediately active and available to operators within the Management Module.

5.2. Computer requirements

CMDBuild is a web-based environment, so both modules are available using a standard web browser.

The user must have a recent web browser (Mozilla Firefox, Google Chrome, Microsoft Edge, Apple Safari).

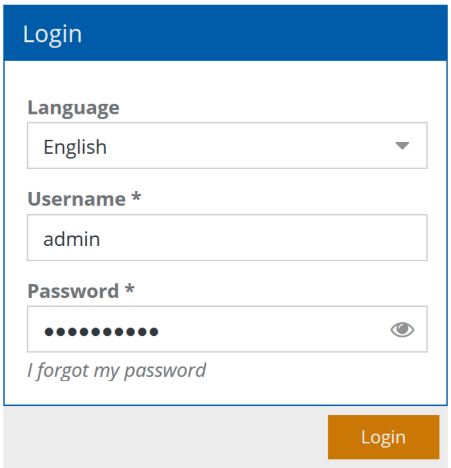
A recent CPU and appropriate RAM is also required (at least 4 GB).

The web architecture ensures complete usability to any organization that operates in multiple locations; any entrusted client can connect and interact with the system.

5.3. Authentication

You must login to use the application.

The login form requires a language selection (if the application has been configured to handle multiple languages), the username and the password.



Login

Language: English

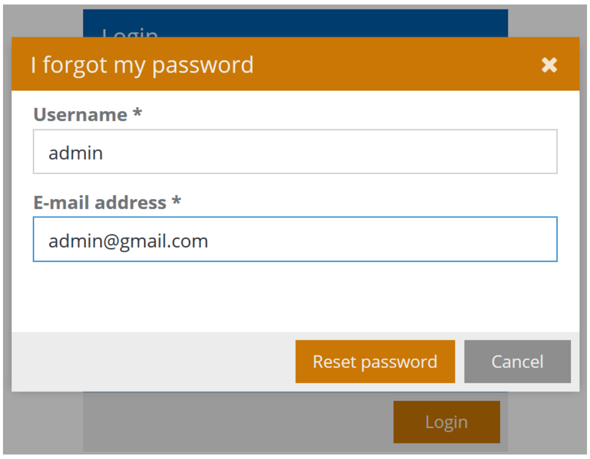
Username *: admin

Password *: ●●●●●●●●●●

I forgot my password

In order to access to the Administration Module, the user must have an administrator account.

In case users forgot their password, they can access the following form by clicking on the link "I forgot my password":



I forgot my password

Username *

admin

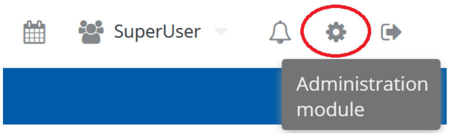
E-mail address *

admin@gmail.com

The system checks that the email address corresponds to the one registered in the database and sends then an email with a temporary code (token). By inserting that code in the password space, the form for changing the password will appear. At this point, the user can regularly access the application with the new password. Please note that the password has to meet the security policies defined in the proper page for the system configuration.

CMDBuild is released with a preconfigured administrator account, with username "admin" and password "admin". You can use this for the first access to the application. It is necessary to edit the password to avoid unauthorized access and any following security issue.

Once entered CMDBuild and having the necessary permissions, you can access the Administration Module through the icon on the top right (highlighted in red in this image):



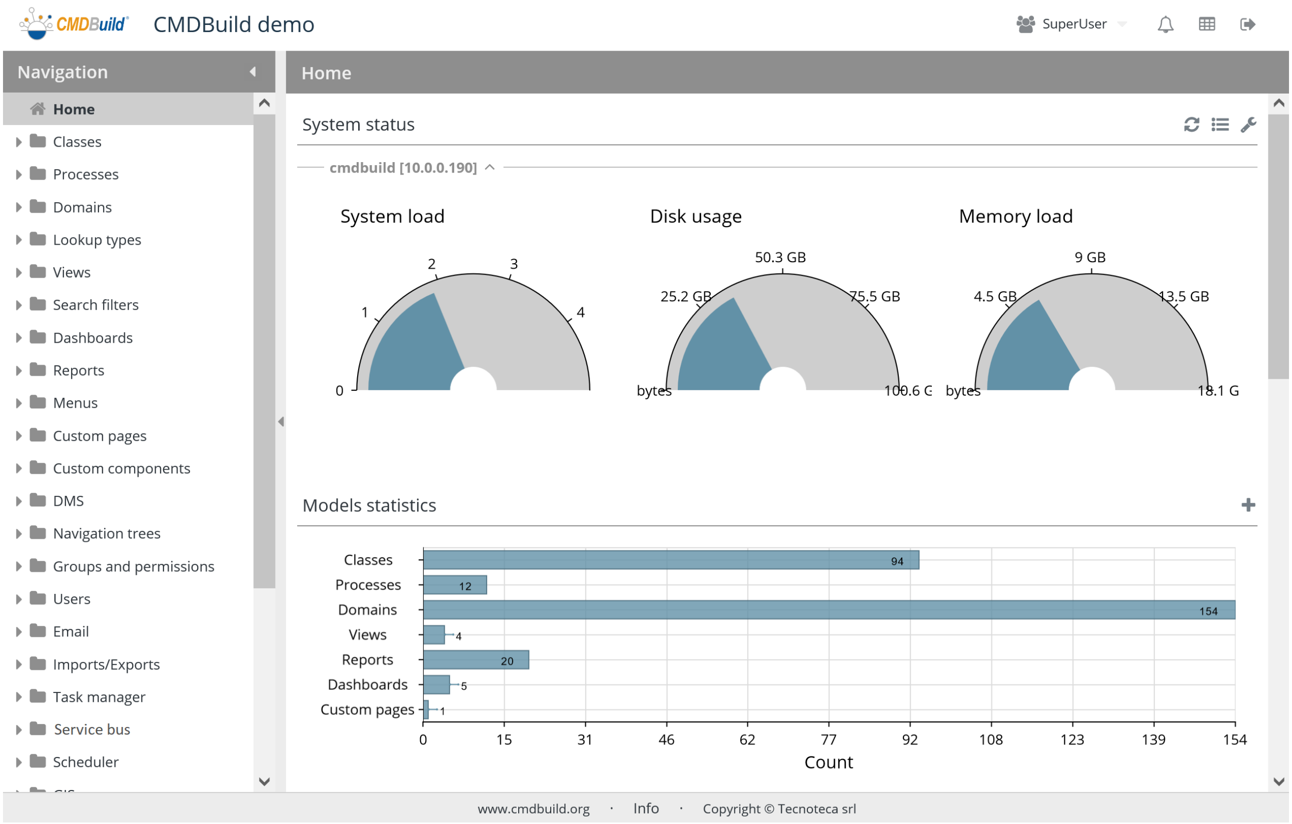
6. Home

When you open the Administration Module, you will be offered a homepage showing some graphs and information useful to understand the state of the system.

A first batch of information includes:

- system status (with a set of graphs for each node in case of cluster architectures): system load, used disk space, RAM memory in use
- stats on types of entities configured in CMD Build: number of classes, number of processes, number of domains, number of visits, number of views, number of reports, number of dashboards, number of custom pages

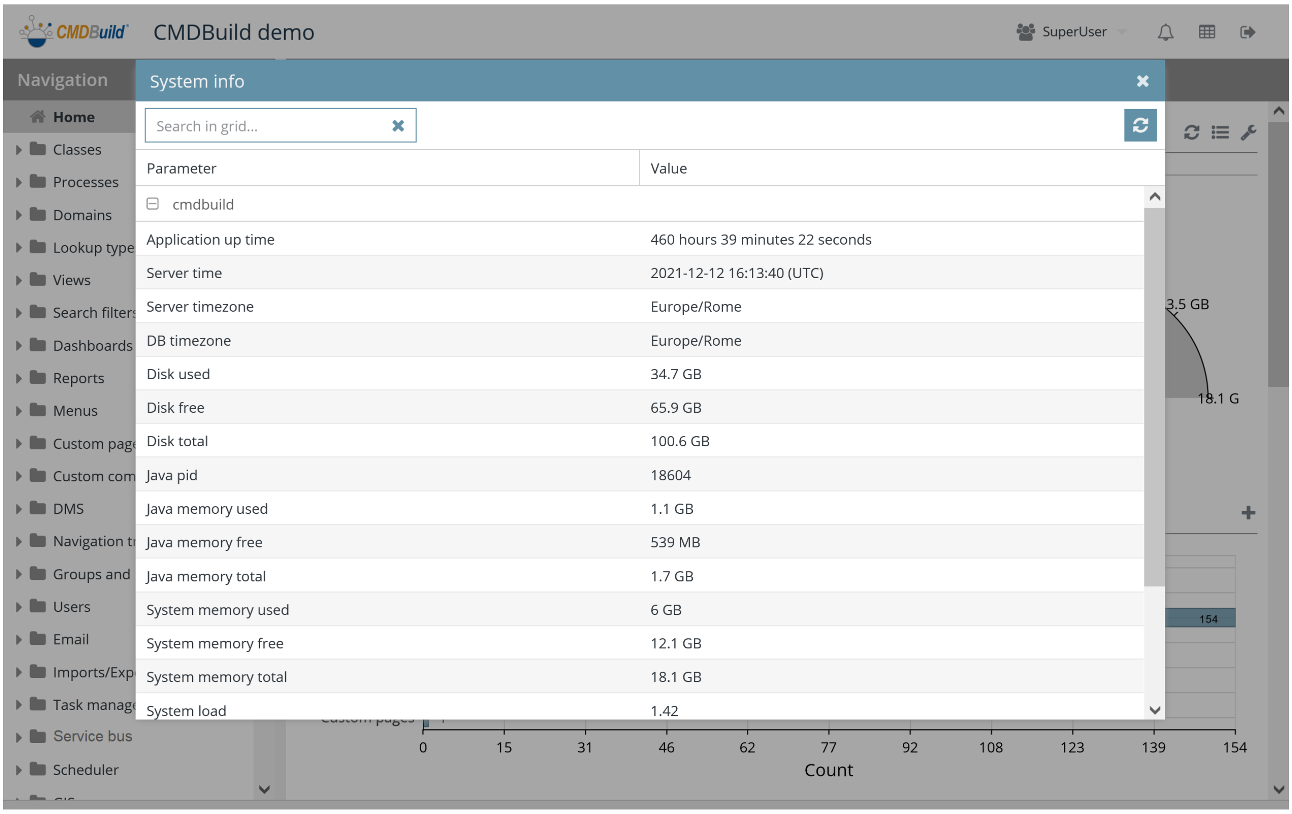
Here you can see the related screenshot



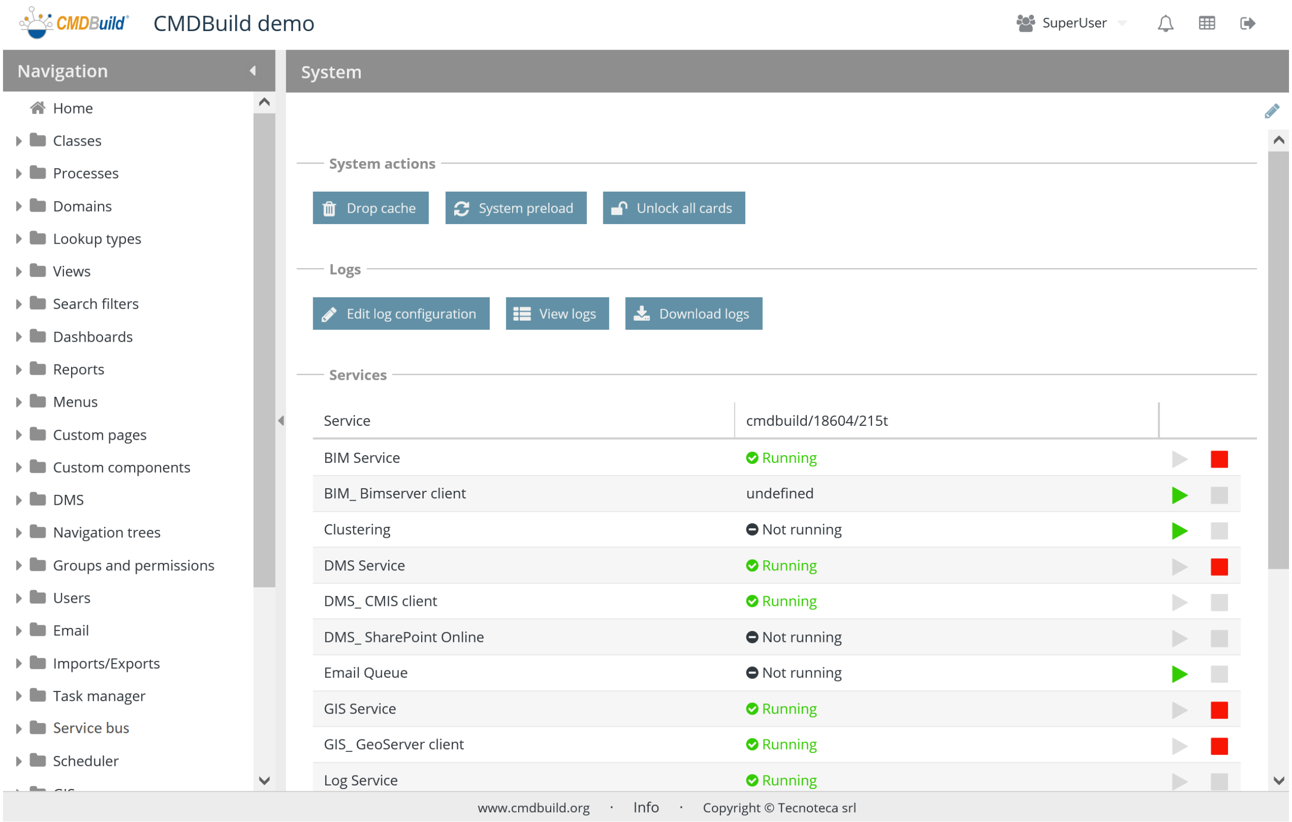
The interface represents also an access point to the correlated features.

The buttons placed on the right of the server status graphs allow to:

- update the page
- access more detailed info about the state of the system (first following screenshot)
- access the server management page and the active services one (second following screenshot)

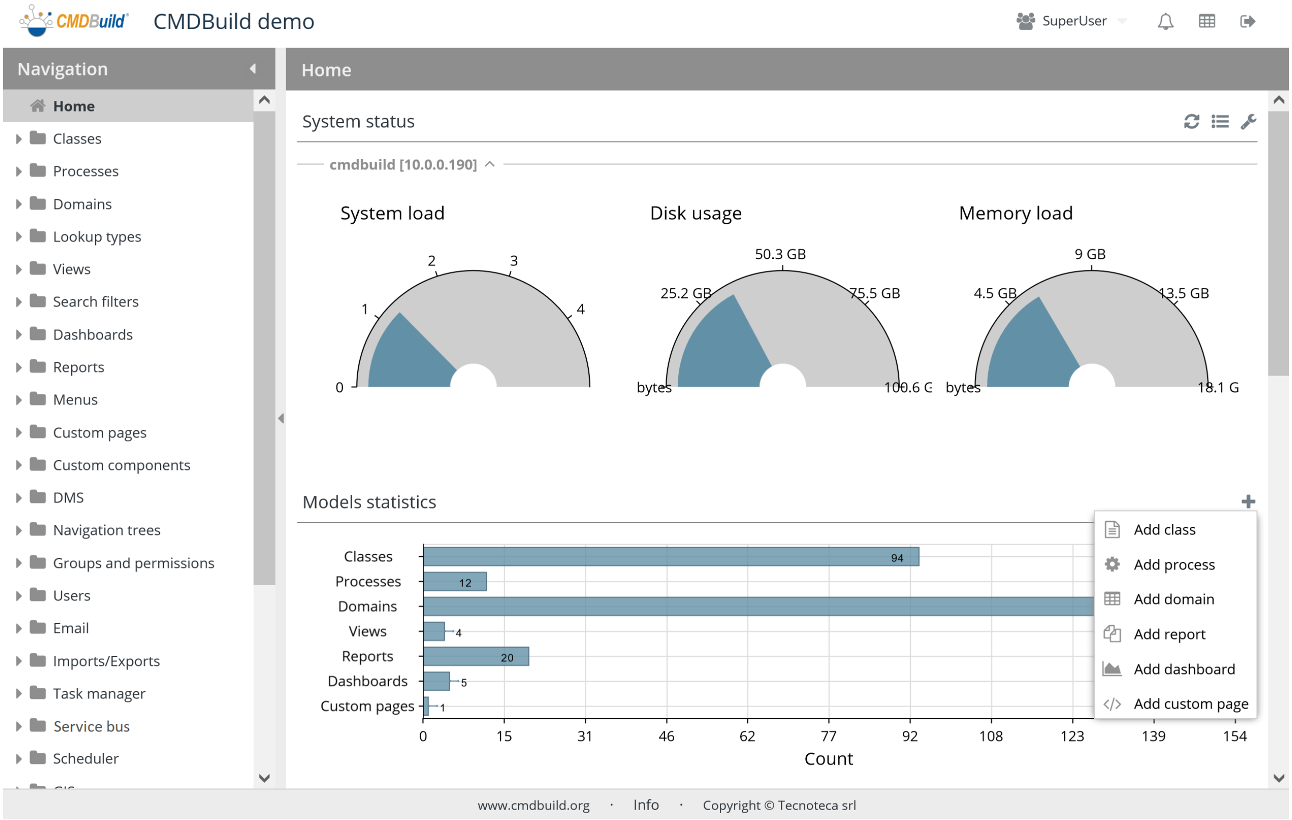


Parameter	Value
⊟ cmdbuild	
Application up time	460 hours 39 minutes 22 seconds
Server time	2021-12-12 16:13:40 (UTC)
Server timezone	Europe/Rome
DB timezone	Europe/Rome
Disk used	34.7 GB
Disk free	65.9 GB
Disk total	100.6 GB
Java pid	18604
Java memory used	1.1 GB
Java memory free	539 MB
Java memory total	1.7 GB
System memory used	6 GB
System memory free	12.1 GB
System memory total	18.1 GB
System load	1.42



Service	cmdbuild/18604/215t
BIM Service	✓ Running
BIM_ Bimserver client	undefined
Clustering	● Not running
DMS Service	✓ Running
DMS_ CMIS client	✓ Running
DMS_ SharePoint Online	● Not running
Email Queue	● Not running
GIS Service	✓ Running
GIS_ GeoServer client	✓ Running
Log Service	✓ Running

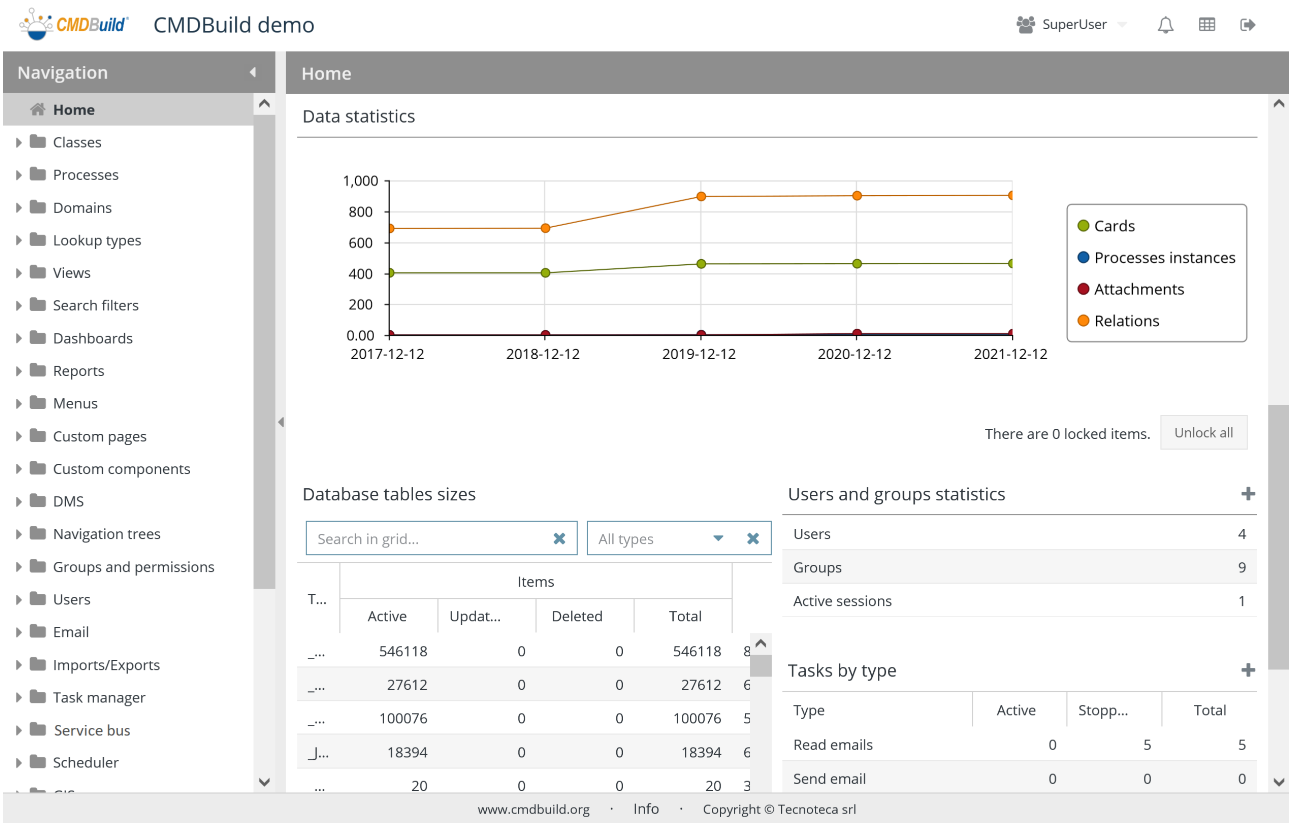
The button "+" placed to the right of the server status graphs allows to move over the creation features for each entity type.



A second information batch includes:

- volume of data present in CMDB (data cards, process instances, attached documents, relations) and the latest years growth trend
- Indications on the active lock number in the system
- List of charts present in the database, allowing research and filtering by type, detail of chart number, split in active, edited and erased, indication of the occupied space
- number of users and configured groups
- number of configured batch tasks, differentiated by type

Here you can see the related screenshot



There are 0 locked items.

	Items				
T...	Active	Updat...	Deleted	Total	
_...	546118	0	0	546118	8
_...	27612	0	0	27612	6
_...	100076	0	0	100076	5
_J...	18394	0	0	18394	6
...	20	0	0	20	3

Users	4
Groups	9
Active sessions	1

Type	Active	Stopp...	Total
Read emails	0	5	5
Send email	0	0	0

Also in this case the interface represents an access point to the involved features.

The button "Unlock all" allows to reset any lock applied to data cards and processes.

The buttons "+" placed to the right of the charts allow moving over the creation features for:

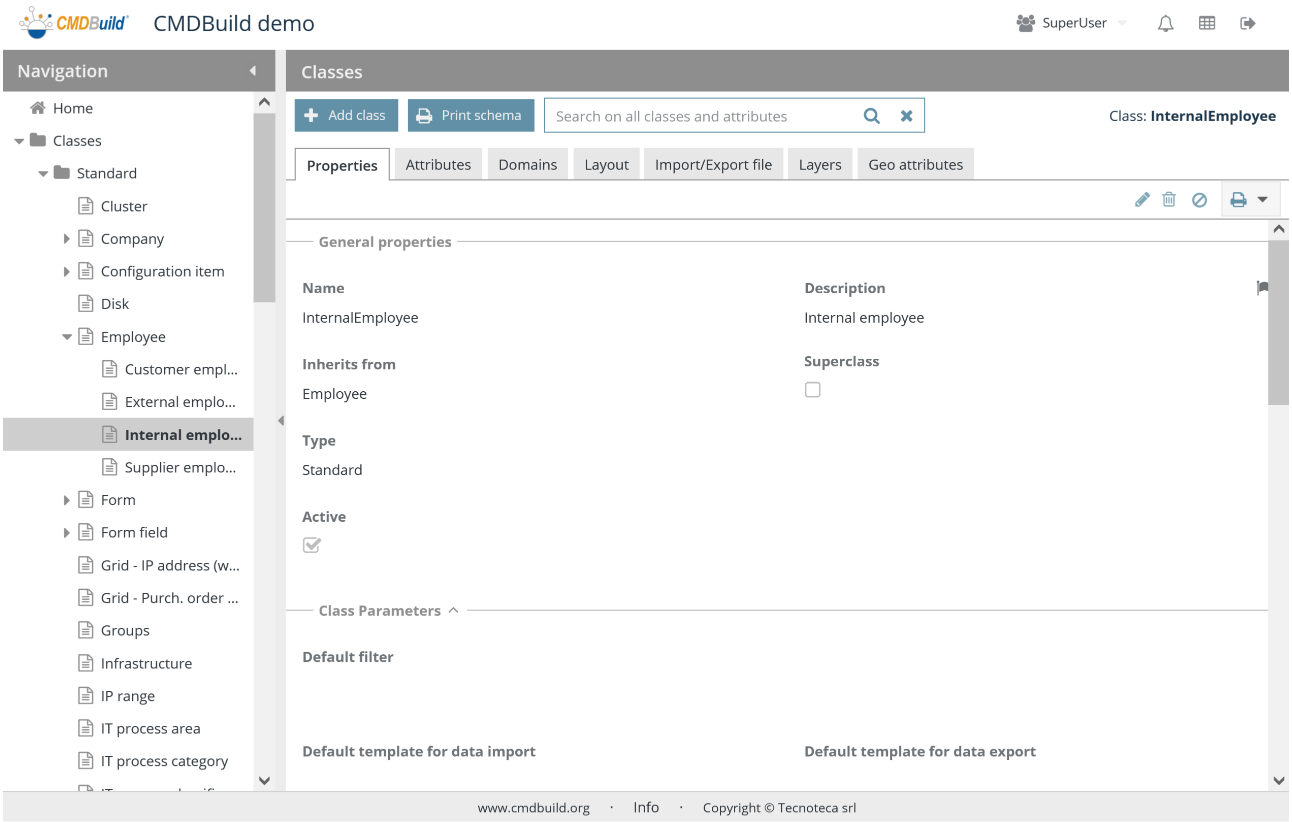
- users and groups
- task batch of different types

7. Classes

In CMDBuild, a class represents objects to be stored in the CMDB. The system creates automatically all the forms to manage data. The input fields match the attributes defined in the class, and cards data is stored in the related database table (automatically created).

7.1. Properties tab

The "Properties" tab allows you to create new classes and edit the old ones.



CMDBuild demo

SuperUser

Navigation

- Home
- Classes
  - Standard
    - Cluster
    - Company
    - Configuration item
    - Disk
    - Employee
      - Customer empl...
      - External emplo...
      - Internal emplo...
      - Supplier emplo...
    - Form
    - Form field
    - Grid - IP address (w...
    - Grid - Purch. order ...
    - Groups
    - Infrastructure
    - IP range
    - IT process area
    - IT process category

Classes

Add class | Print schema | Search on all classes and attributes

Class: **InternalEmployee**

Properties | Attributes | Domains | Layout | Import/Export file | Layers | Geo attributes

General properties

Name
InternalEmployee

Description
Internal employee

Inherits from
Employee

Superclass
□

Type
Standard

Active
☑

Class Parameters

Default filter

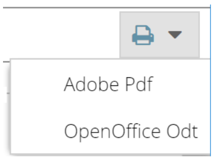
Default template for data import

Default template for data export

www.cmdbuild.org · Info · Copyright © Tecnoteca srl

There is the possibility of performing the following operations:

• add a new class	+ Add class
• print data model schema (PDF or ODT)	Print schema
• do searches on the configured classes and the related attributes	Search on all classes and attributes 🔍 ✖
• edit the selected class	✎
• delete the selected class (physical removal if not	🗑

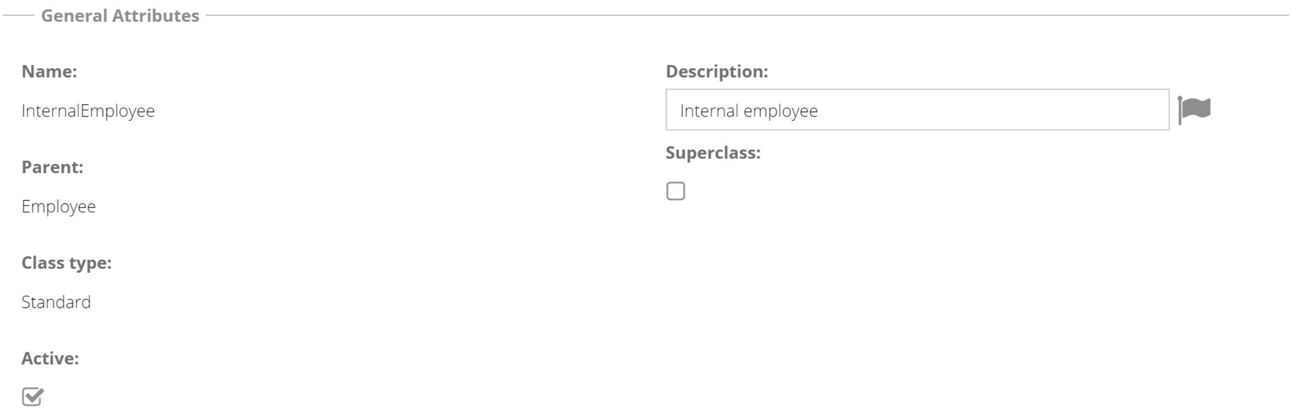
containing data, otherwise logic)	
• disable the selected class (i.e. set the status as non-active)	⊘
• print current class details	

Each class requires several parameters to be compiled (metadata), that are divided into the following selections.

7.1.1. General properties

The following information is required:

- Name: class name, that corresponds to the physical name of the table in the database
- "Description" - the class name used in the application (the flag on the right allows you to open a popup window to localize the description in the enabled languages)
- Inherit from: name of the class (superclass), by default all classes inherit from the first-level system superclass "Class")
- Superclass: it states whether it is a superclass (it can have inherited classes and therefore it is just a template not containing data) or not
- Type, which takes the following values:
  - Standard: common CMDBuild class
  - Simple: class not containing any mechanisms (inheritance, data history, attributes and reference, relations, etc.), which is optimized in order to manage huge archives (registration of data received from monitoring systems, IOT, etc.)
- Multitenant mode, which can take the following values:
  - Never: all class cards are visible to all tenants
  - Always: all class cards are visible to one tenant
  - Mix: some class cards are visible to all tenants, other ones to one
- Active: the class is active and can be used



General Attributes

Name: InternalEmployee

Description: Internal employee

Parent: Employee

Superclass: □

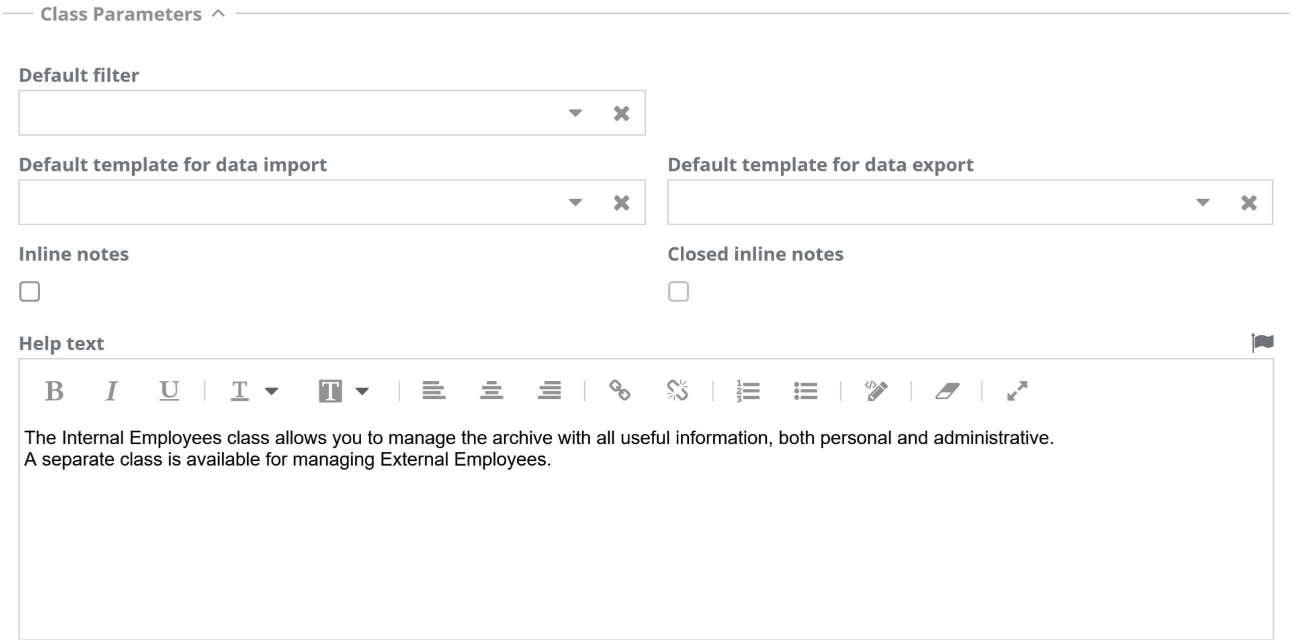
Class type: Standard

Active: ☑

7.1.2. Other class parameters

The following information is required:

- Default filter: it allows you to specify the default filter (selected among the filters defined for that class in the specific function) that has to be applied in the list of class cards
- Default template for import: it allows to specify the default template in order to execute the import of data on the current class, which must already be configured
- Default template for export: it allows to specify the default template in order to execute the export of data from the current class, which must already be configured
- Inline notes: it allows you to state whether the field Notes (usually managed in a specific TAB) has to be inserted in a specific "fieldset" (group of information) in the base card
- Closed inline notes: it is possible to state whether the "fieldset" located in the base card has to appear as closed by default
- Help text: allows to prepare the help text which will be available in the Data management module (localizable)



Default filter:

Default template for data import:

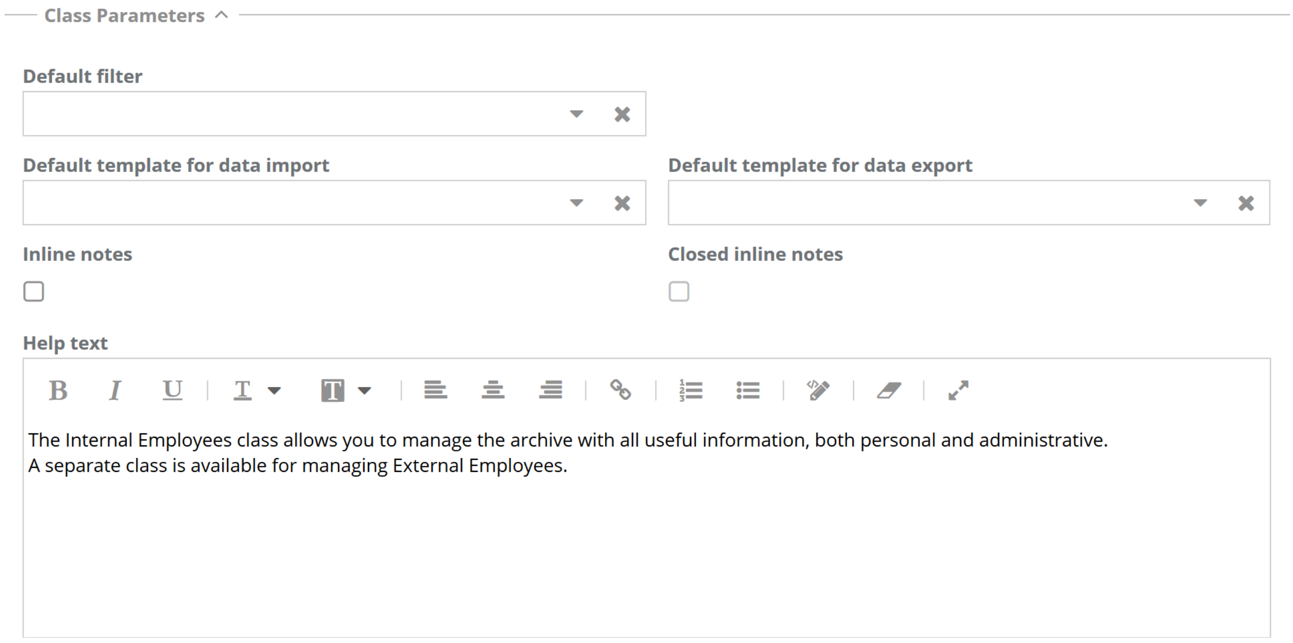
Default template for data export:

Inline notes: □

Closed inline notes: □

Help text:

The Internal Employees class allows you to manage the archive with all useful information, both personal and administrative.
A separate class is available for managing External Employees.



Class Parameters

Default filter

Default template for data import

Default template for data export

Inline notes

□

Closed inline notes

□

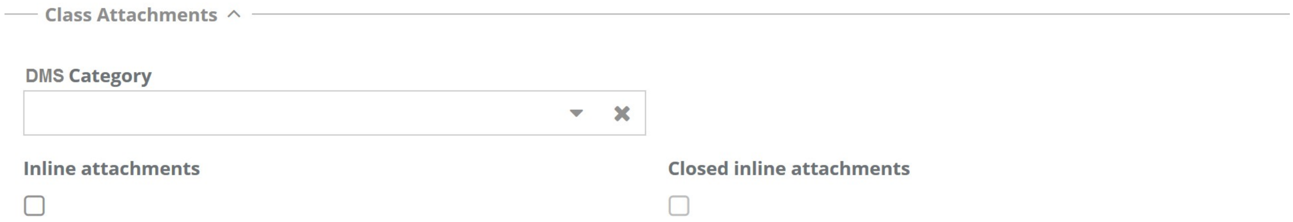
Help text

The Internal Employees class allows you to manage the archive with all useful information, both personal and administrative.
A separate class is available for managing External Employees.

7.1.3. Attachments

The following information is required:

- Document category: it allows you to specify the document category containing the entries that have to be used to classify the attachments of the current class (unless otherwise stated, the general class of the CMDBuild instance is used, which is specified in the general configurations)
- Inline attachments: it allows you to state whether the attachment list (usually managed in a specific TAB) has to be inserted in a specific "fieldset" (group of information) in the base card
- Closed inline attachments: it is possible to state whether the "fieldset" located in the base card has to appear as closed by default



Class Attachments

DMS Category

Inline attachments

□

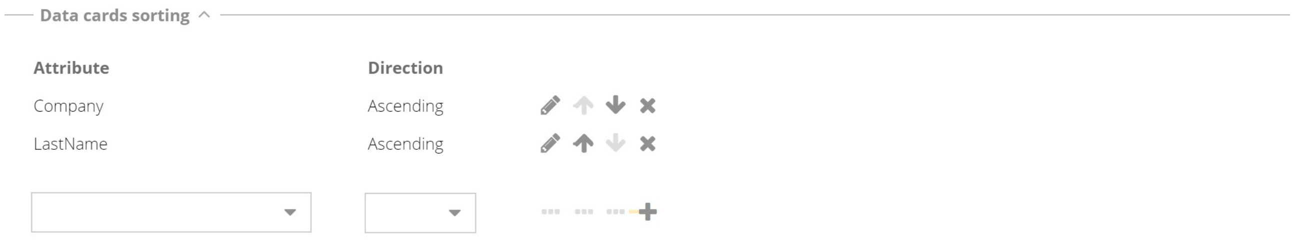
Closed inline attachments

□

7.1.4. Data organization

It is possible to organize the data cards in the Management Module specifying which attributes have to be organized, their sequence and the type of organization.

If not otherwise specified, the default organization is on the attribute "Description" (ascending).



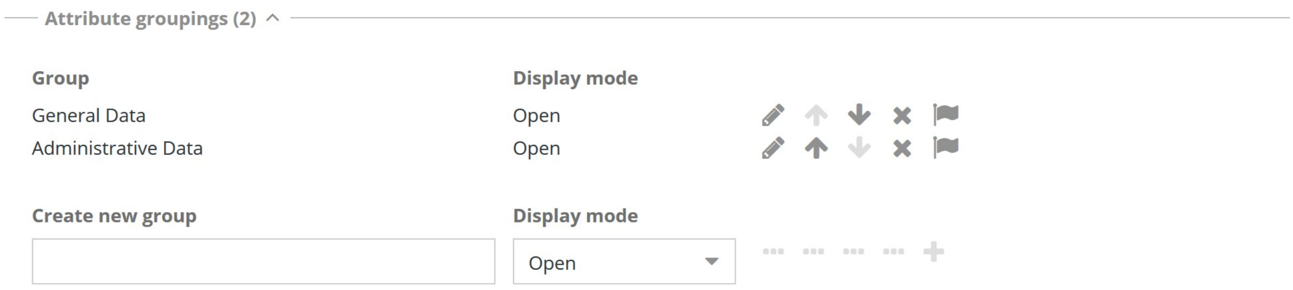
Data cards sorting

Attribute	Direction
Company	Ascending
LastName	Ascending

7.1.5. Attribute grouping

It is possible to list and localize the attribute groupings (fieldsets) that have to be used for the current class.

For each group, it is possible to state whether it must appear open or closed by default.



Attribute groupings (2)

Group	Display mode
General Data	Open
Administrative Data	Open

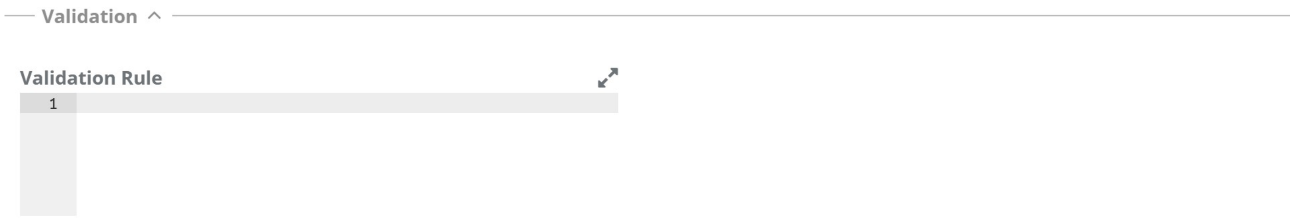
Create new group

Display mode: Open

7.1.6. Validation rules

It is possible to write Javascript code in order to verify that the data card compiled in the Management Module satisfies the desired criteria (with the "expander" button.

In case the validation is not satisfied, the button for the card saving is not enabled and the card remains in edit mode.



Validation

Validation Rule

7.1.7. Form trigger

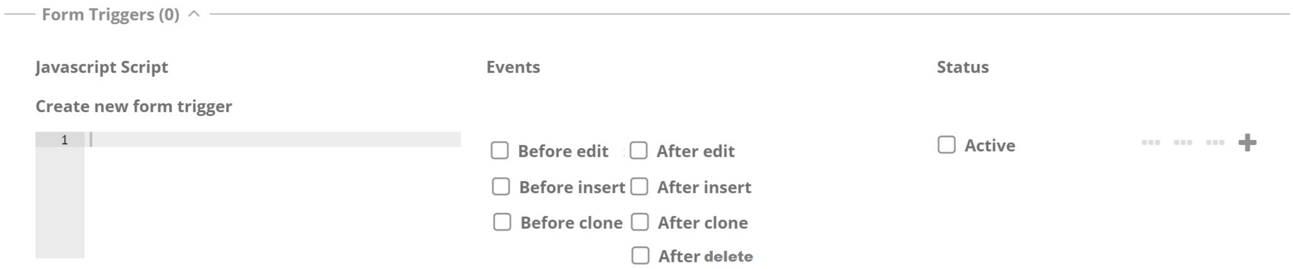
The form trigger mechanism allows you to define the operations that have to be executed in the Management Module on data of the CMDB, before or after that a user carries out an operation like inserting, editing, cloning or deleting.

It is possible to define several triggers in the same class.

For each form trigger you have to define:

- the Javascript
- the type of event that script has to activate: before an insertion, after an insertion, before a change, after a change, before a cloning, after a cloning, after a cancellation
- the trigger status (active, not active)

It is possible to organize the form triggers in the desired mode.



Form Triggers (0)

Javascript Script	Events	Status
Create new form trigger		
1	□ Before edit □ After edit □ Before insert □ After insert □ Before clone □ After clone □ After delete	□ Active

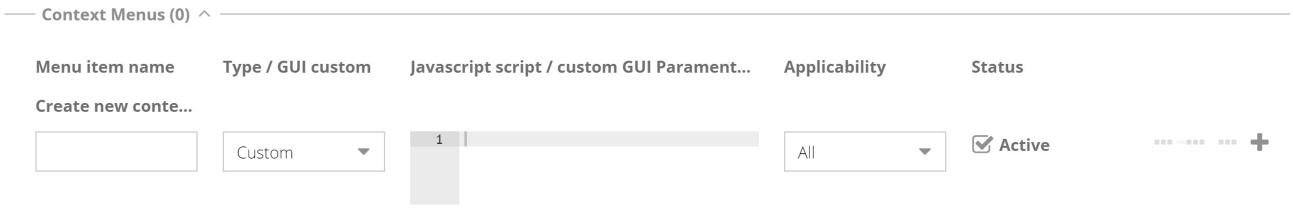
7.1.8. Contextual menu

The purpose of the contextual menu is making available additional features, which are related to the class where the user is working and useful to work faster.

For each entry of the contextual menu you have to define:

- the name of the menu entry
- the type of the menu entry:
  - Component: reuse of a component already available in a system library
  - Customized: custom code
  - Separator: separator line between two menu sections
- the component to reuse, in case the type of Component has been chosen
- the implementation of the menu entry:
  - Javascript, for custom implementations
  - sequence of parameters launching a previous component
- type of applicability: only to the current row, to the selected rows, to all rows of the class
- the status of the menu entry (active, not active)

It is possible to organize the entries of the contextual menu in the desired mode.



Context Menus (0)

Menu item name	Type / GUI custom	Javascript script / custom GUI Parament...	Applicability	Status
Create new conte...				
	Custom	1	All	☑ Active

7.1.9. Form widget

This feature allows you to configure some features that will be available as buttons in the Management Module (and later as further fieldsets of the main card).

It is possible to choose among the CMDBuild standard widgets or create custom ones.

The available widgets are as follows:

- Calendar: it shows the specified deadlines on a calendar
- Create / modify card: it allows to insert / edit a data card into a class that is different to the current one
- Create report: it allows to print a report
- Start workflow: it allows to start the specified workflow by using a popup window (the workflow will then go ahead with the common CMDBuild functions)

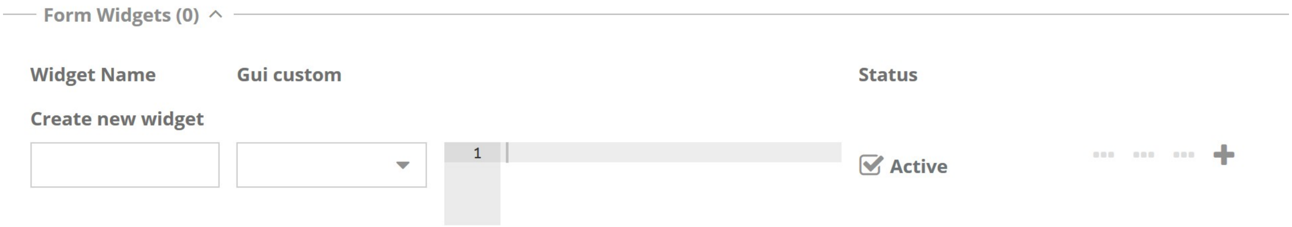
In case the custom widgets are configured in the system, they will appear in the selection list.

Further standard widgets can be used just within the workflows (see the Workflow Manual).

For each widget you have to define:

- the widget name
- the type of widget, to be chosen in a list including both standard and custom ones
- the sequence of activation parameters
- the status of widgets (active, not active)

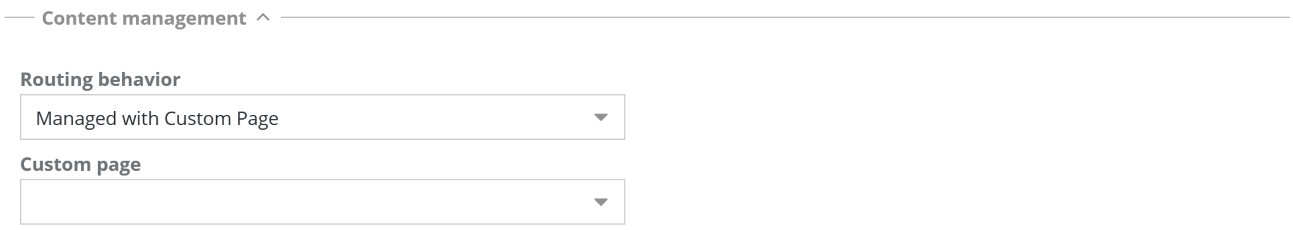
It is possible to organize the widgets in the desired mode.



7.1.10. Content management

This feature allows you to set the navigation mode for the current class:

- standard: standard page generated by the system
- managed with custom page:: custom page selected among the available ones
- managed with views: view selected among the available ones
- managed with custom addressing: destination page selected among the available ones for the current class



Content management

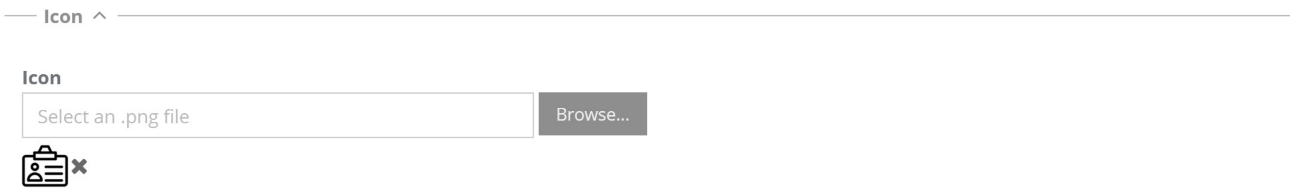
Routing behavior: Managed with Custom Page

Custom page:

7.1.11. Icon

The function allows you to import an icon that has to be combined to the class in PNG format or to delete the current icon.

The icon is also used to represent the data cards of that class in the Relation Graph.



Icon

Icon: Select an .png file

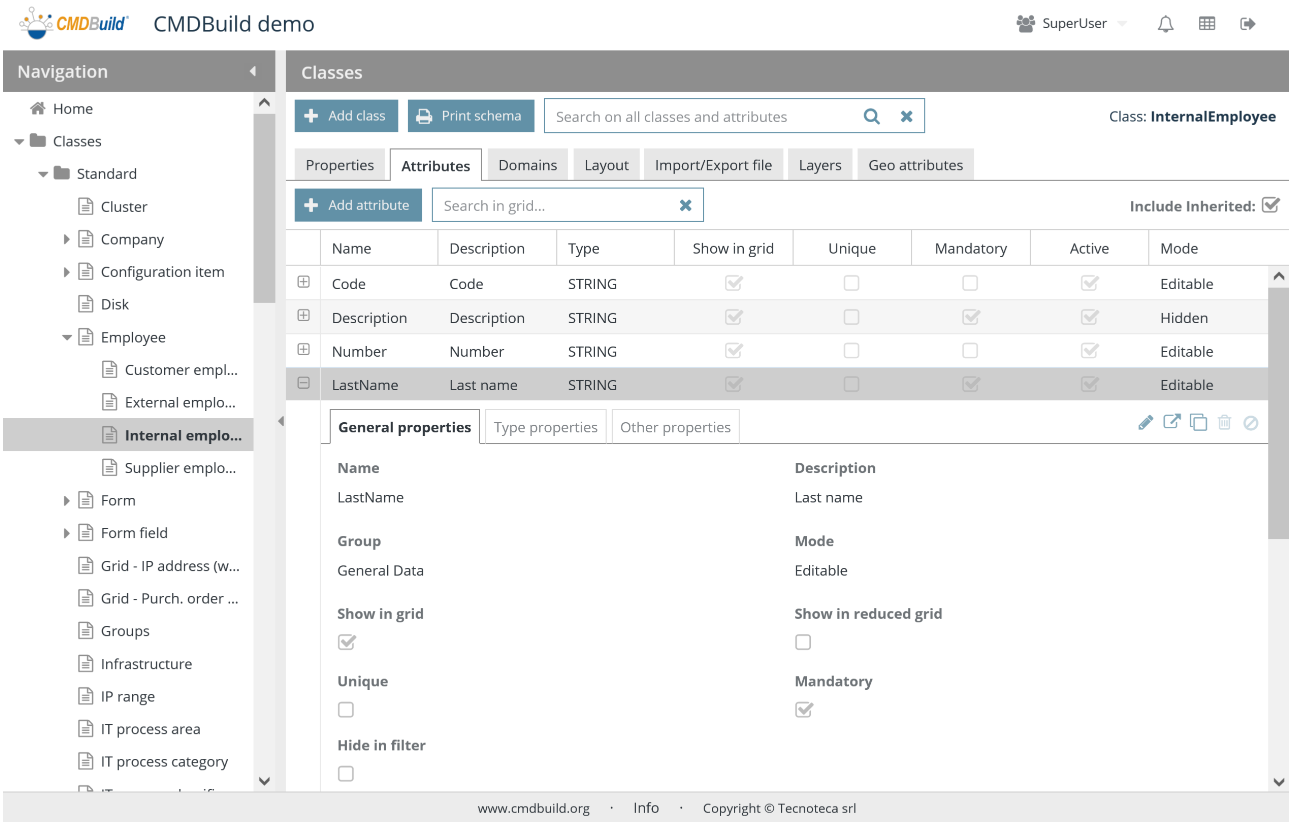
7.2. Attributes tab

The Attributes tab allows you to manage the class attributes.

This tab is used to configure information to a new class, or update an existing class.

The attributes of a class, corresponding to the columns of the related database table, can be:

- inherited from a superclass (from the first-level superclass called “Class” and containing the attributes “Code”, “Description” and “Notes”, in common with all classes of data model, or from an intermediate-level superclass defined by the user)
- specific of the class in question



CMDBuild demo

Class: **InternalEmployee**

Include Inherited: ☑

	Name	Description	Type	Show in grid	Unique	Mandatory	Active	Mode
⊞	Code	Code	STRING	☑	□	□	☑	Editable
⊞	Description	Description	STRING	☑	□	☑	☑	Hidden
⊞	Number	Number	STRING	☑	□	□	☑	Editable
⊟	LastName	Last name	STRING	☑	□	☑	☑	Editable

Name: LastName

Description: Last name

Group: General Data

Mode: Editable

Show in grid: ☑

Show in reduced grid: □

Unique: □

Mandatory: ☑

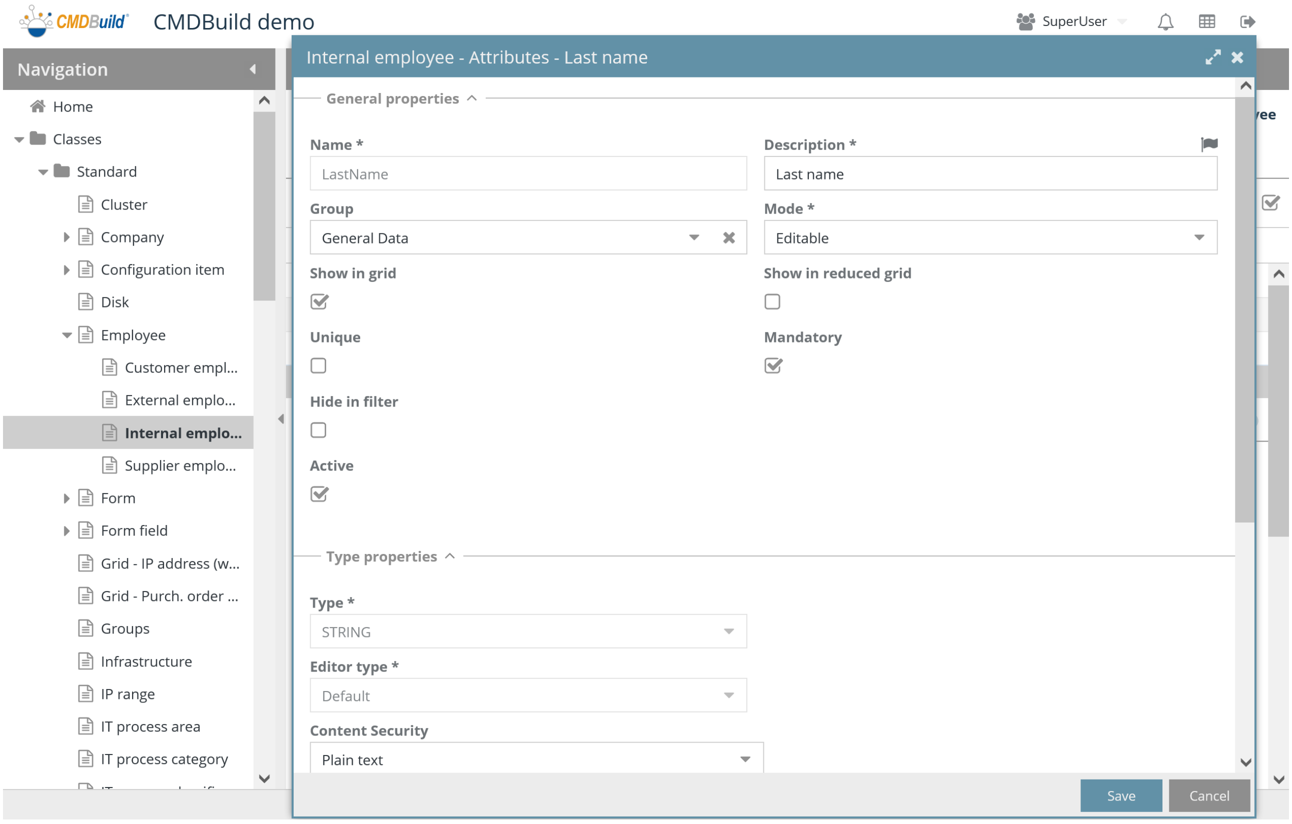
Hide in filter: □

There is the possibility of performing the following operations:

• create a new attribute	✚ Add attribute
• search for the attribute in the class	Search in grid... ✖
• show all attributes or just the not inherited ones	Include Inherited: ☑
• expand the current row and display its data on TABs	⊞
• edit the selected attribute (edit available information)	✎

• display the complete information related to the selected attribute	↗
• clone the current attribute	⧉
• delete the selected attribute (if not inherited and not containing data)	🗑
• disable the selected attribute (i.e. set the status as non-active)	⊘

Each attribute requires several parameters to be compiled (metadata), that are divided into the two following selections.



Internal employee - Attributes - Last name

General properties

Name *: LastName

Description *: Last name

Group: General Data

Mode *: Editable

Show in grid: ☑

Show in reduced grid: □

Unique: □

Mandatory: ☑

Hide in filter: □

Active: ☑

Type properties

Type *: STRING

Editor type *: Default

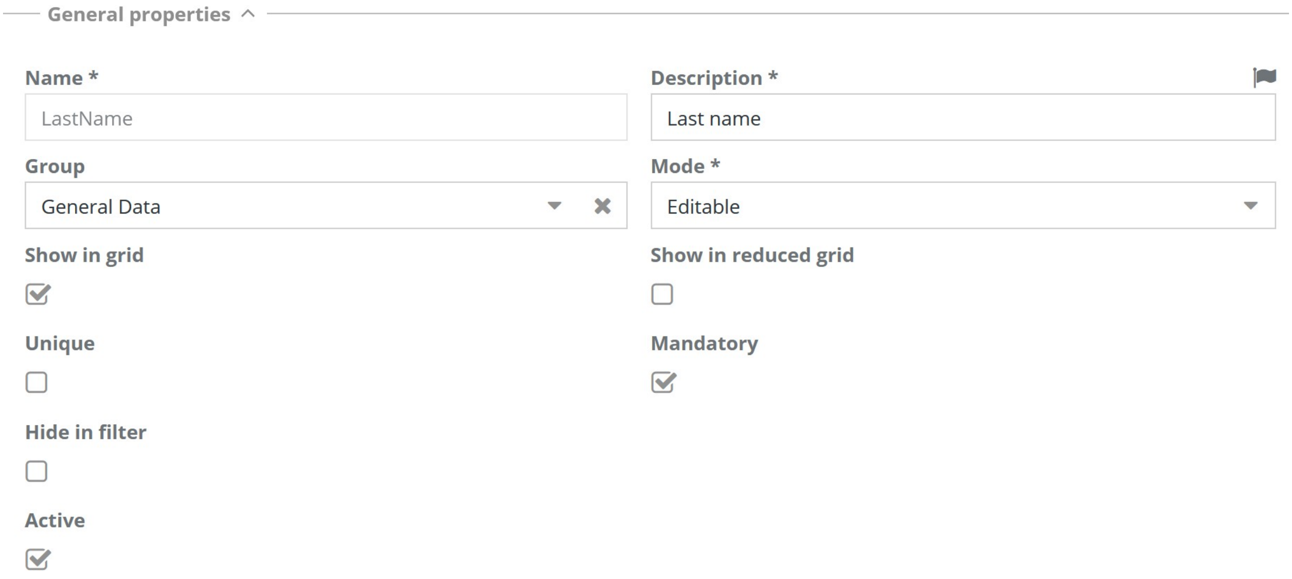
Content Security: Plain text

7.2.1. General properties

The following information is required:

- Name: attribute name, that corresponds to the physical name of the database column
- "Description": used to report the attribute name in the application
- Group: used to group together attributes in the Management Module (through tabs when expanding the grid row, through fieldsets when editing or displaying)
- Editing mode, that can be:
  - Editable: normal editable attribute
  - Read Only: visible but not editable

    - Hidden: attribute not visible
    - Unchanging: this attribute can be set once and not be edited any more
- Display in grid: it states if the attribute has to be displayed in the grid display in the Management Module
- Display in reduced grid: it states if the attribute has to be displayed in the reduced grid in the Management Module (GIS, BIM) or in the mobile APP
- Unique: it states that the attribute represents a unique key
- Mandatory: it states that the attribute has to be present
- Hide in filter: the attribute does not appear among the ones that can be used to set filters
- Active: the attribute is active and can be used



General properties

Name *	Description *
LastName	Last name
Group	Mode *
General Data	Editable
Show in grid	Show in reduced grid
☑	□
Unique	Mandatory
□	☑
Hide in filter	
□	
Active	
☑	

7.2.2. Property of the attribute type

Some always-available information is required, such as other differentiated according to the type of attribute.

The following information is always available:

- Type of attribute, which can be:
    - Biginteger: integer (8 byte)
    - Boolean: with values Yes / No
    - Char: Single Character
    - Date
    - Decimal
    - Double: double precision floating point
    - File: to upload an attachment as form field
    - Formula: to set a formula (SQL or Groovy) for the dynamic calculation

- Foreign key: link to another class, only for "Simple" classes type
- Integer: integer (4 byte)
- IP address: IP V4 or IP V6 address, with possibility of specifying the mask value
- Link: mention to addressable resources through URL
- Lookup: list of closed values (which can be configured in the Lookup management)
- LookupArray: lookup with multiple selection
- Reference: link to another class through a domain (type of relation), available only for "Normal" classes
- String
- Text
- Time
- TimeStamp

Here's the additional information, differentiated according to the type of attribute.

Biginteger-type attributes

It is similar to the integer-type but with extended capacity, as from specifications of the PostgreSQL database (8 bytes instead of 4 bytes).

Boolean attributes

It is possible to state if it must be initialised by default as "false"

Char attributes

There is no additional information.

Date attributes

With a proper icon, it is possible to link the date to a generation rule for deadlines configured with the specific function.

Decimal attributes

For Decimal attributes these additional information are required:

- Precision: indicates the total number of digits
- Scale: indicates the number of digits in the decimal part
- Unit of measurement (free text, for example Kg, Km, Num, etc.)
- Location of unit of measurement (before the number and after the number)
- Show separator

Double attributes

For Double attributes further information is required:

- Unit of measurement (free text, for example Kg, Km, Num, etc.)

- Location of unit of measurement (before the number and after the number)
- Show separator of thousands
- Showing digits number (if there are more the number gets rounded)

File attributes

You can manage an attachment, which has been associated to a DMS category and filed into the DMS system, as a field of the card. In this way you can individually manage its presence, obligation, etc.

For File attributes this additional information is required:

- DMS Category, it can be selected among those defined for the current class
- It shows the preview on the card

Formula-type attributes

These are fictitious attributes, not filed into the database but dynamically recalculated through a SQL function (useful for processes inside the CMDB) or a Groovy script (useful to interact with external systems via webservice).

For Formula attributes this additional information is required:

- Formula (SQL function or Groovy script)
- A function among the available ones, in case of SQL function
- A script among the available ones, in case of Groovy script

Integer attributes

For Integer this additional information is required:

- Unit of measurement (free text, for example Kg, Km, Num, etc.)
- Location of unit of measurement (before the number and after the number)
- Show separator of thousands

IP address attributes

For IP address attributes the following information is required:

- IP type: it states the type IPV4 or IPV6 or mix

Link attributes

It allows you to address an URL resource through a card field

For Link attributes this additional information is required:

- Show label
- Mandatory label

Lookup attributes

Lookup attributes allow you to select the value of the attribute from a predefined list of values; this list is created and modified using Lookup management function in the Administration Module (see specific chapter).

For Lookup attributes this additional information is required:

- Lookup list: it allows you to choose the value list that has to be linked to the configured attribute
- Filter: it allows you to exclude some values from the list according to the current context and can be defined through CQL (CMDBuild Query Language)
- Metadata: through the button "Add metadata" you can access a popup window to set support variables defined in the same CQL language and usable while writing the filter
- Pre-select if unique: it forces the development of the lookup attribute, in case it includes a unique value

LookupArray attributes

LookupArray attributes are similar to the Lookup ones, except for the multiple selection of values.

Unlike the Lookup attributes, they can not be structured on more levels.

For LookupArray attributes this additional information is required:

- Lookup list: it allows you to choose the value list that has to be linked to the configured attribute
- Filter: it allows you to exclude some values from the list according to the current context and can be defined through CQL (CMDBuild Query Language)
- Metadata: through the button "Add metadata" you can access a popup window to set support variables defined in the same CQL language and usable while writing the filter

Reference attributes

The Reference attributes allow you to link a data card to another "parent" card through a domain with cardinality 1:N (where the "N" side is the current class).

In the Management Module it will be possible to enhance a Reference attribute by selecting the parent card from the list of available references (which contains the attribute “Description”).

A reference attribute, in the Management Module, creates or modifies the relation between two cards.

Even if differently configured, a Reference attribute is similar to a Foreign Key column in a relational database table.

For Reference attributes this additional information is required:

- Domain: it allows you to choose the type of link between the current class and the final class (with cardinality 1:N where the "N" side is the current class); each Reference attribute has to use a different domain
- Filter: it allows you to exclude some values from the list of the description of cards linked according to the domain and can be defined through CQL (CMDBuild Query Language)

- Metadata: through the button "Add metadata" you can access a popup window to set support variables defined in the same CQL language and usable while writing the filter
- Preselect if unique

String attributes

For String attributes this additional information is required:

- Maximum length: maximum amount of characters allowed
- Content safety: selection of the type of admitted characters, with the following possibilities:
  - Plain text
  - Secure HTML (some TAGs are excluded)
  - Complete HTML

Text attributes

For Text attributes this additional information is required:

- Editor-type, which can be:
  - Plain text
  - HTML editor: formatted text with a WYSIWYG editor
- Content safety: selection of the type of admitted characters, with the following possibilities:
  - Plain text
  - Secure HTML (some TAGs are excluded)
  - Complete HTML

Time attributes

For Time attributes this additional information is required:

- Show seconds

Timestamp attributes

For Timestamp attributes this additional information is required:

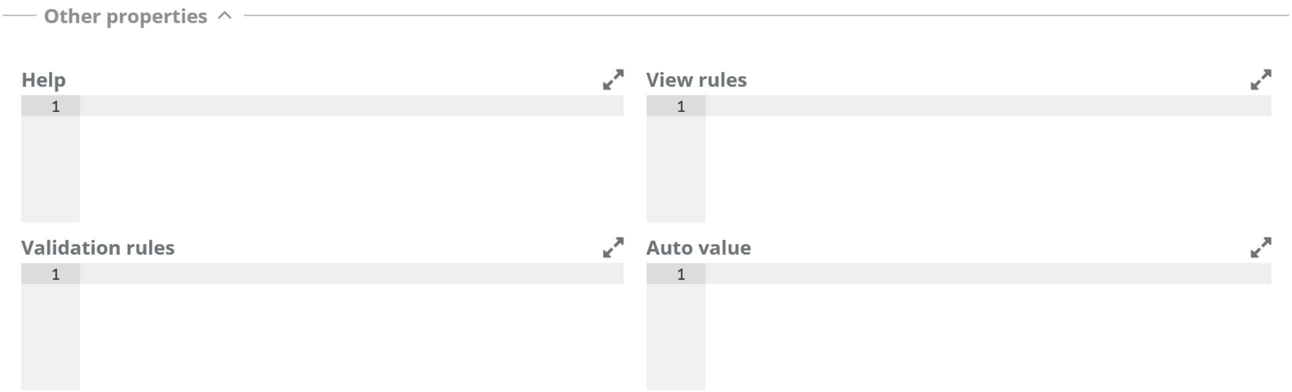
- Show seconds

7.2.3. Other properties

The information related to the "expander" button is required:

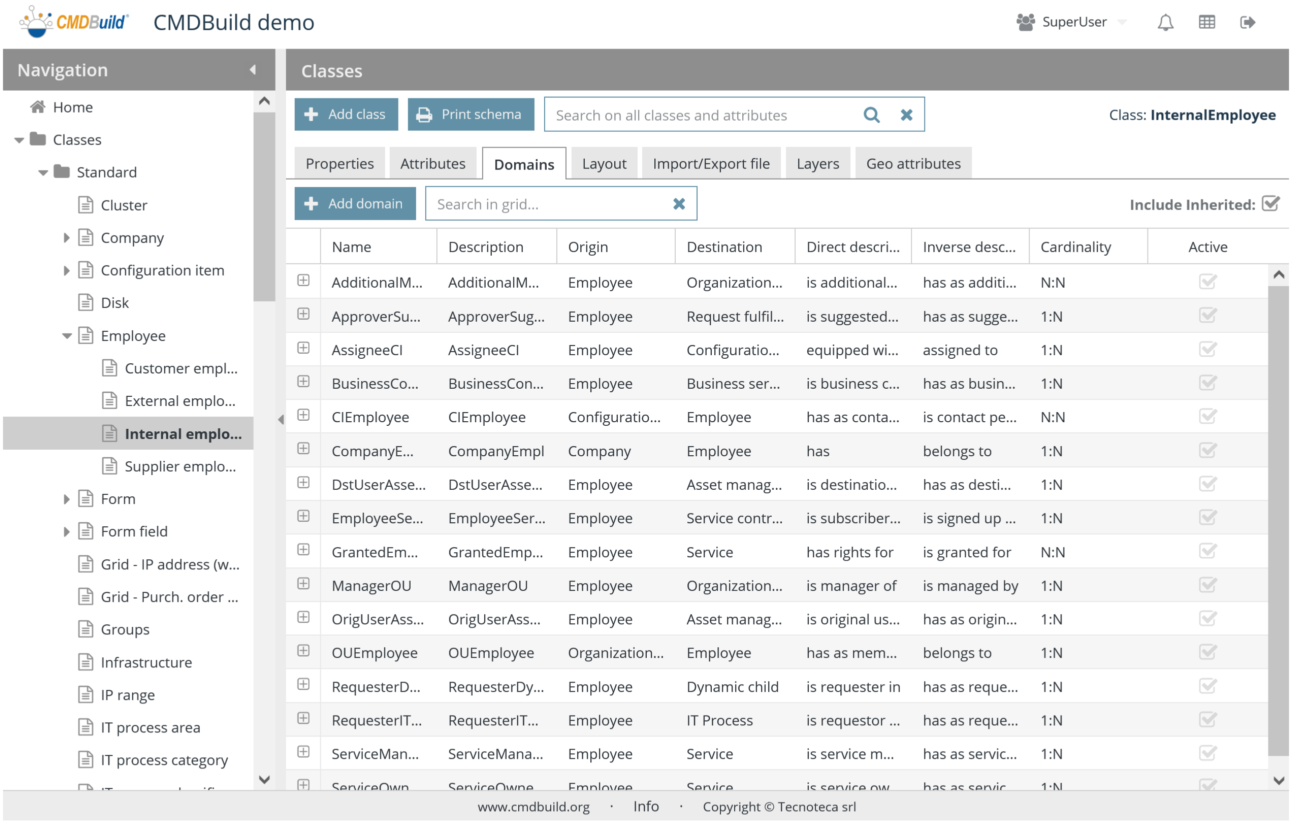
- Help: help text for the operator, with markdown syntax
- View rules: you can make the attribute hidden or not editable in the data Management Module, according to the value of other attributes in the form (through javascript code)
- Validation rules: you can state whether the value inserted into the data Management Module is valid or not (through javascript code)

- Auto value: you can set the field value in the data Management Module according to other attributes of the form (through javascript code)



7.3. Domains tab

The Domains tab allows you to manage the various types of relations (domains) defined for the current class (see also the Domain function at the paragraph 9.1.1).



CMDBuild demo

SuperUser

Navigation

- Home
- Classes
  - Standard
    - Cluster
    - Company
    - Configuration item
    - Disk
    - Employee
      - Customer empl...
      - External emplo...
      - **Internal emplo...**
      - Supplier emplo...
    - Form
    - Form field
    - Grid - IP address (w...
    - Grid - Purch. order ...
    - Groups
    - Infrastructure
    - IP range
    - IT process area
    - IT process category

Classes

Add class | Print schema | Search on all classes and attributes

Class: **InternalEmployee**

Properties | Attributes | **Domains** | Layout | Import/Export file | Layers | Geo attributes

Add domain | Search in grid...

Include Inherited: ☑

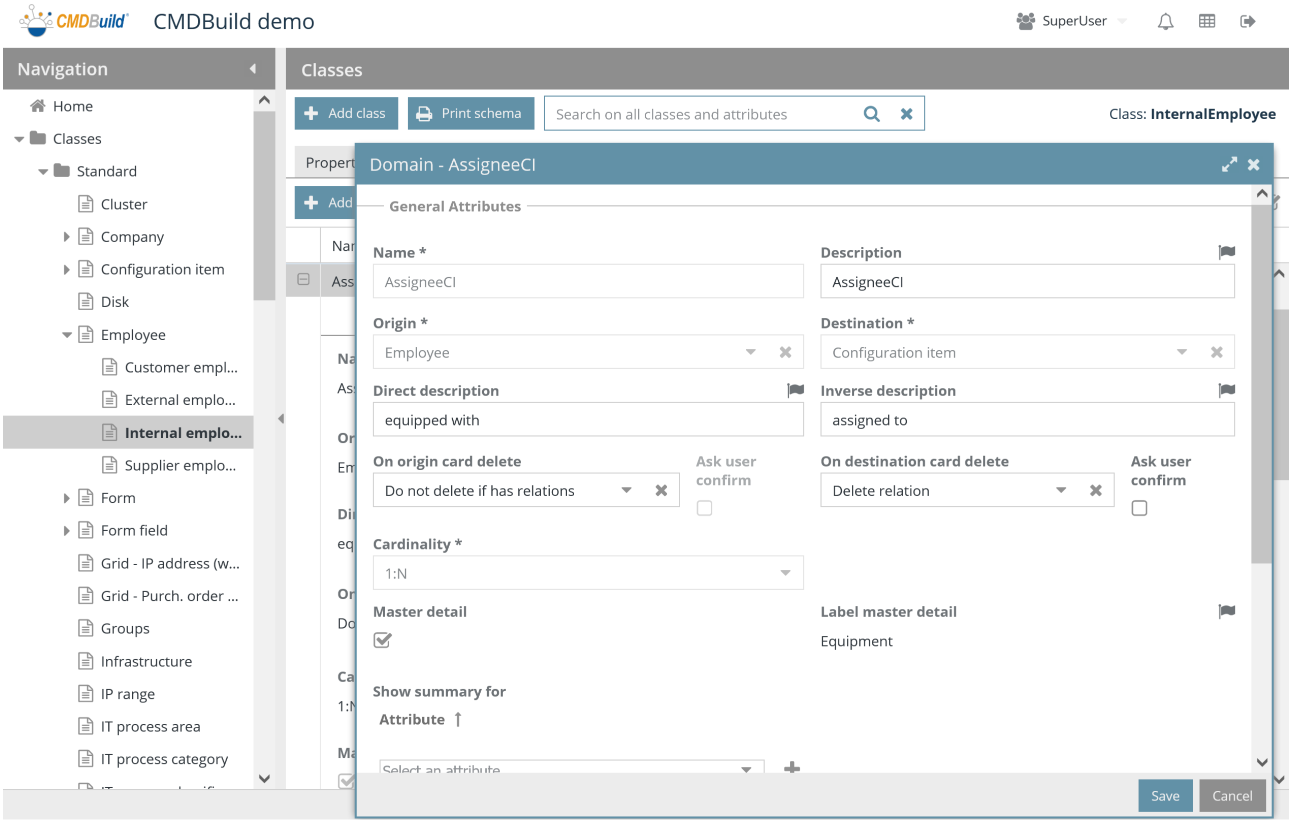
	Name	Description	Origin	Destination	Direct descri...	Inverse desc...	Cardinality	Active
⊞	AdditionalM...	AdditionalM...	Employee	Organization...	is additional...	has as additi...	N:N	✓
⊞	ApproverSu...	ApproverSug...	Employee	Request fulfil...	is suggested...	has as sugge...	1:N	✓
⊞	AssigneeCI	AssigneeCI	Employee	Configuratio...	equipped wi...	assigned to	1:N	✓
⊞	BusinessCo...	BusinessCon...	Employee	Business ser...	is business c...	has as busin...	1:N	✓
⊞	CIEmployee	CIEmployee	Configuratio...	Employee	has as conta...	is contact pe...	N:N	✓
⊞	CompanyE...	CompanyEmpl	Company	Employee	has	belongs to	1:N	✓
⊞	DstUserAsse...	DstUserAsse...	Employee	Asset manag...	is destinatio...	has as desti...	1:N	✓
⊞	EmployeeSe...	EmployeeSer...	Employee	Service contr...	is subscriber...	is signed up ...	1:N	✓
⊞	GrantedEm...	GrantedEmp...	Employee	Service	has rights for	is granted for	N:N	✓
⊞	ManagerOU	ManagerOU	Employee	Organization...	is manager of	is managed by	1:N	✓
⊞	OrigUserAss...	OrigUserAss...	Employee	Asset manag...	is original us...	has as origin...	1:N	✓
⊞	OUEmployee	OUEmployee	Organization...	Employee	has as mem...	belongs to	1:N	✓
⊞	RequesterD...	RequesterDy...	Employee	Dynamic child	is requester in	has as reque...	1:N	✓
⊞	RequesterIT...	RequesterIT...	Employee	IT Process	is requestor ...	has as reque...	1:N	✓
⊞	ServiceMan...	ServiceMana...	Employee	Service	is service m...	has as servic...	1:N	✓
⊞	ServiceOwn...	ServiceOwne...	Employee	Service	is service ow...	has as servic...	1:N	✓

www.cmdbuild.org · Info · Copyright © Tecnoteca srl

There is the possibility of performing the following operations:

• add a new domain	➕ Add domain
• search for the domain in the class	Search in grid... ✖
• expand the current row and display its data	⊞
• edit the selected domain (edit available information)	✎
• display the complete information related to the selected domain	↗
• delete the selected domain (if not inherited and not containing data)	🗑
• disable the selected domain (i.e. set the status as non-active)	⊘
• move in the complete management of the selected domain	➔

Each domain requires the parameters (metadata) below to be compiled.



Domain - AssigneeCI

General Attributes

Name *: AssigneeCI

Description: AssigneeCI

Origin *: Employee

Destination *: Configuration item

Direct description: equipped with

Inverse description: assigned to

On origin card delete: Do not delete if has relations

Ask user confirm: □

On destination card delete: Delete relation

Ask user confirm: □

Cardinality *: 1:N

Master detail: ☑

Label master detail: Equipment

Show summary for

Attribute

Select an attribute

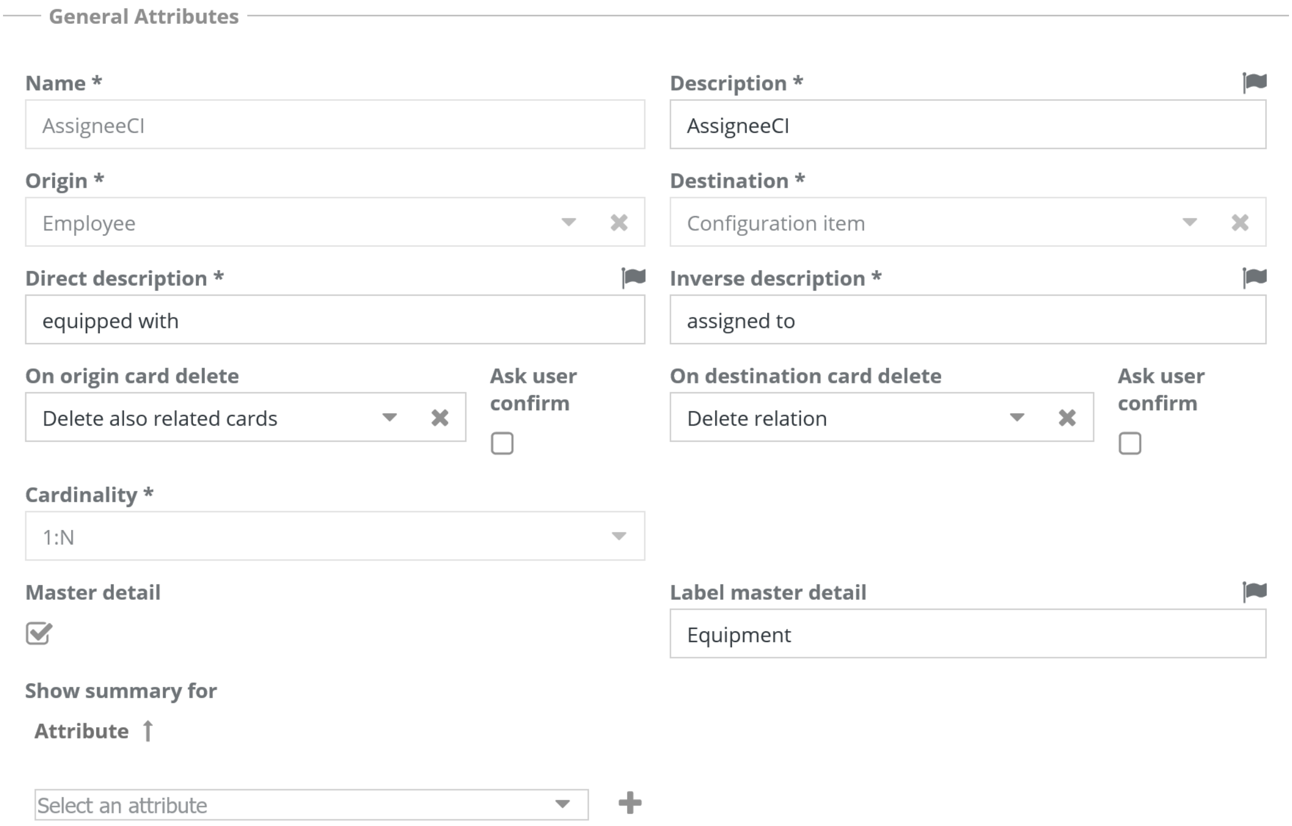
7.3.1. General properties

The following information is required:

- Name: domain class, that corresponds to the physical name of the table in the database
- Description: domain description, that has no real employment in the application and does not appear in the Management Module
- Origin: starting class of the relation (e.g. "Supplier")
- Destination: arrival class of the relation (e.g. "Asset")
- Direct description: relation characterization from the starting class to the arrival class (e.g. "has provided" the assets)
- Inverse description: relation characterization from the arrival class to the starting class (e.g. "provided by the supplier")
- In case of deletion of the original data card: don't delete anything else, delete only the relation, delete also the destination data card of the relation
- In case of deletion, ask confirmation to the operator: yes or no
- In case of deletion of the destination data card: don't delete anything else, delete only the relation, delete also the original data card of the relation ("cascade")
- In case of deletion, ask confirmation to the operator: yes or no
- Cardinality: a constraint on the number of times an entity can appear in a relation (1:1,1:N,

N:1, N:N), used by the application to insert- new data

- Master Detail: it indicates whether managing the cards of the class linked on the "N" side to the Master Detail tab of the current class; in this case, in the Master Detail tab the list of "children" cards is displayed with the functions of inserting, editing and deleting
- Master Detail label: it allows you to specify the label of the button that manages the "children" cards for the current domain in the Master Detail tab
- Shows total for the selected attributes for the Detail class (only by number)
- Inline origin: states if, in the Management Module, the card list related to the original class have to be shown in the primary TAB instead of in the Master Detail TAB
- Closed default: in case you chose to view the origin cards in the inline mode, you can request the related fieldset to be shown as closed when you open the card
- Inline destination: states if, in the Management Module, the card list related to the destination class have to be shown in the primary TAB instead of in the Master Detail TAB
- Closed default: in case you chose to view the destination cards in inline mode, you can request the related fieldset to be shown as closed when you open the card
- Display condition: it allows you to specify a CQL filter that limits the number of displayed "children" cards
- Active: the domain is active and can be used



General Attributes

Name *: AssigneeCI

Description *: AssigneeCI

Origin *: Employee

Destination *: Configuration item

Direct description *: equipped with

Inverse description *: assigned to

On origin card delete: Delete also related cards

Ask user confirm: □

On destination card delete: Delete relation

Ask user confirm: □

Cardinality *: 1:N

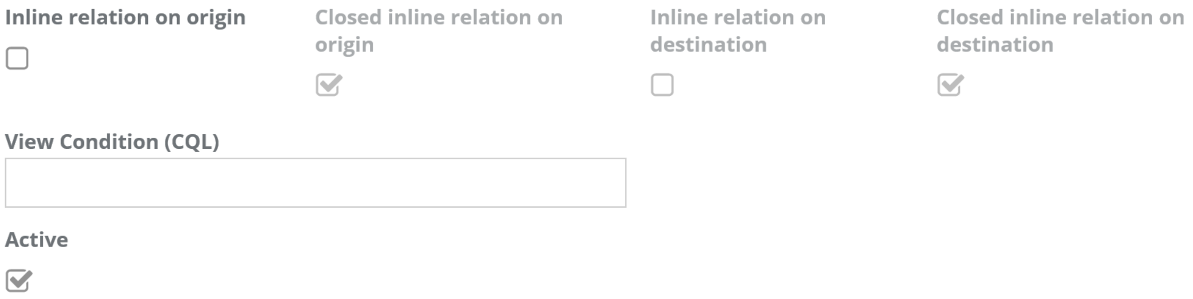
Master detail: ☑

Label master detail: Equipment

Show summary for

Attribute ↑

Select an attribute



Inline relation on origin ☐

Closed inline relation on origin ☑

Inline relation on destination ☐

Closed inline relation on destination ☑

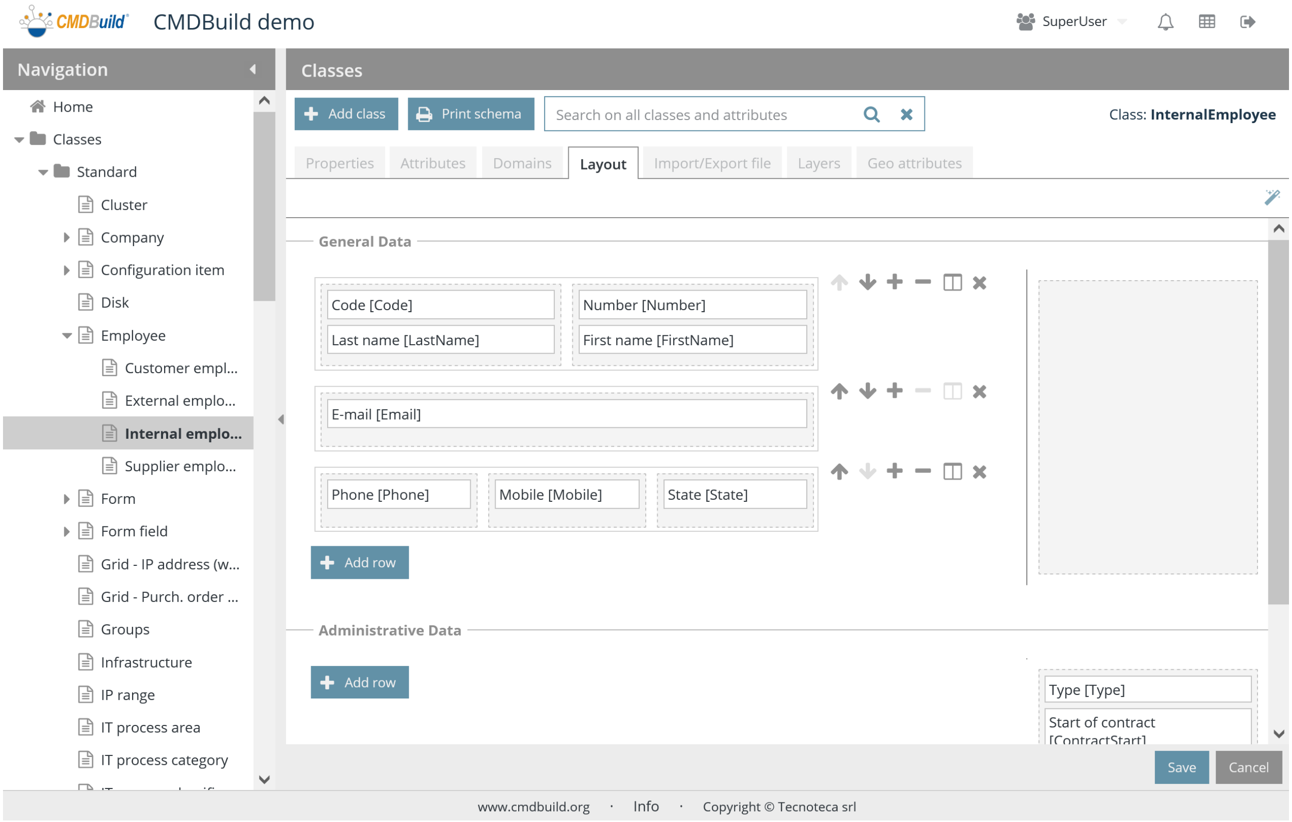
View Condition (CQL)

Active ☑

7.4. Layout TAB

The “Layout” TAB allows to define a custom layout for each class, in terms of number of rows, number and width of columns and positioning of attributes in the resulting cells.

The custom layout is defined separately for each group of attributes (fieldset).



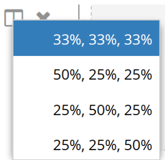
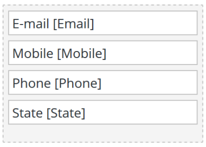
There is the possibility of performing the following operations:

• create / edit the layout	✎
• delete the layout	🗑
• disable the layout (i.e. set the status as non-active)	⊘

7.4.1. Configuration of a custom layout

The definition of a custom layout implicates the following possibilities for each group of attributes (fieldset):

• add a new row	+ Add row
• remove a row	✖

- move a row up	↑
- move a row down	↓
- add a new column	+
- remove a column	−
- edit the proportion among the configured columns	 33%, 33%, 33% 50%, 25%, 25% 25%, 50%, 25% 25%, 25%, 50%
- position one of the available attributes for that group (fieldset) into the layout (with the drag & drop function)	 E-mail [Email] Mobile [Mobile] Phone [Phone] State [State]

Once saved, the layout will be automatically applied by CMDBuild into the Data Management Module both when displaying and editing a data card.

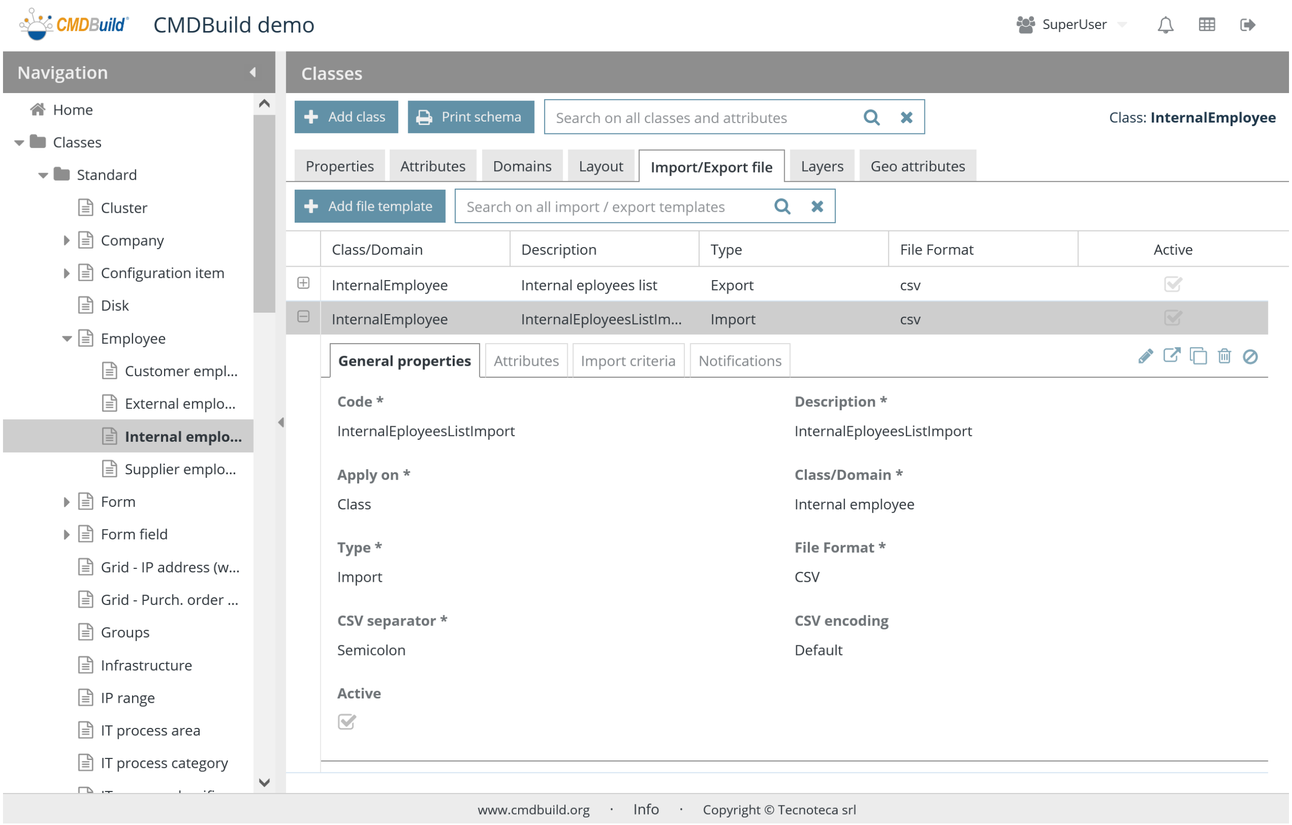
The data card for those classes without a custom layout will be displayed on two columns according to the organization defined in the Attributes TAB.

Any attributes not located in the layout will be displayed with the organization defined in the Attributes TAB.

7.5. Import / Export tab

The "Import / Export" tab displays the list of templates set in the system to execute import / export operations from file in an interactive way (they will be available in the contextual menu in the data Management Module) or in batches (configuring in this case a specific task in the Task Manager in this Administration Module).

Templates gather all the necessary information in order to automatize the operation in case of batch execution and to avoid the duty to specify every time a high number of parameters in case of an interactive execution.



Class: **InternalEmployee**

	Class/Domain	Description	Type	File Format	Active
⊞	InternalEmployee	Internal eployees list	Export	csv	✓
⊟	InternalEmployee	InternalEployeesListIm...	Import	csv	✓

Code *
InternalEployeesListImport

Description *
InternalEployeesListImport

Apply on *
Class

Class/Domain *
Internal employee

Type *
Import

File Format *
CSV

CSV separator *
Semicolon

CSV encoding
Default

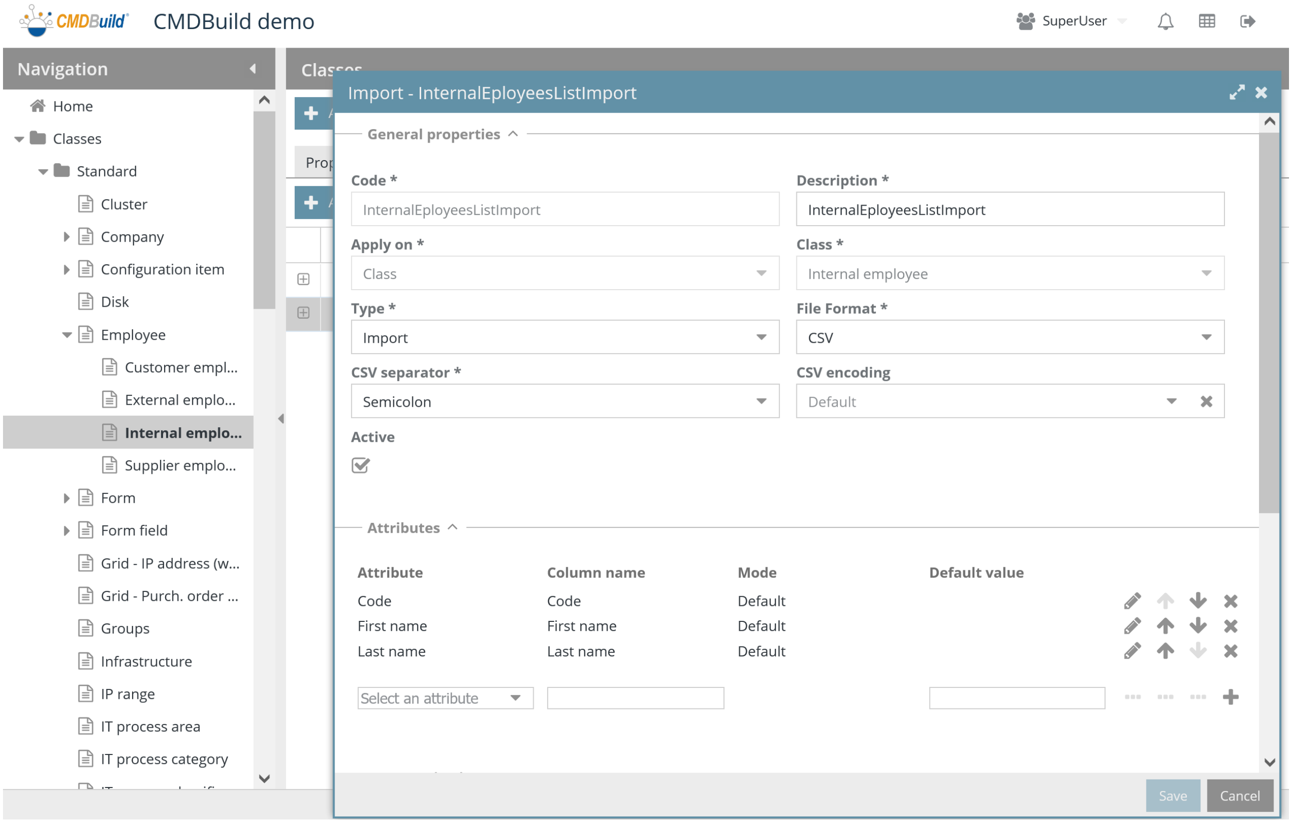
Active
✓

There is the possibility of performing the following operations:

• create a new template	✚ Add file template
• search the template in the list	Search on all import / export templates 🔍 ✖
• expand the current row and display its data	⊞
• edit the selected template	✎
• display the complete information related to the selected template	↗

• clone the selected template	⧉
• delete the selected attribute	🗑
• disable the selected attribute (i.e. set the status as non-active)	⊘

Each domain requires the parameters (metadata) below to be compiled.



Import - InternalEployeesListImport

General properties

Code *: InternalEployeesListImport

Description *: InternalEployeesListImport

Apply on *: Class

Class *: Internal employee

Type *: Import

File Format *: CSV

CSV separator *: Semicolon

CSV encoding: Default

Active: ☑

Attributes

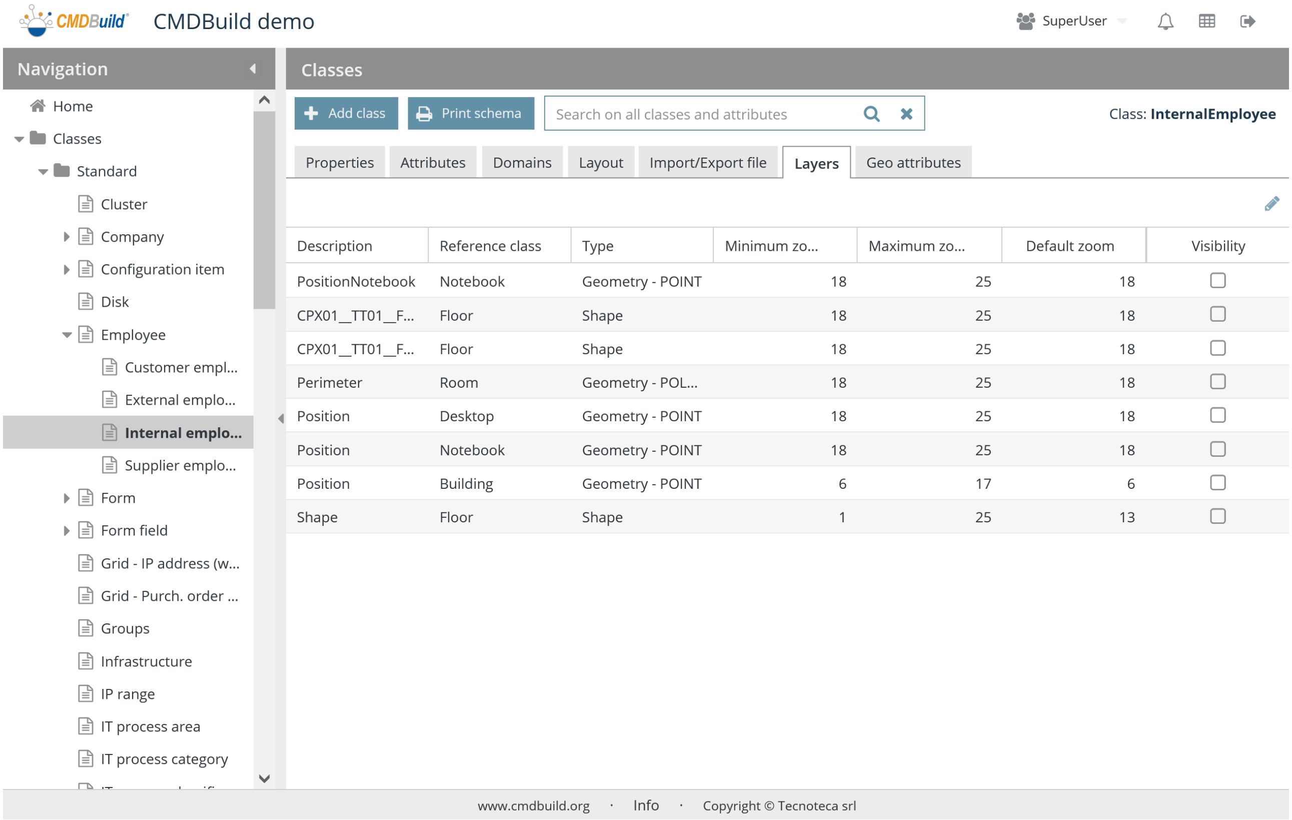
Attribute	Column name	Mode	Default value
Code	Code	Default	
First name	First name	Default	
Last name	Last name	Default	
Select an attribute			

For a complete description of the configuration mode, please see the related section of this manual.

7.6. Layers tab

The "Layers" tab displays the list of all geographical attributes available on the map for the current class.

It includes both layers made of geographic attributes of CMDB objects (room perimeters, computer positions, etc.) and layers loaded by external systems and available as "background" on the map for the current class (shape linked to the planimetry of a floor made by items which are not managed in CMDB such as walls, windows, stairs, etc.)

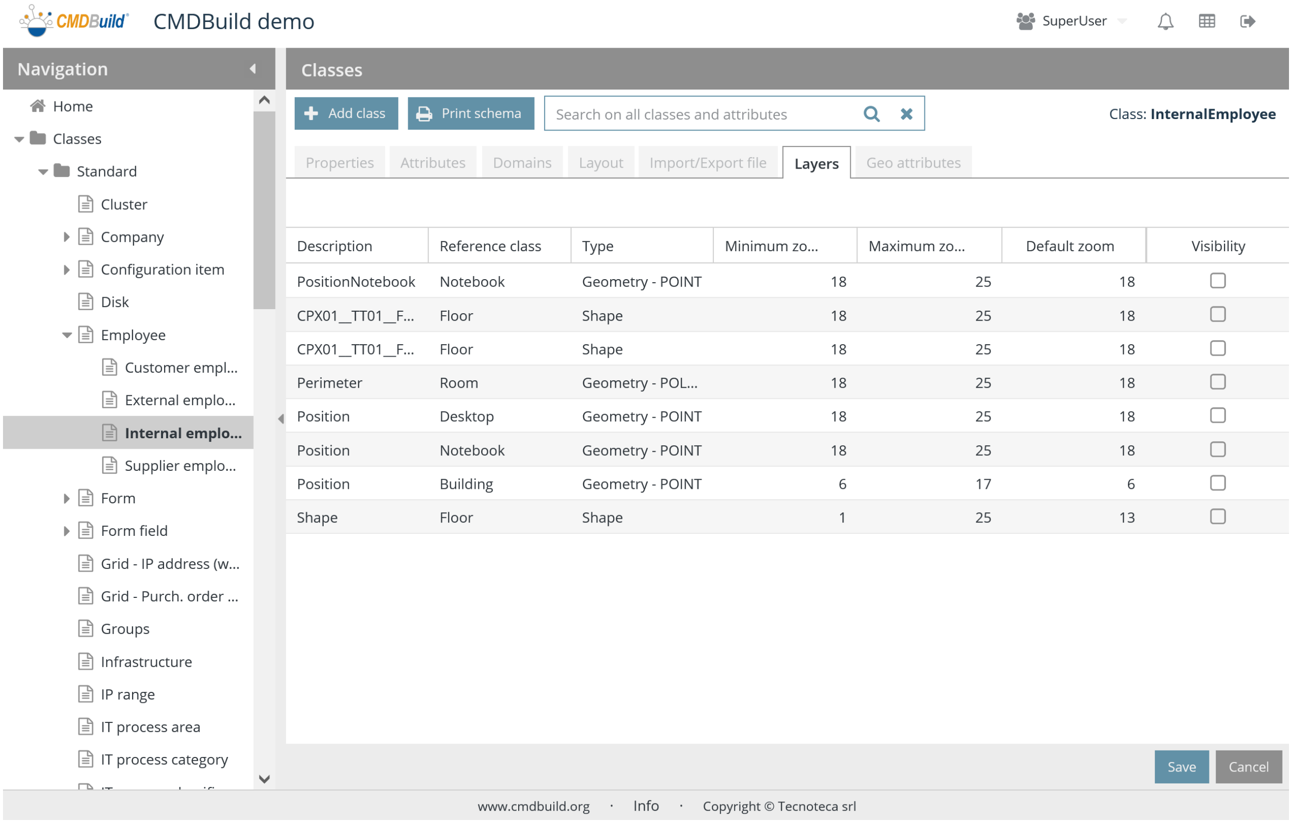


Description	Reference class	Type	Minimum zo...	Maximum zo...	Default zoom	Visibility
PositionNotebook	Notebook	Geometry - POINT	18	25	18	□
CPX01__TT01__F...	Floor	Shape	18	25	18	□
CPX01__TT01__F...	Floor	Shape	18	25	18	□
Perimeter	Room	Geometry - POL...	18	25	18	□
Position	Desktop	Geometry - POINT	18	25	18	□
Position	Notebook	Geometry - POINT	18	25	18	□
Position	Building	Geometry - POINT	6	17	6	□
Shape	Floor	Shape	1	25	13	□

There is the possibility of performing the following operations:

• edit the visibility of the layers available for the current class	✎

When clicking on the edit icon, you will enter the list of layers and for each one you can edit the visibility, as shown in the following screenshot.

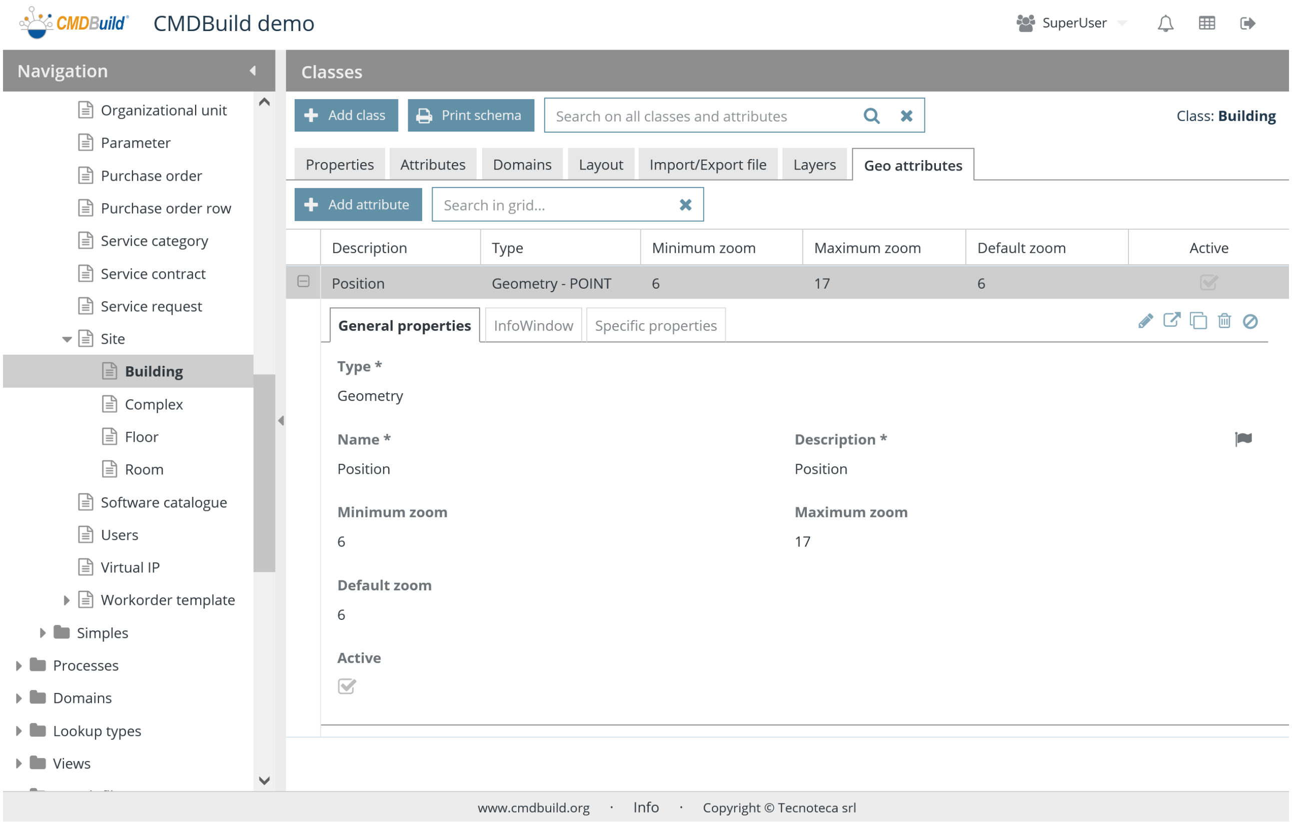


Class: **InternalEmployee**

Description	Reference class	Type	Minimum zo...	Maximum zo...	Default zoom	Visibility
PositionNotebook	Notebook	Geometry - POINT	18	25	18	☐
CPX01__TT01__F...	Floor	Shape	18	25	18	☐
CPX01__TT01__F...	Floor	Shape	18	25	18	☐
Perimeter	Room	Geometry - POL...	18	25	18	☐
Position	Desktop	Geometry - POINT	18	25	18	☐
Position	Notebook	Geometry - POINT	18	25	18	☐
Position	Building	Geometry - POINT	6	17	6	☐
Shape	Floor	Shape	1	25	13	☐

7.7. Geographic attributes tab

The "geographical attributes" tab allows you to manage the geographic attributes of the selected class.



Description	Type	Minimum zoom	Maximum zoom	Default zoom	Active
Position	Geometry - POINT	6	17	6	☑

Type *
Geometry

Name *
Position

Description *
Position

Minimum zoom
6

Maximum zoom
17

Default zoom
6

Active
☑

There is the possibility of performing the following operations:

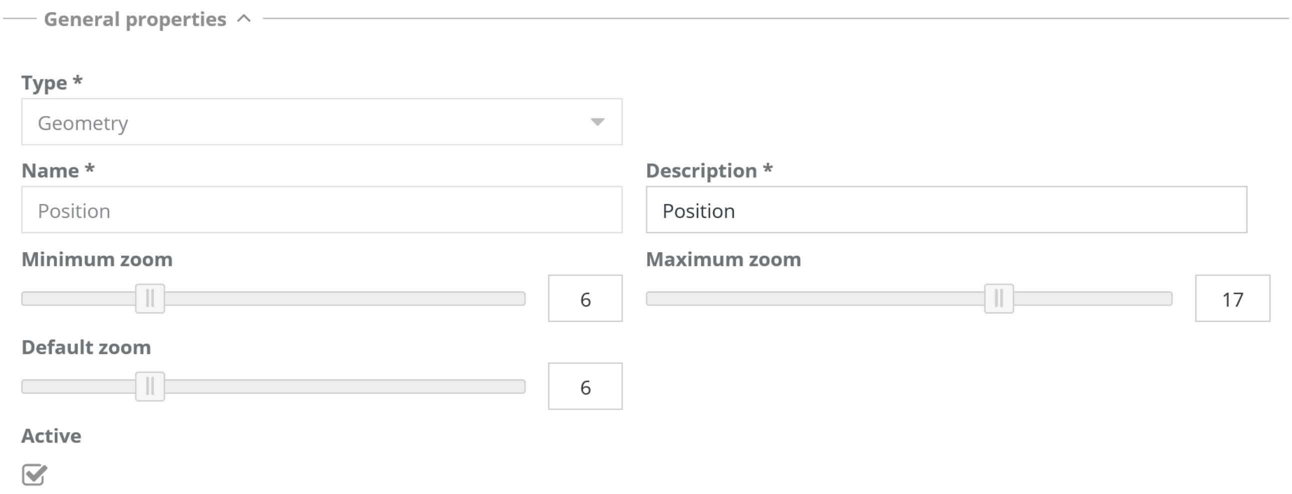
• create a new geographic attribute	Add attribute
• search for the geographical attribute in the class	Search in grid... ✖
• expand the current row and display its data	⊞
• edit the selected attribute (edit available information)	✎
• display the complete information related to the selected attribute	↗
• clone the selected attribute	⧉
• delete the selected attribute (if not inherited and not containing data)	🗑
• disable the selected attribute (i.e. set the status as non-active)	⊘

Each geographical attribute requires the parameters (metadata) below to be compiled.

7.7.1. General properties

The following information is required:

- Type: geometry (point, open polygonal, closed polygonal) or shape (background file)
- Name: attribute name, that corresponds to the physical name of the database column
- "Description": used to report the attribute name in the application
- Minimum zoom: the minimum zoom level required to display the current attribute
- Maximum zoom: the maximum zoom level required to display the current attribute
- Default zoom: the zoom level required to open the map at startup displaying the current geographical attribute
- Active: the attribute is active and can be used



General properties

Type *: Geometry

Name *: Position

Description *: Position

Minimum zoom: 6

Maximum zoom: 17

Default zoom: 6

Active: ☑

7.7.2. Specific properties

For the Geometry type attributes the following information is required:

- Type of geographical attribute, which can be:
  - Line
  - Point
  - Polygon

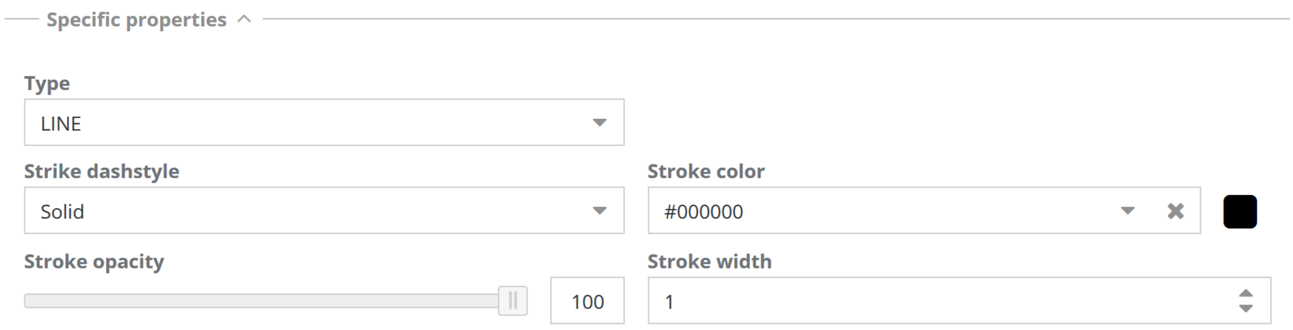
Here's the additional information, differentiated according to the type of geographical attribute.

Line attributes

For Line attributes this additional information is required:

- Stroke dashstyle: dot, dash, dashdot, longdash, longdashdot, solid

- Stroke color
- Stroke opacity
- Stroke width: from 1 to 10



Specific properties

Type: LINE

Strike dashstyle: Solid

Stroke color: #000000

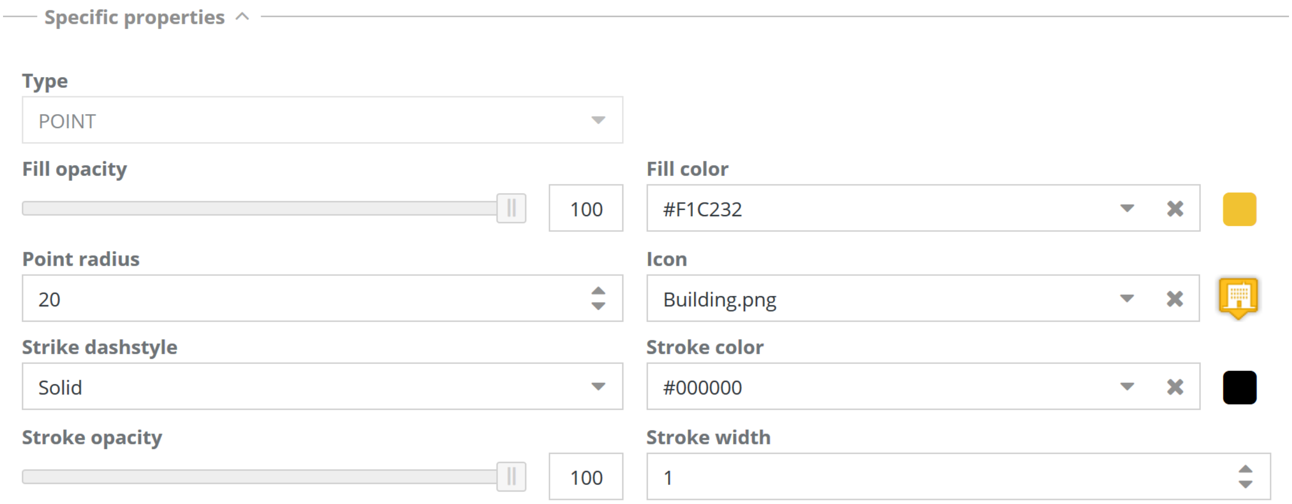
Stroke opacity: 100

Stroke width: 1

Point attributes

For Point attributes this additional information is required:

- Fill opacity
- Fill color
- Point radius: the size of the icon
- Icon: the name of the icon (which should be loaded using GIS pages in the Administration Module) to display on the map
- Stroke dashstyle: dot, dash, dashdot, longdash, longdashdot, solid
- Stroke color
- Stroke opacity
- Stroke width: from 1 to 10



Specific properties

Type
POINT

Fill opacity
100

Fill color
#F1C232

Point radius
20

Icon
Building.png

Strike dashstyle
Solid

Stroke color
#000000

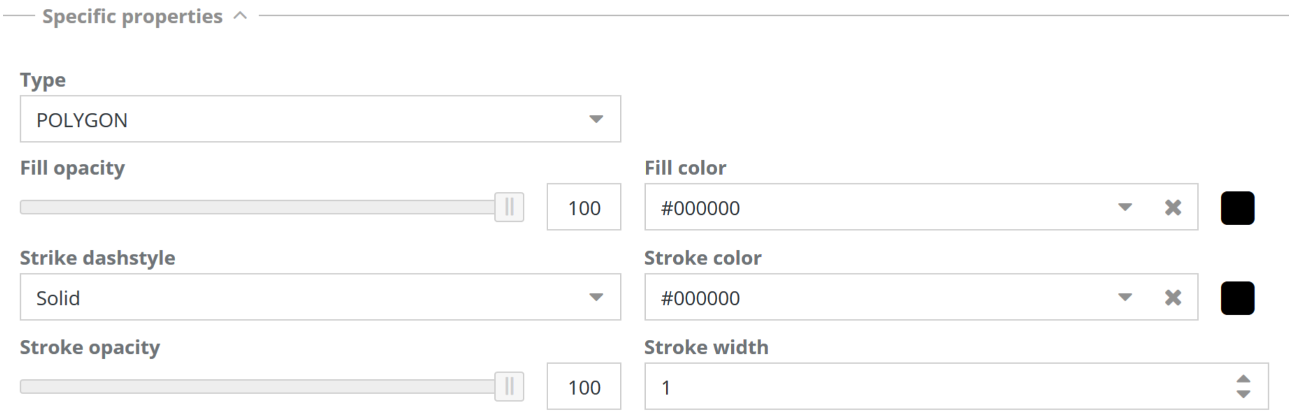
Stroke opacity
100

Stroke width
1

Polygon attributes

For Polygon attributes this additional information is required:

- Fill opacity
- Fill color
- Stroke dashstyle: dot, dash, dashdot, longdash, longdashdot, solid
- Stroke color
- Stroke opacity
- Stroke width: from 1 to 10



Specific properties

Type
POLYGON

Fill opacity
100

Fill color
#000000

Strike dashstyle
Solid

Stroke color
#000000

Stroke opacity
100

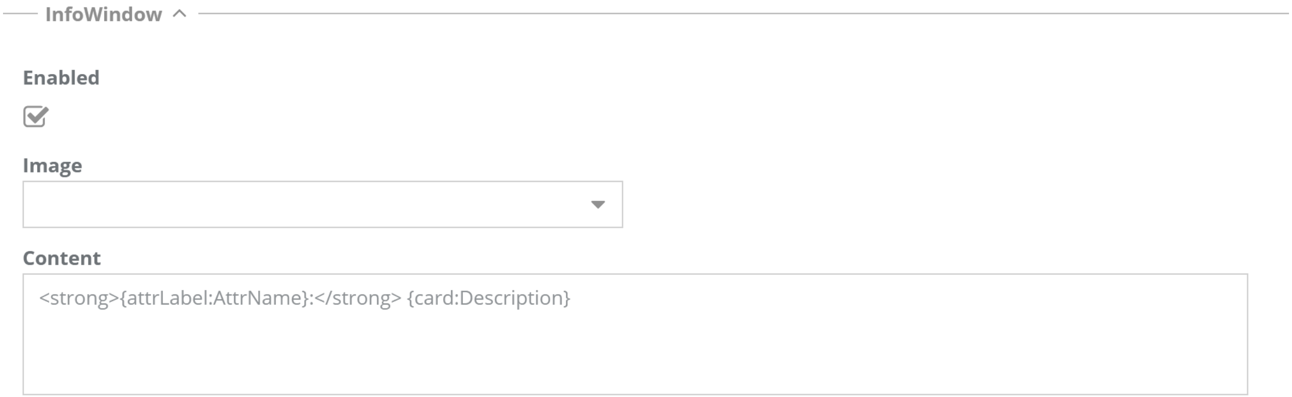
Stroke width
1

7.7.3. Infowindows

You can enable the display of a box containing text and/or images on a map, in correspondence of the selected element.

The following information is required:

- Enabled
- Images: you can select a File attribute to get the image for the infowindow
- Content: text to be inserted into the box, written with CQL, the CMDBuild templating language



InfoWindow

Enabled

☑

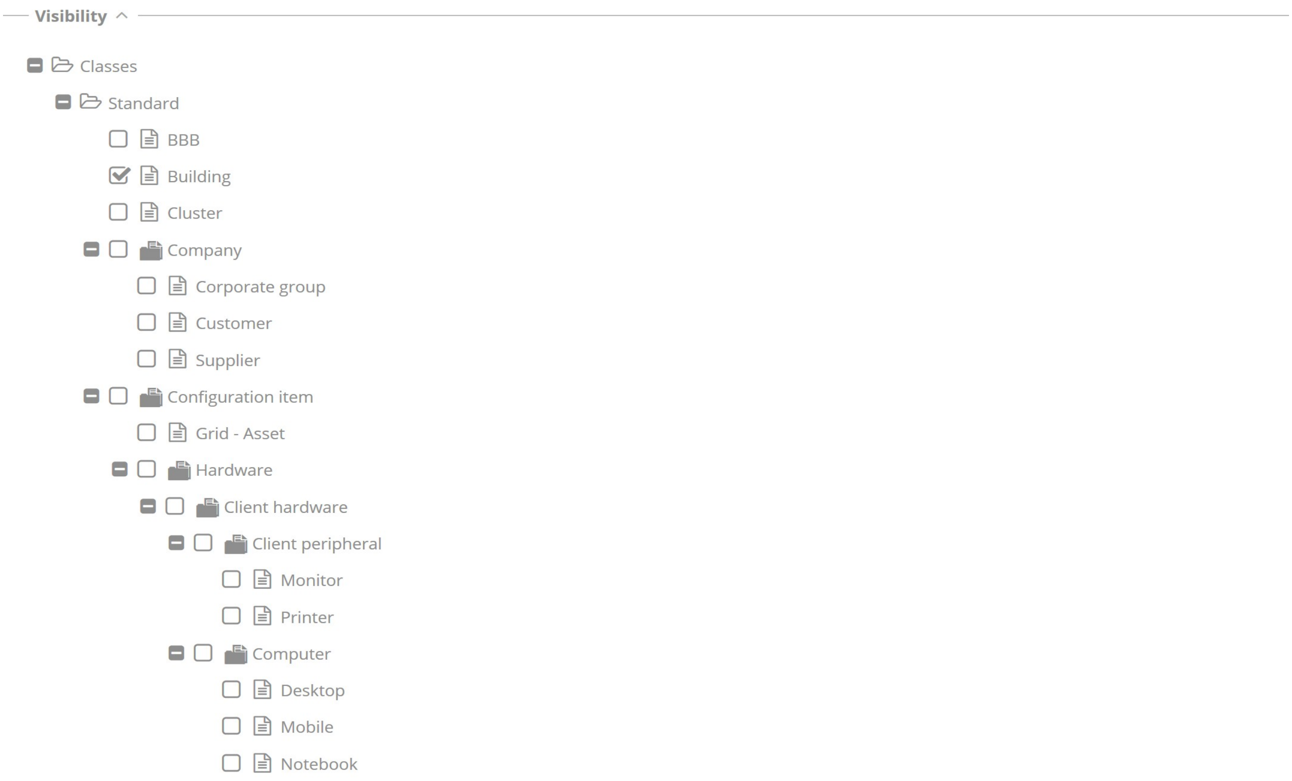
Image

Content

<strong>{attrLabel:AttrName}:</strong> {card:Description}

7.7.4. Visibility

Here's a part of classes configured in the system, represented hierarchically (at the first level the not-inherited superclasses and classes, then the first-level subclasses, the second-level ones, etc.) with the possibility of specifying on which classes the current geographical attribute has to be visible.



Visibility

- Classes
  - Standard
    - □ BBB
    - ☑ Building
    - □ Cluster
    - □ Company
      - □ Corporate group
      - □ Customer
      - □ Supplier
    - □ Configuration item
      - □ Grid - Asset
      - □ Hardware
        - □ Client hardware
          - □ Client peripheral
            - □ Monitor
            - □ Printer
          - □ Computer
            - □ Desktop
            - □ Mobile
            - □ Notebook

8. Processes

A process (o workflow) is a sequence of activities and each of these can be handled by operators with different roles. Each activity represents an action to reach a pre-defined aim.

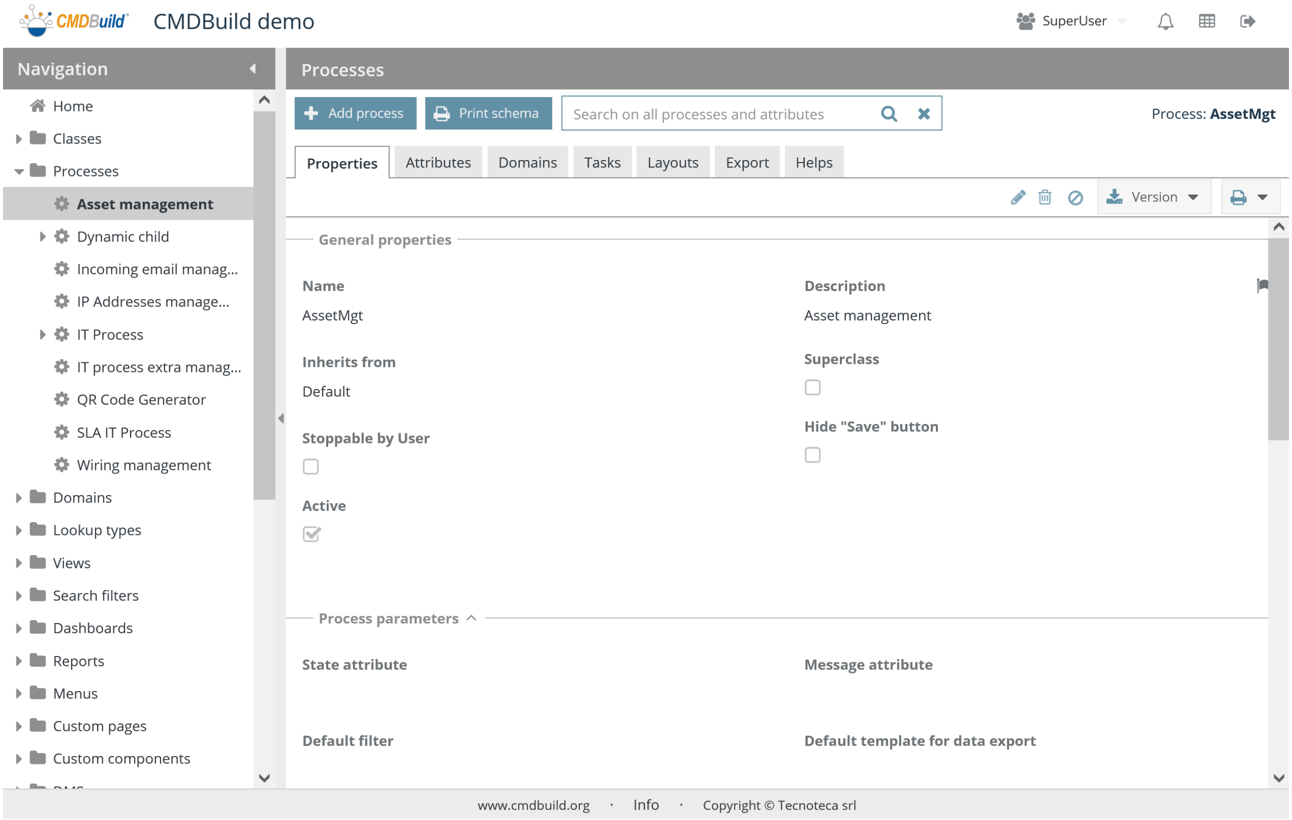
In CMDBuild the Administration Module allows you to define the data structure for the information persistence managed in the workflow; the visual editor TWE allows to design its flow (see the paragraph 2.4.2); the Management Module offers to operators the forms that have to be compiled for the process advancement.

8.1. Properties tab

The "Properties" tab allows you to create new processes and edit the old ones.

The definition of the data structure of a process is similar to the definition of the data structure of a class, as described in the previous chapter.

Hereinafter the specific features of processes will be described and the features of a normal class not applicable to processes and the specific additions of the processes will be highlighted.



General properties

Name: AssetMgt

Description: Asset management

Inherits from: Default

Superclass: ☐

Stoppable by User: ☐

Hide "Save" button: ☐

Active: ☑

Process parameters

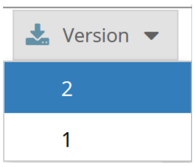
State attribute

Message attribute

Default filter

Default template for data export

All operations are available for the classes, including the following one:

• download the XPDL file in the chosen version	

The download feature allows you to export the XPDL template to start working on the process structure using an external editor (TWE).

The exported file includes:

- process name
- list of attributes process
- the list of users' groups defined in CMDBuild, among which you can choose the "actors" (roles) that take part in the process
- data type description for custom Lookup, LookupArray and Reference attributes, which can be used in the process scripts

Also the processes require several parameters to be compiled (metadata), divided into various sections.

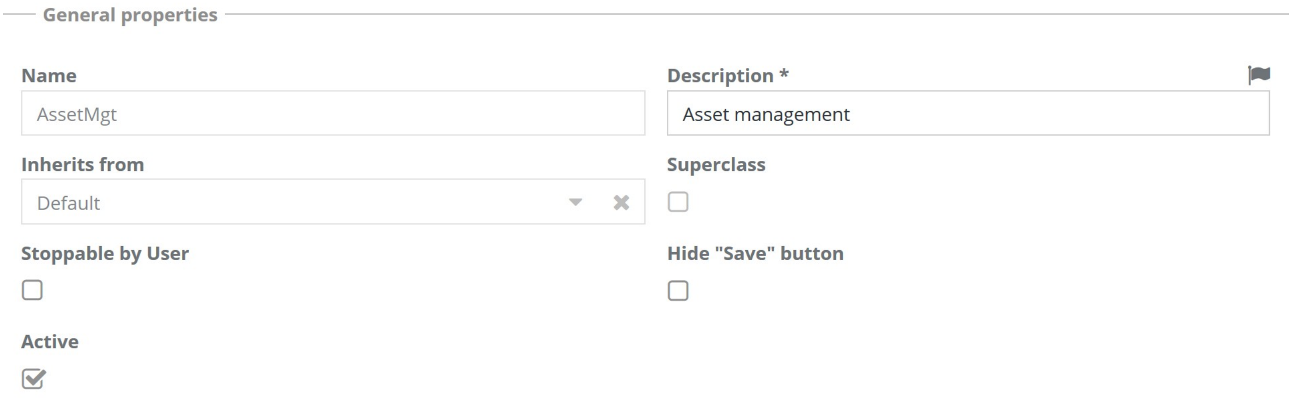
8.1.1. General properties

Regarding the class definition, the following information is required for processes:

- Stoppable by user: if enabled, it makes available an additional button that interrupts the process (independently of its status)
- Hide "Save" button: you do not see the button to save the current status of the process before proceeding to the following status

The following information is not managed:

- Type, which can take the Standard and Simple values



General properties

Name: AssetMgt

Description *: Asset management

Inherits from: Default

Superclass: □

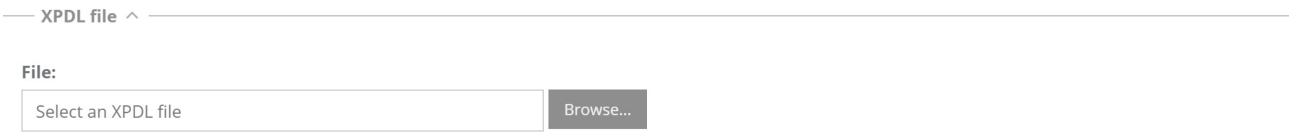
Stoppable by User: □

Hide "Save" button: □

Active: ☑

8.1.2. XPDL file

It allows to import the process XPDL descriptor designed with the external editor TWE (Together Workflow Editor).



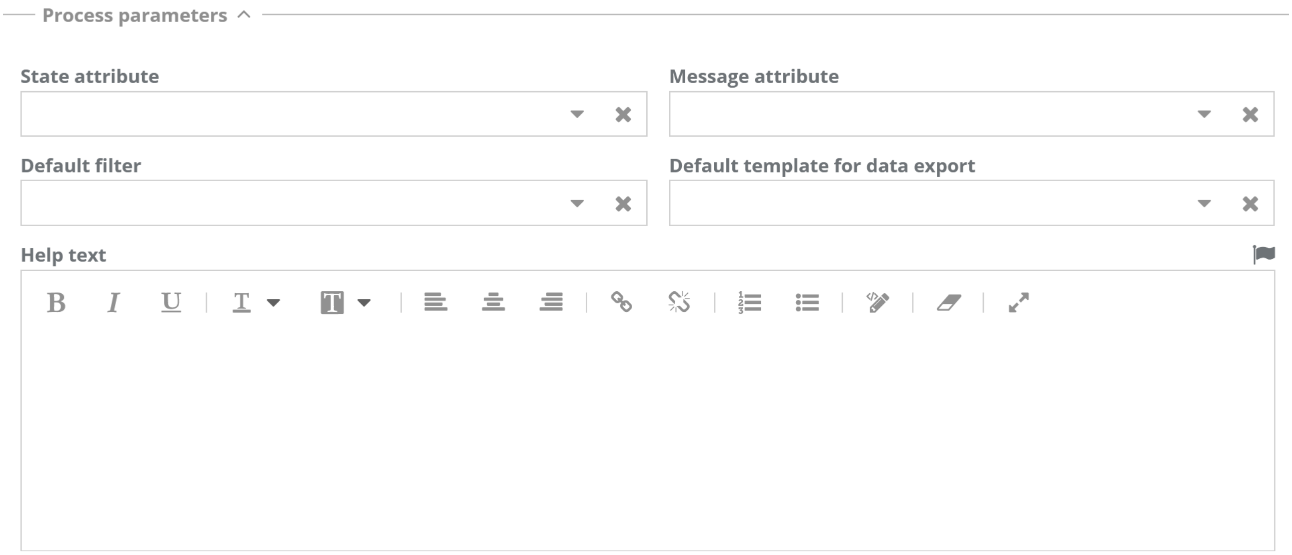
8.1.3. Other process parameters

The following information is required:

- State attribute: it allows to specify which process attribute (among the Lookup types) graphically represents (on the top of the card) the advance status of the process in the Management Module
- Message attribute: it allows to specify which process attribute is used to display error messages, locating it under the status representation
- Default filter: it allows you to specify the default filter (selected among the filters defined for that process in the specific function) that has to be applied in the list of process instances
- Default template for export: it allows to specify the default template in order to execute the export of data from the current process, which must already be configured
- Help text: available for operators in the Management Module, just like for classes

The advancement card of a process instance is characterized by another important information related to the action. Since the corresponding attribute is specific for each step of advancement process, CMDBuild requires that it is specified within the XPDL descriptor. In case the advancing action is univocal for the current step (i.e. the operator has no multiple choices), on the user interface there will be the text "Proceed" as non-editable.

From the version CMDBuild 3.4 the execution workflow engine is Tecnoteca River. The option Enhydra Shark is no longer available.



8.1.4. Other sections

Sections related to the management of Attachments, Data Organization, Grouping of attributes, form trigger, Configuration of Contextual Menus and Contents are managed as for Classes. For further details read the relative paragraphs.

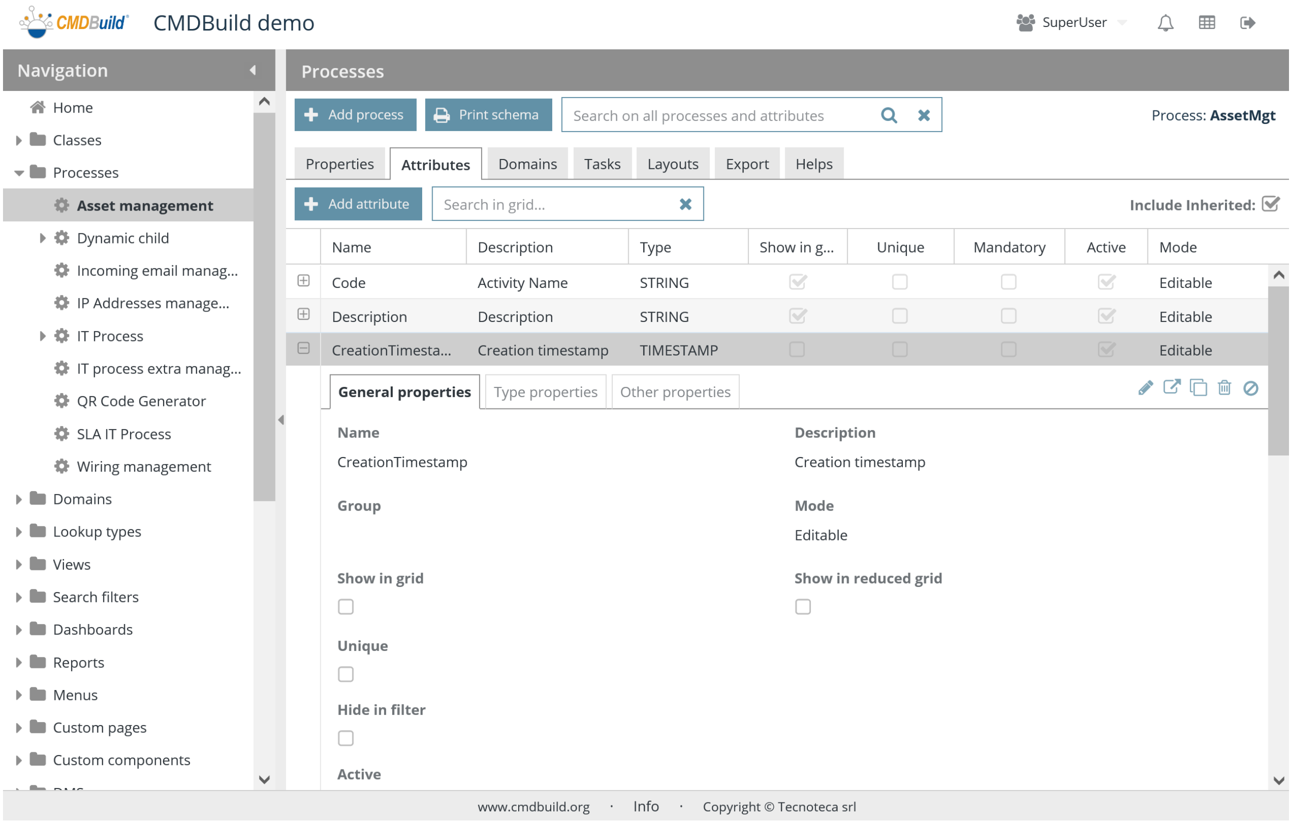
The following sections are not available for processes:

- Validation rules (since managed in the scripts of the XPDL file)
- Form trigger (since managed in the scripts of the XPDL file)
- Form widget (since managed in the scripts of the XPDL file)

8.2. Attributes tab

The Attributes tab allows you to manage the attributes of the selected process.

The managed information is the same as described for Classes. For further details, please read the relative paragraph.



Process: **AssetMgt**

Include Inherited: ☑

	Name	Description	Type	Show in g...	Unique	Mandatory	Active	Mode
⊞	Code	Activity Name	STRING	☑	□	□	☑	Editable
⊞	Description	Description	STRING	☑	□	□	☑	Editable
⊟	CreationTimesta...	Creation timestamp	TIMESTAMP	□	□	□	☑	Editable

Name
CreationTimestamp

Description
Creation timestamp

Group

Mode
Editable

Show in grid
□

Show in reduced grid
□

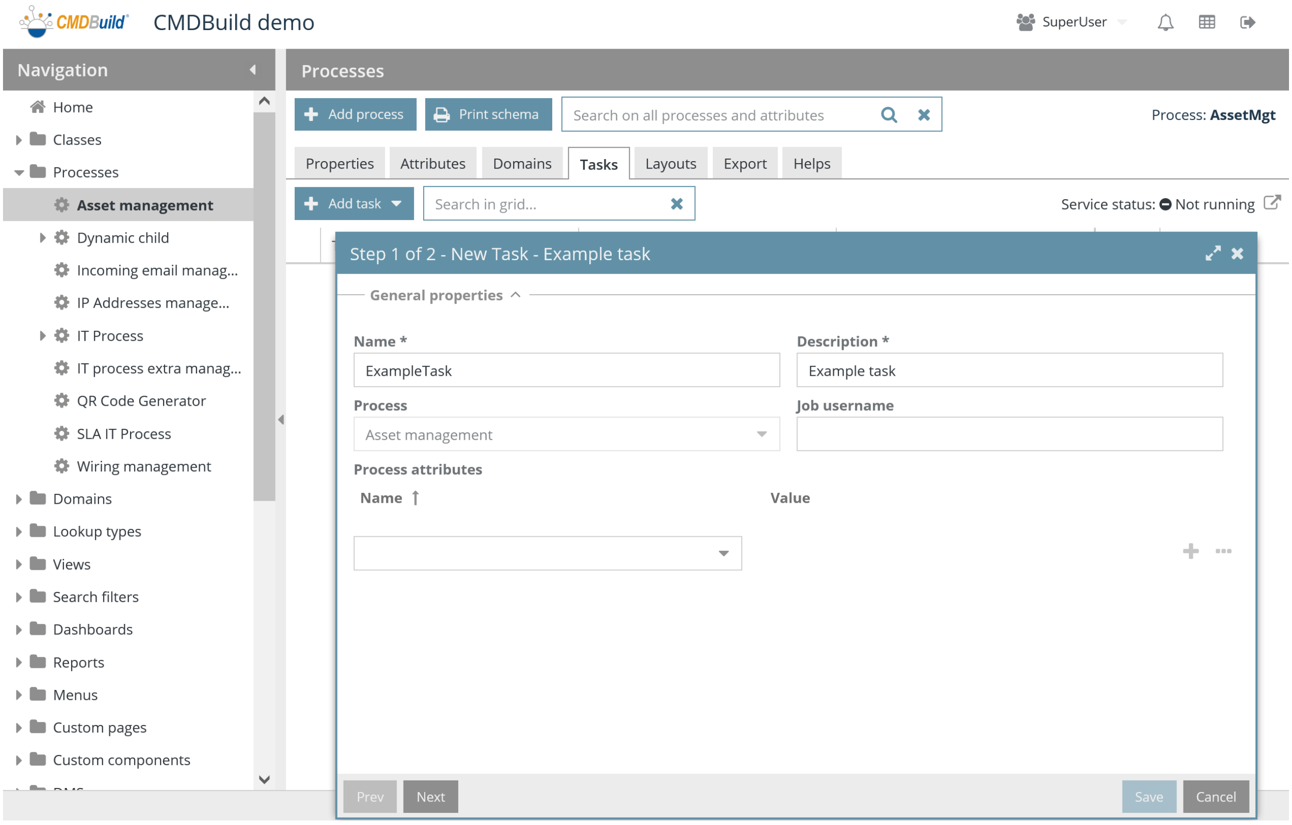
Unique
□

Hide in filter
□

Active

8.3. Task tab

The function allows you to display any Tasks (chapter 20) already configured to start automatically the processes, or configure new ones.



Name *: ExampleTask

Description *: Example task

Process: Asset management

Job username:

8.4. Other tabs

The information managed in the tabs Domains, Layout and Export are the same as described for Classes.

For further details, please read the corresponding paragraphs.

The Help TAB allows you to insert a help text suitable for each process step.

9. Domains

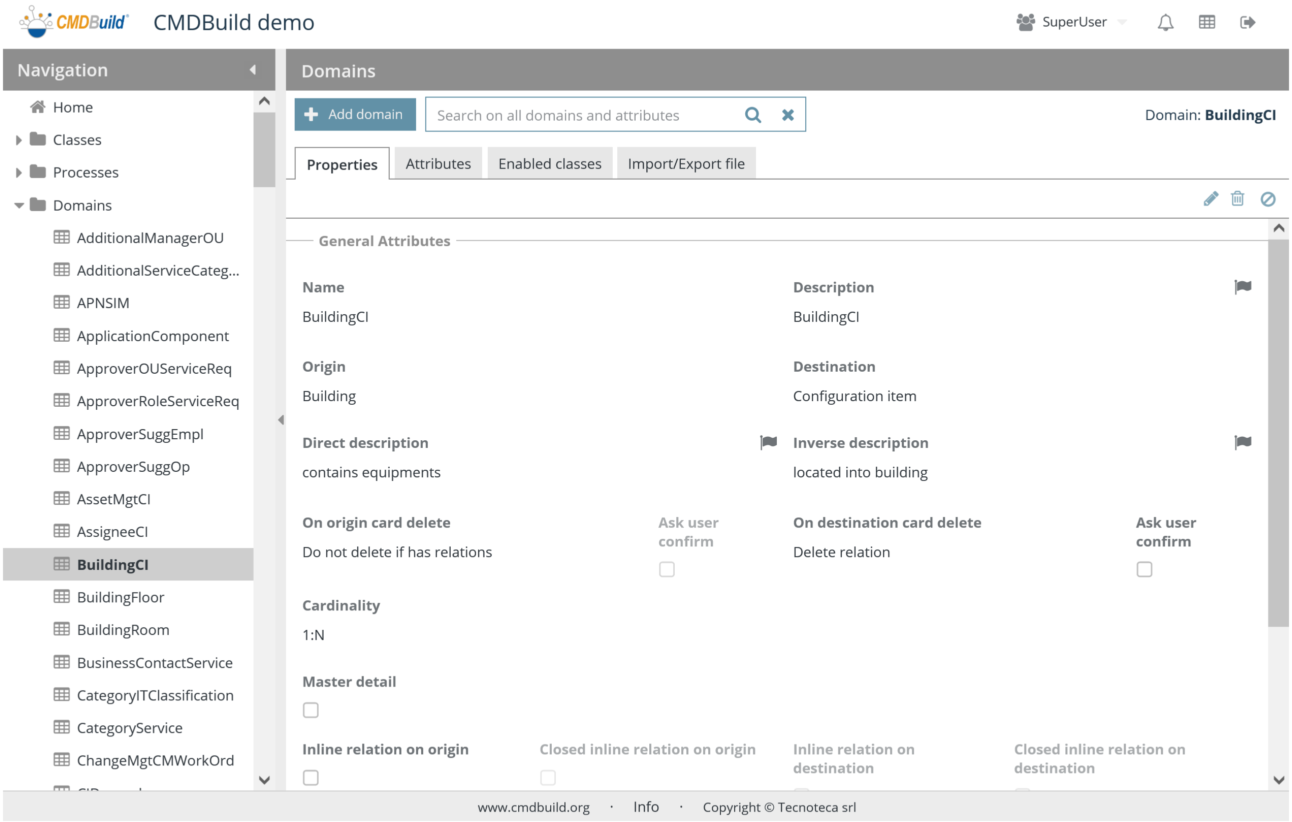
A domain in CMDBuild includes all couples of elements of the CMDB (data cards of configuration items) which are linked thanks to a specific correlation, for example "supplier supplies assets", "user uses job roles", "computer installed software", etc.

A domain is also defined by a couple of classes, by a description of the relation in two directions and by its cardinality.

A domain can be enriched by attributes (similar to a class, excluding the "Reference" type and geographical attributes) and requires a table in the CMDBuild database in order to file the included relations (also in case of cardinality 1:N).

9.1. Properties tab

The "Properties" tab allows you to create new domains and edit the old ones.



Domain: **BuildingCI**

General Attributes

Name: BuildingCI

Description: BuildingCI

Origin: Building

Destination: Configuration item

Direct description: contains equipments

Inverse description: located into building

On origin card delete: Do not delete if has relations — Ask user confirm ☐

On destination card delete: Delete relation — Ask user confirm ☐

Cardinality: 1:N

Master detail ☐

Inline relation on origin ☐ Closed inline relation on origin ☐ Inline relation on destination Closed inline relation on destination

There is the possibility of performing the following operations:

- add a new domain	✚ Add domain
- do searches on configured domains	Search on all domains and attributes 🔍 ✖

• edit the selected domain	✏
• delete the selected domain (physical removal if not containing data, otherwise logic)	🗑
• disable the selected domain (i.e. set the status as non-active)	⊘

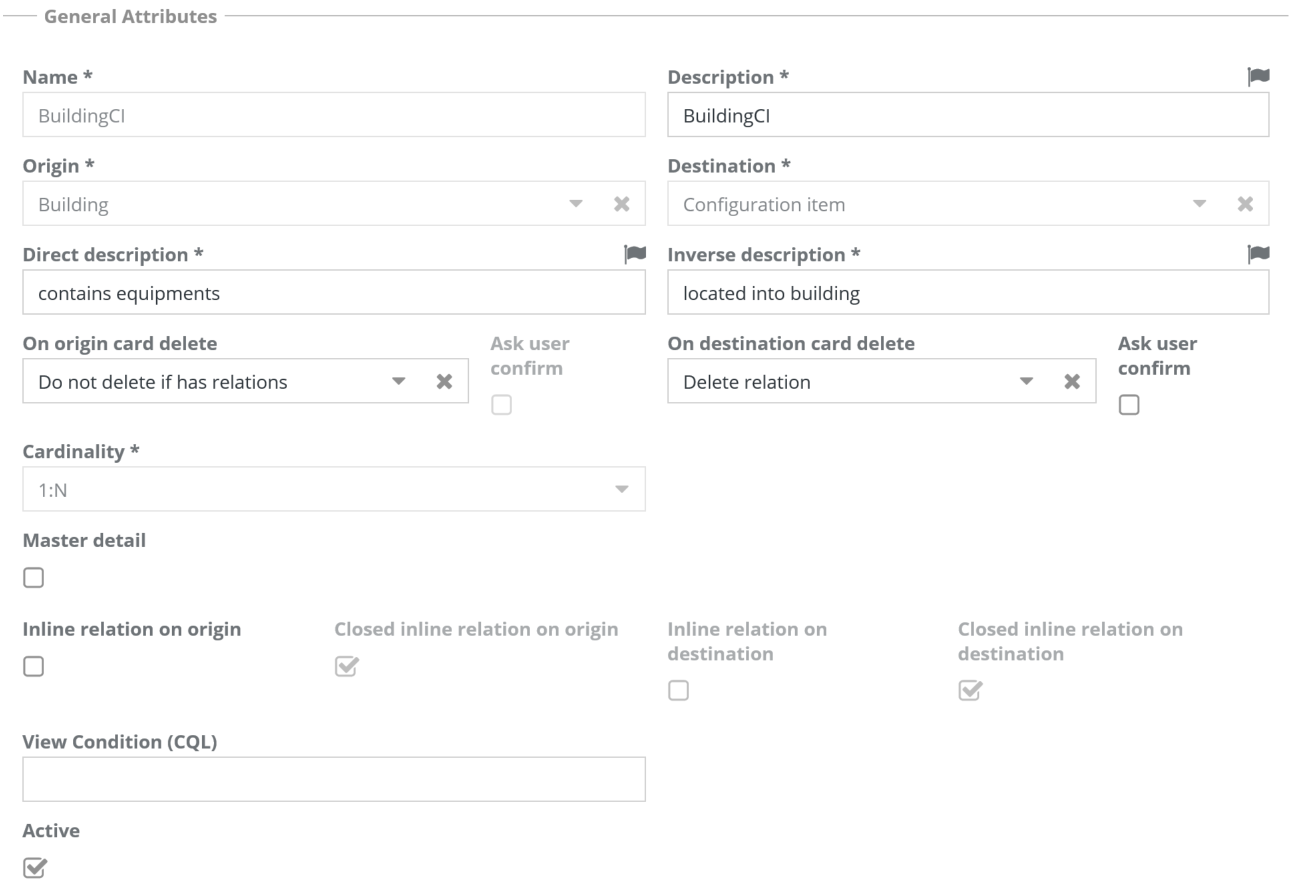
Each domain requires several parameters to be compiled (metadata), which are included in the section of the general properties.

9.1.1. General properties

The following information is required:

- Name: domain class, that corresponds to the physical name of the table in the database
- Description: domain description, that has no real employment in the application and does not appear in the Management Module
- Origin: starting class of the relation (e.g. "Supplier")
- Destination: arrival class of the relation (e.g. "Asset")
- Direct description: relation characterization from the starting class to the arrival class (e.g. "has provided" the assets)
- Inverse description: relation characterization from the arrival class to the starting class (e.g. "provided by the supplier")
- In case of deletion of the original data card: don't delete anything else, delete only the relation, delete also the destination data card of the relation
- In case of deletion, ask confirmation to the operator: yes or no
- In case of deletion of the destination data card: don't delete anything else, delete only the relation, delete also the original data card of the relation ("cascade")
- In case of deletion, ask confirmation to the operator: yes or no
- Cardinality: a constraint on the number of times an entity can appear in a relation (1:N, N:1, N:N), used by the application to insert- new data
- Master Detail: it indicates whether managing the cards of the class linked on the "N" side to the Master Detail tab of the current class; in this case, in the Master Detail tab the list of "children" cards is displayed with the functions of inserting, editing and deleting
- Master Detail label: it allows you to specify the label of the button that manages the "children" cards for the current domain in the Master Detail tab
- Inline origin: states if, in the Management Module, the card list related to the original class have to be shown in the primary TAB instead of in the Master Detail TAB
- Closed default: in case you chose to view the origin cards in the inline mode, you can request the related fieldset to be shown as closed when you open the card
- Inline destination: states if, in the Management Module, the card list related to the destination class have to be shown in the primary TAB instead of in the Master Detail TAB
- Closed default: in case you chose to view the destination cards in inline mode, you can request the related fieldset to be shown as closed when you open the card

- Display condition: it allows you to specify a CQL filter that limits the number of displayed "children" cards
- Active: the domain is active and can be used



General Attributes

Name *: BuildingCI

Description *: BuildingCI

Origin *: Building

Destination *: Configuration item

Direct description *: contains equipments

Inverse description *: located into building

On origin card delete: Do not delete if has relations

Ask user confirm □

On destination card delete: Delete relation

Ask user confirm □

Cardinality *: 1:N

Master detail □

Inline relation on origin □

Closed inline relation on origin ☑

Inline relation on destination □

Closed inline relation on destination ☑

View Condition (CQL):

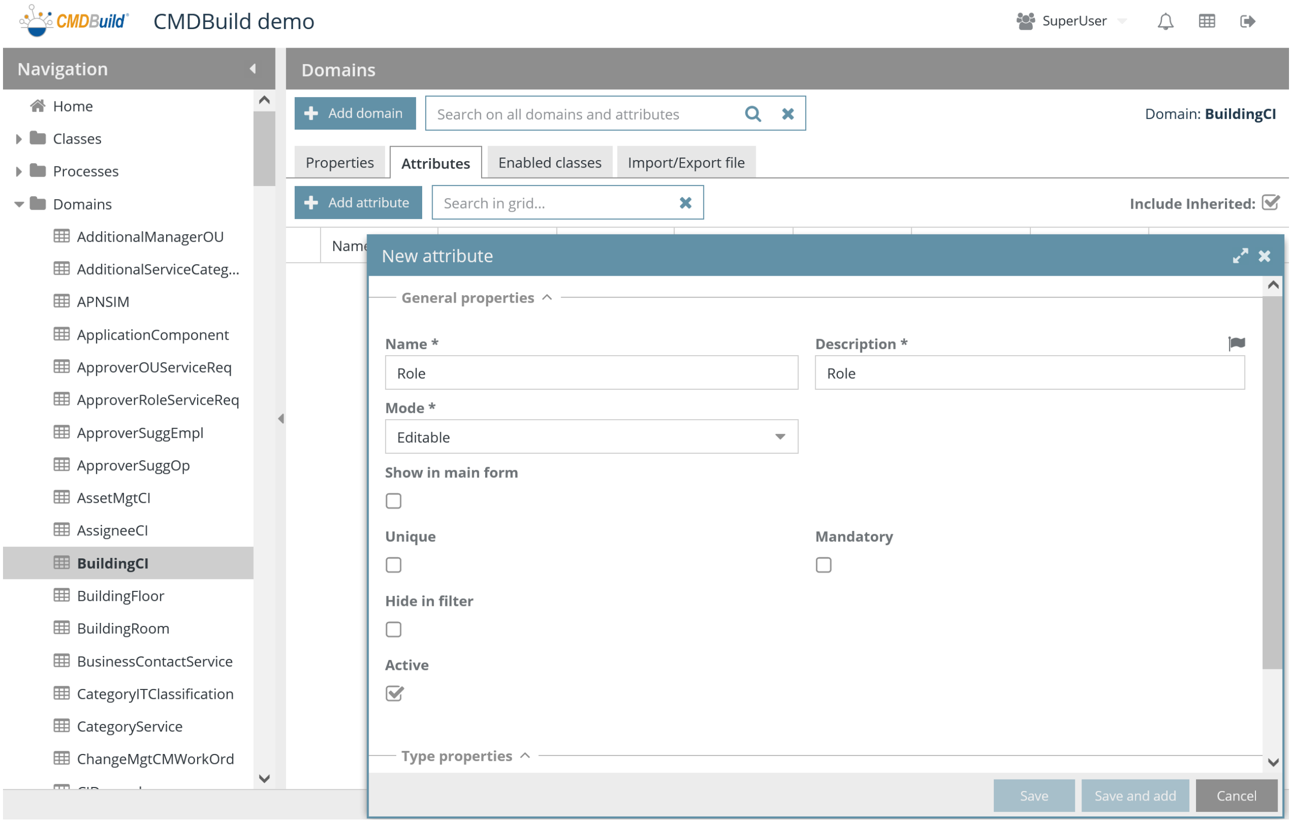
Active ☑

9.2. Attributes tab

The "Attributes" tab allows you to manage the domain attributes.

The management of attributes on domains is similar to the management of attributes on classes. But the Reference attribute is not available (otherwise the situation would be recursive).

For detailed information regarding the attribute configuration please read the corresponding chapter.



Domain: **BuildingCI**

Include Inherited: ☑

Name * Role

Description * Role

Mode * Editable

Show in main form ☐

Unique ☐

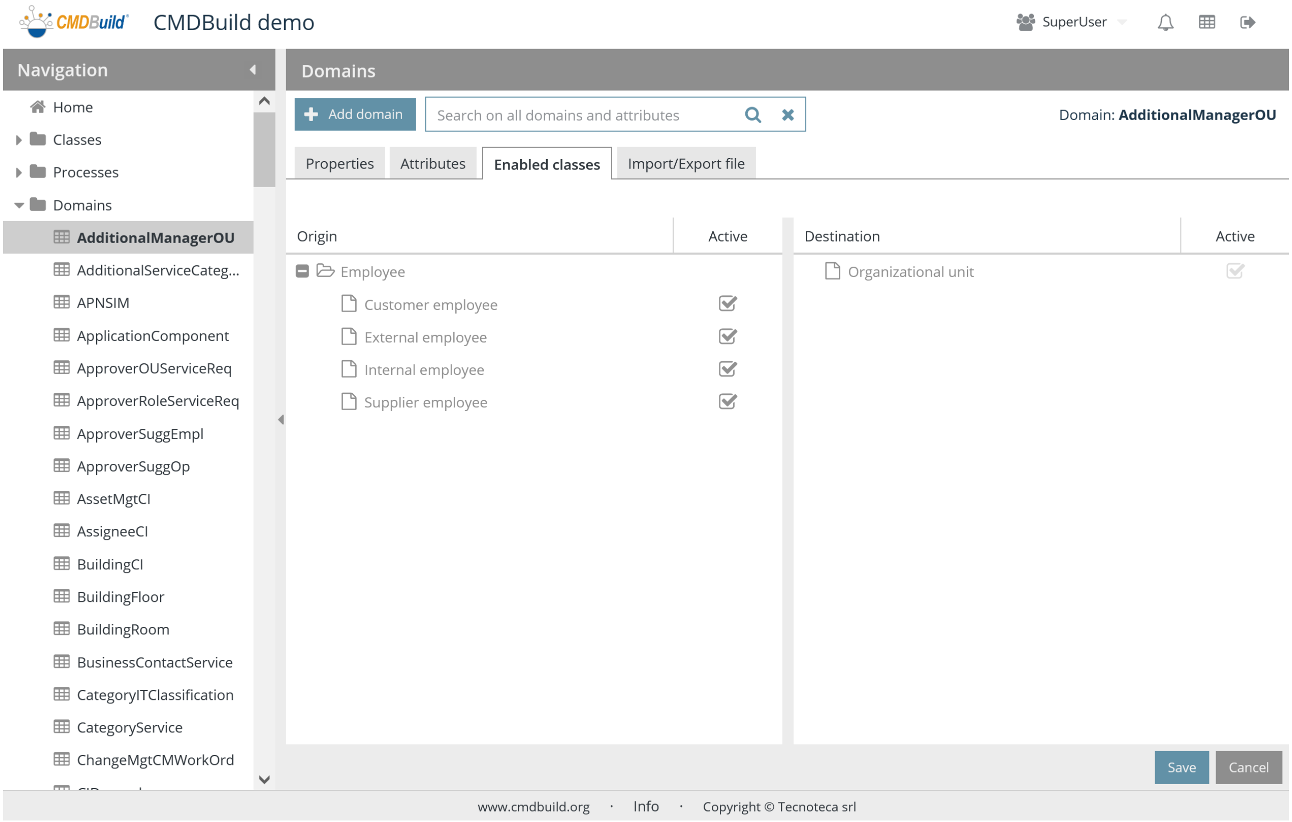
Mandatory ☐

Hide in filter ☐

Active ☑

9.3. TAB Enabled classes

The TAB “Enabled classes” allows you to disable a domain defined on a superclass on one or more subclasses.



Domain: **AdditionalManagerOU**

Origin	Active
Employee	
Customer employee	☑
External employee	☑
Internal employee	☑
Supplier employee	☑

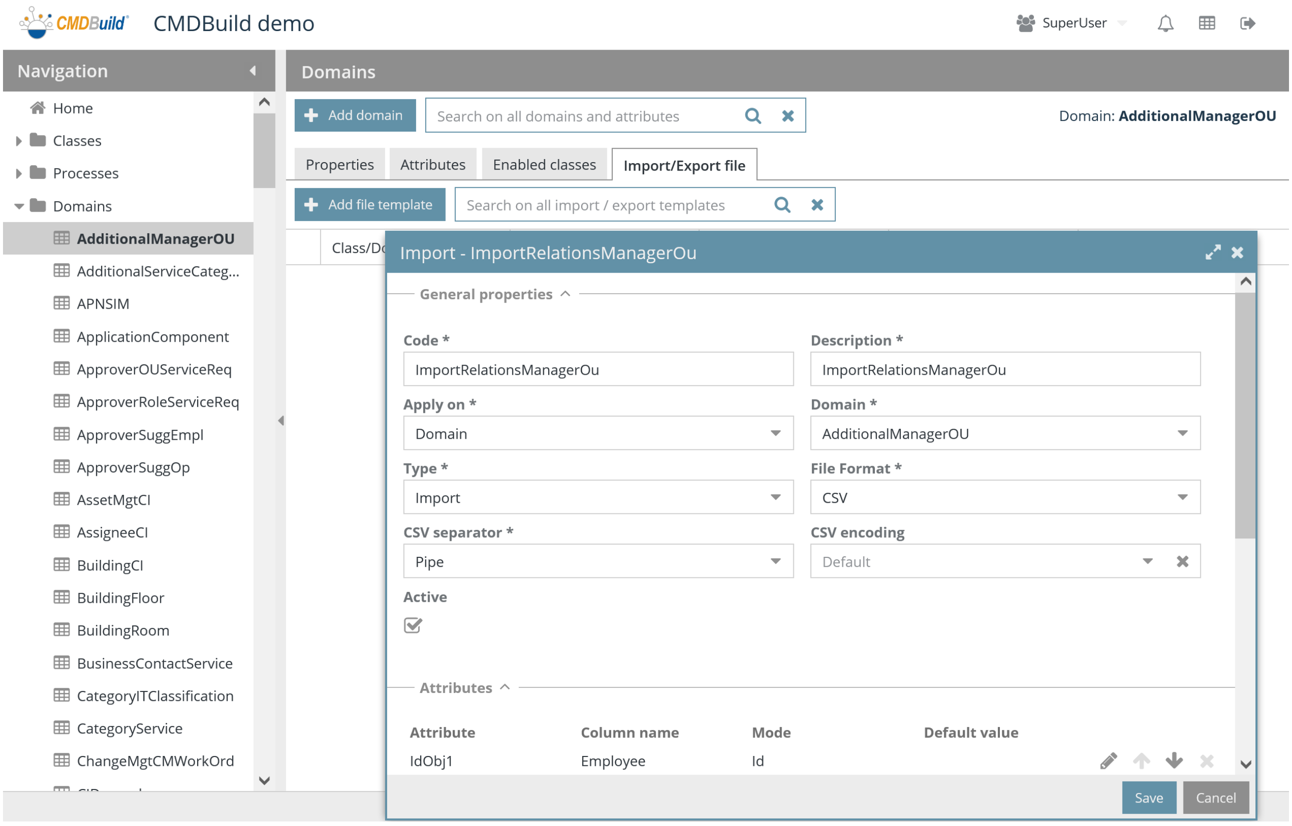
Destination	Active
Organizational unit	☑

9.4. Import / Export tab

The "Import / Export" tab displays the list of templates set in the system to execute interactive import / export operations or batches.

The management is similar to that described in the Import / Export TAB of classes.

For detailed information regarding the template configuration please read the corresponding chapter.



Domain: **AdditionalManagerOU**

Import - ImportRelationsManagerOu

General properties

Code *	ImportRelationsManagerOu
Description *	ImportRelationsManagerOu
Apply on *	Domain
Domain *	AdditionalManagerOU
Type *	Import
File Format *	CSV
CSV separator *	Pipe
CSV encoding	Default
Active	☑

Attributes

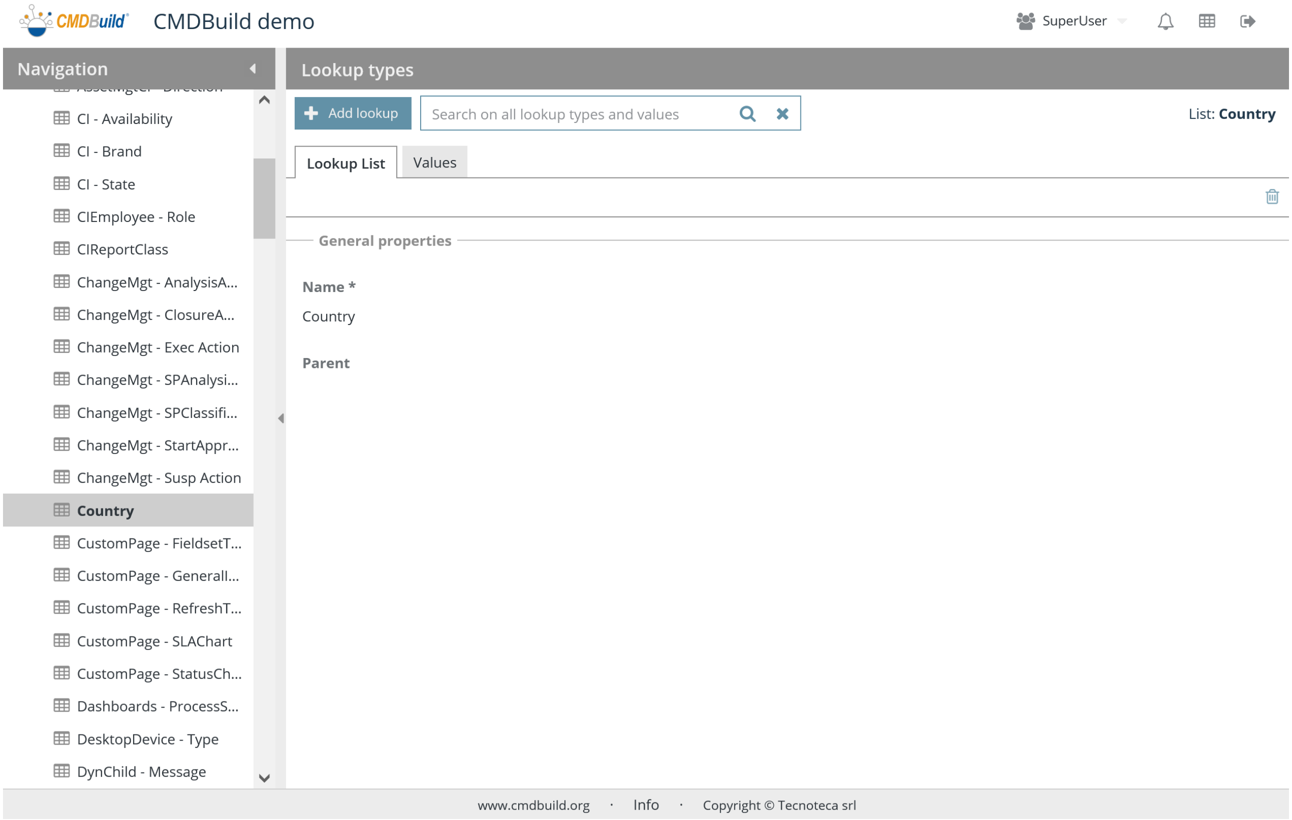
Attribute	Column name	Mode	Default value
IdObj1	Employee	Id	

10. Lookup lists

CMDBuild uses the Lookup lists so that an attribute shows up as a selection with predefined options.
Lookup lists can be used for both single-choice and multiple-choice Lookup attributes.
Therefore, the user must select one or more options from the list of values available, so that some information can be standardized.
For single-choice lookup attribute, you can create multi-level lookup attributes.

10.1. Properties tab

The "Properties" tab allows you to create new Lookup lists and edit the old ones.



General properties

Name *
Country

Parent

There is the possibility of performing the following operations:

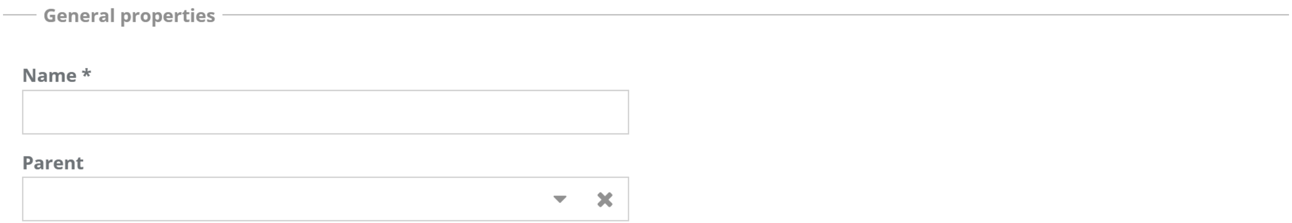
• add a new lookup list	+ Add lookup
• do searches on Lookup lists	Search on all lookup types and values 🔍 ✖
• delete the selected Lookup list (physical removal if not containing data, otherwise logic)	🗑

Each Lookup list requires the parameters (metadata) below to be compiled.

10.1.1. General properties

The following information is required:

- Name: name of Lookup list
- Parent: selection of any parent Lookup list in order to define hierarchical lists



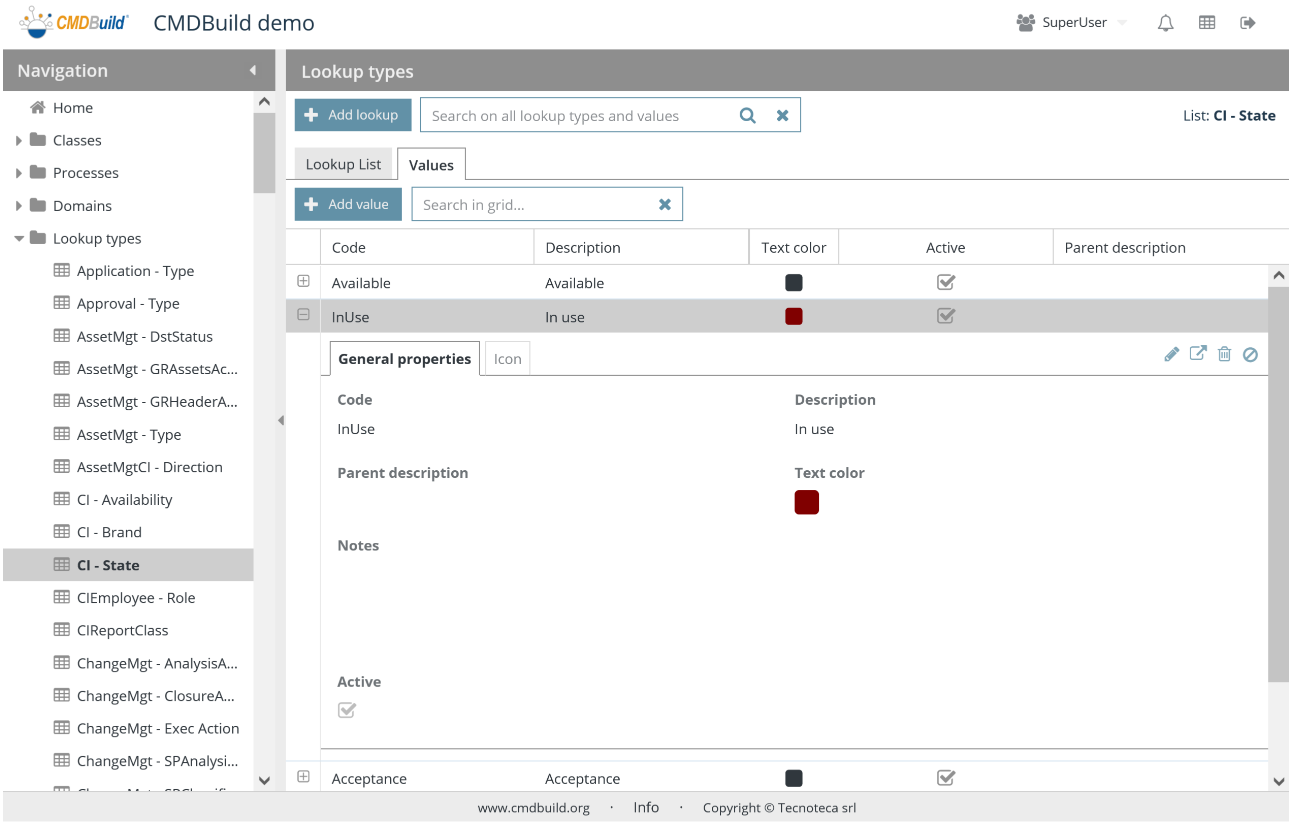
General properties

Name *

Parent

10.2. Values TAB

The Values tab allows to define entries that operators can choose to enhance the attribute associated to the selected Lookup list.



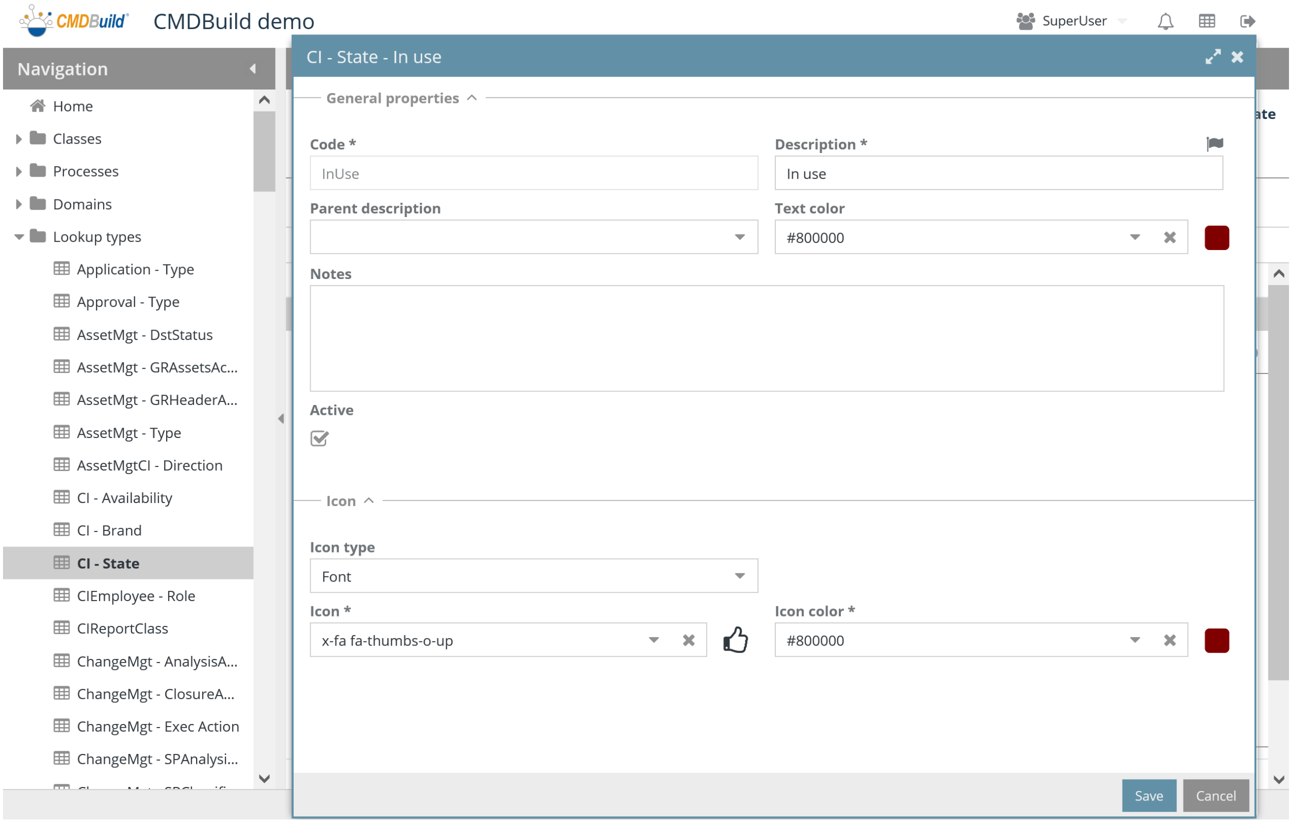
Code	Description	Text color	Active	Parent description
Available	Available	■	☑	
InUse	In use	■	☑	
Acceptance	Acceptance	■	☑	

Code: InUse
Description: In use
Parent description:
Text color: ■
Notes:
Active: ☑

There is the possibility of performing the following operations:

• add a new entry in a lookup list	+ Add value
• do searches on entries of the Lookup list	Search in grid... ✖
• edit the selected Lookup entry	✎
• display the selected Lookup entry	↗
• delete the selected Lookup entry	🗑
• disable the selected Lookup entry (i.e. set the status as non-active)	⊘

Each Lookup list requires some parameters (metadata) to be compiled, as mentioned below.



CI - State - In use

General properties

Code *: InUse

Description *: In use

Parent description:

Text color: #800000

Notes:

Active: ☑

Icon

Icon type: Font

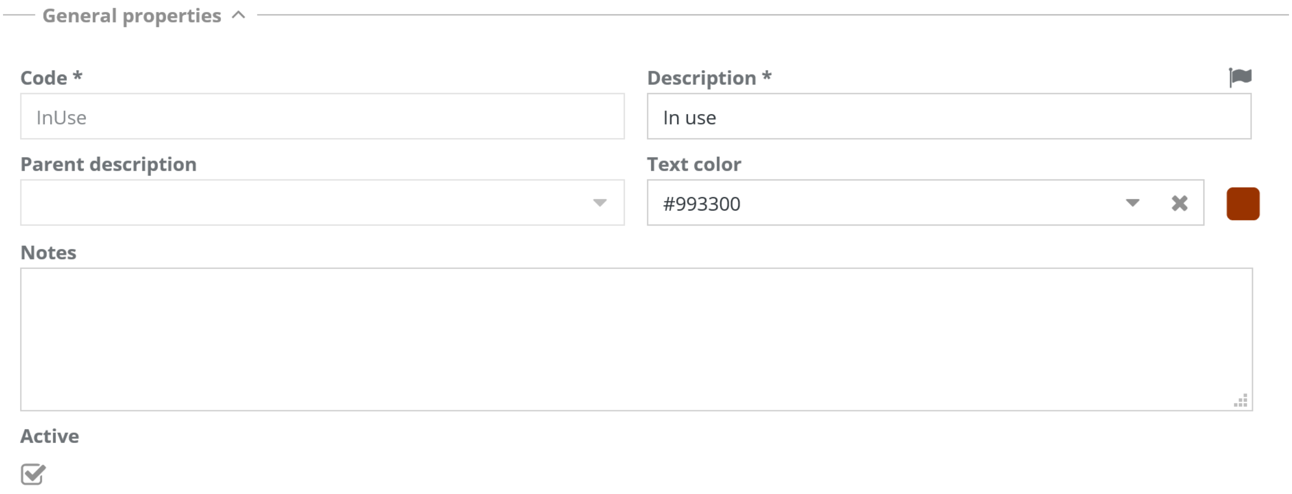
Icon *: x-fa fa-thumbs-o-up

Icon color *: #800000

10.2.1. General properties

The following information is required:

- Code: it is the univocal identification code within the selected Lookup list. It can be used from some system automatisms in order to recognize that Lookup entry (we suggest you not to edit it after defining it since you can work on the Description)
- Description: Lookup description, it appears in the multiple-choice list
- Parent description: selection of the entry in the parent Lookup list to link the current entry, in case of defining hierarchical lists
- Text color: usable to highlight interesting information (such as the status of an asset or of a process)
- Notes: free field for administrator's notes (it does not appear in the Management Module)
- Active: the Lookup entry is active and can be used



General properties

Code *: InUse

Description *: In use

Parent description:

Text color: #993300 🟥

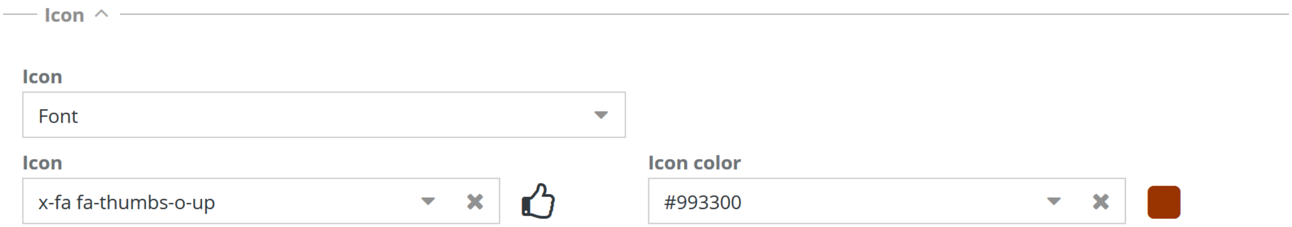
Notes:

Active: ☑

10.2.2. Icon

The following information is required:

- Type: font or picture
- Font: in case the icon is made by a font, you are required to select it among the suggested ones
- Icon color: in case the icon is made by a font, you are required to choose its color among the suggested ones
- Picture: in case the icon is made by a picture, you are required to select it in the file system in order to upload it into the system



Icon

Icon: Font

Icon: x-fa fa-thumbs-o-up 👍

Icon color: #993300 🟥

11. Views

Through the views mechanism, not-corresponding groups of cards of a class can be provided to users.

Views can be defined in three ways:

- through a filter, i.e. as subsystems of cards in the same class / process to which a filter has been applied
- through visual join among classes, with selection of attributes and filters
- through SQL query, i.e. as cards filed in various classes / processes extracted through a database function

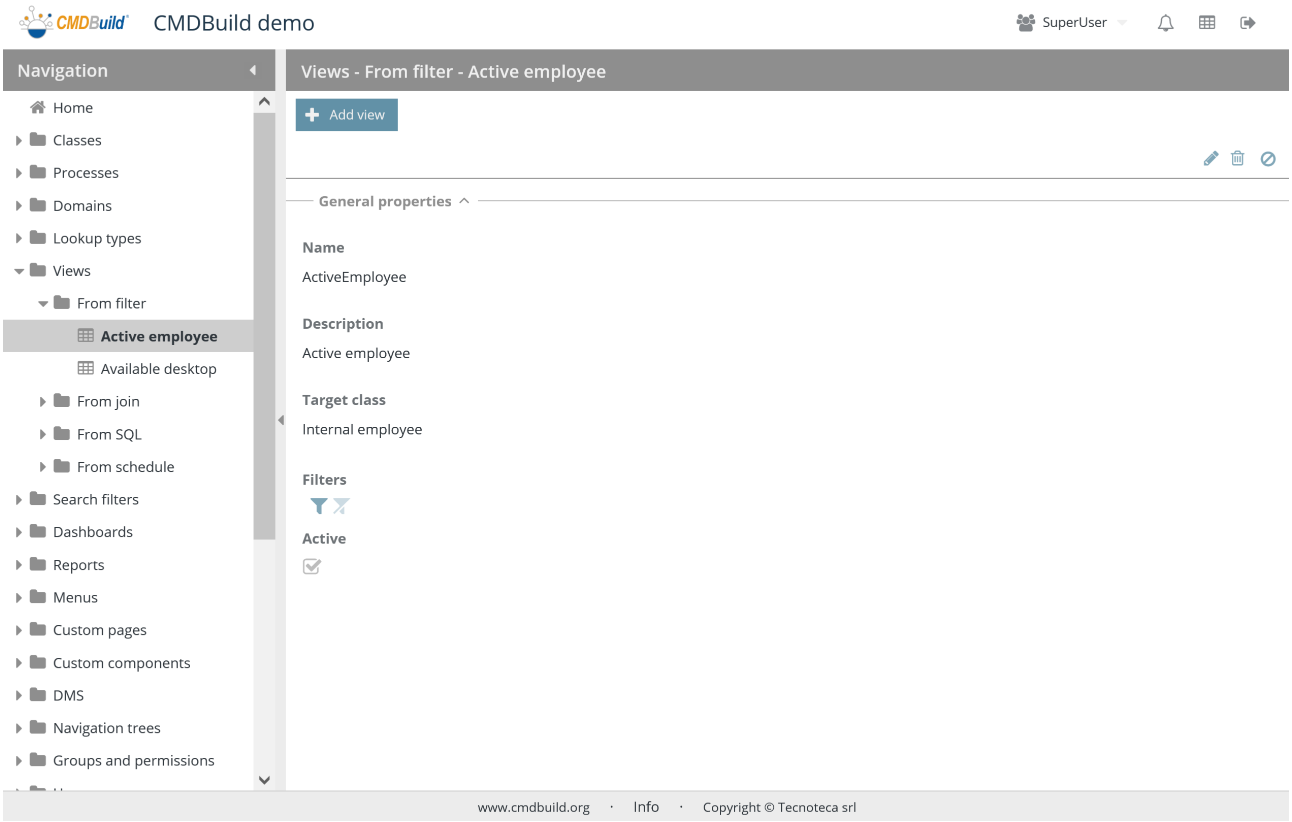
In the first case they keep their common functionalities, in the second and third ones they are Read Only.

Despite the permissions with filters on rows, that exclude the consultation of the related information, the views based on the filter, the operator can decide to remove the filter and display the complete information of the current class / process.

Finally, it is possible to define views on the Scheduling file.

11.1. Views based on filters

The function allows you to create new views based on filters and edit the old ones.



Name
ActiveEmployee

Description
Active employee

Target class
Internal employee

Filters

Active
☑

There is the possibility of performing the following operations:

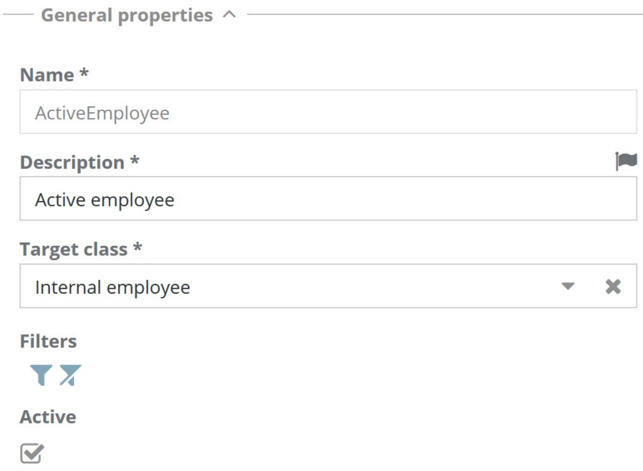
add a new view list	+ Add View
edit the selected view	✎
delete the selected view	🗑
disable the selected view (i.e. set the status as non-active)	⊘

Each view needs some parameters to be compiled (metadata), as mentioned.

11.1.1. General properties

The following information is required:

- Activity name
- Description
- Target class: original class or process
- Active: the view is active and can be used



General properties ^

Name *

ActiveEmployee

Description *

Active employee

Target class *

Internal employee

Filters

Active

☑

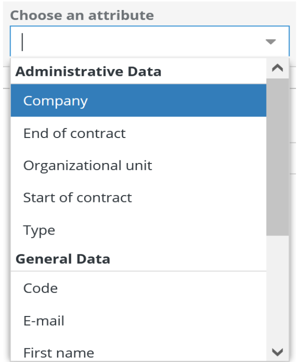
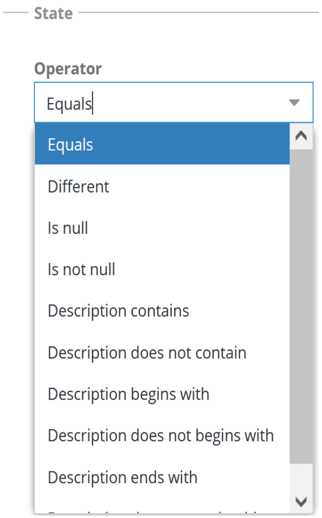
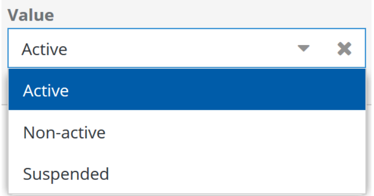
This form shares the following icons:

filter configuration	⏷
filter cancellation	⏷̸

11.1.2. Filters

The ways of setting a filter are similar to those used in the Data Management Module (User Manual chapter 8.1) to set a filter on the attributes of a class.

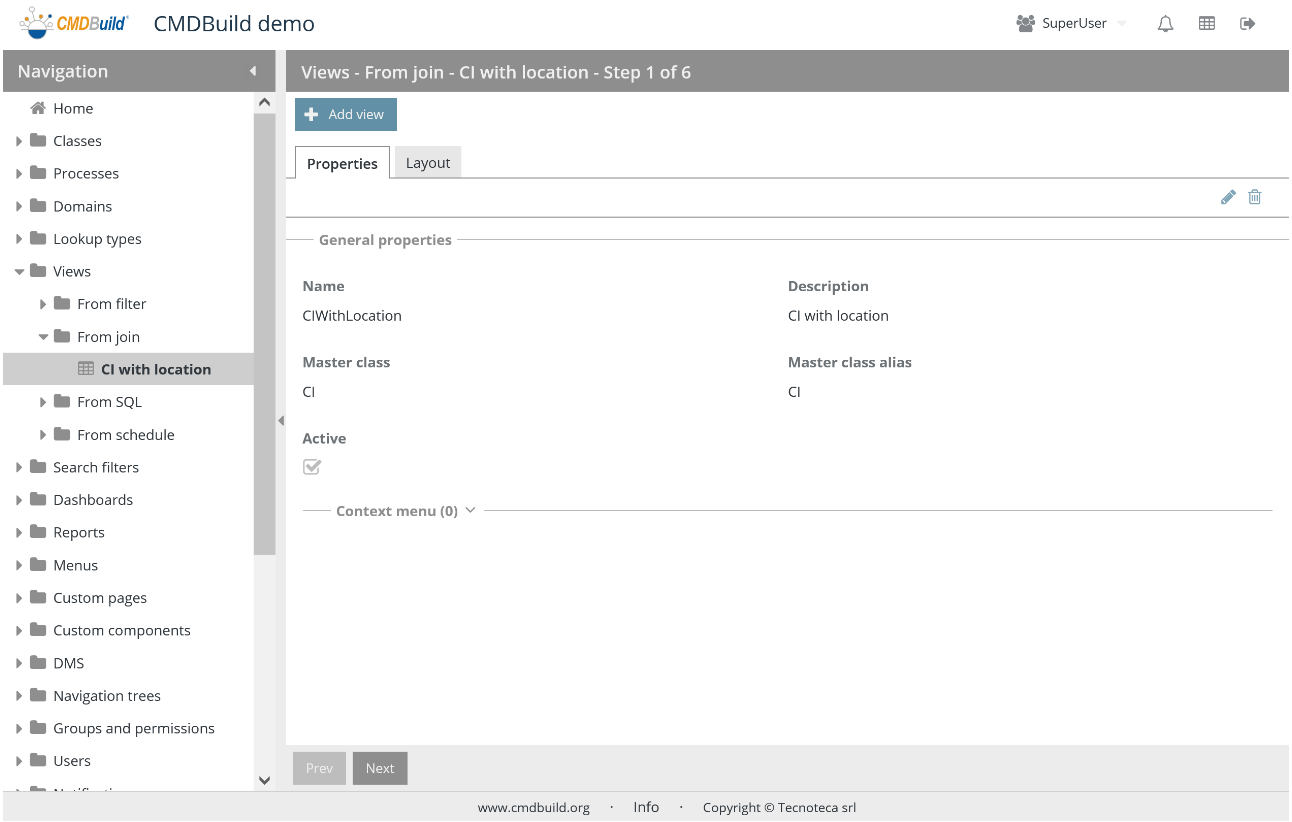
There is the possibility of performing the following operations, also many times:

• choose an attribute where the filter condition can be applied to. After this, a row with that attribute is added and you can set the filter	 Choose an attribute Administrative Data Company End of contract Organizational unit Start of contract Type General Data Code E-mail First name
• choose an operator that depends on the type of attribute	 State Operator Equals Equals Different Is null Is not null Description contains Description does not contain Description begins with Description does not begins with Description ends with
• digit (in case of strings or numbers) or choose (in case of references or lookups) a value	
• delete a set filter condition	🗑
• apply the filter	Ok
• delete the operation	Close

A filter can have more conditions. Any multiple conditions applied to the same attribute will be considered in "OR", while the conditions applied to different attributes will be considered in "AND".

11.2. Views based on join

The function allows you to create new views based on join and edit the old ones.



Views - From join - CI with location - Step 1 of 6

General properties

Name: CIWithLocation

Description: CI with location

Master class: CI

Master class alias: CI

Active: ☑

There is the possibility of performing the following operations:

• add a new view list	✚ Add View
• edit the selected view	✎
• delete the selected view	🗑

11.2.1. Properties tab

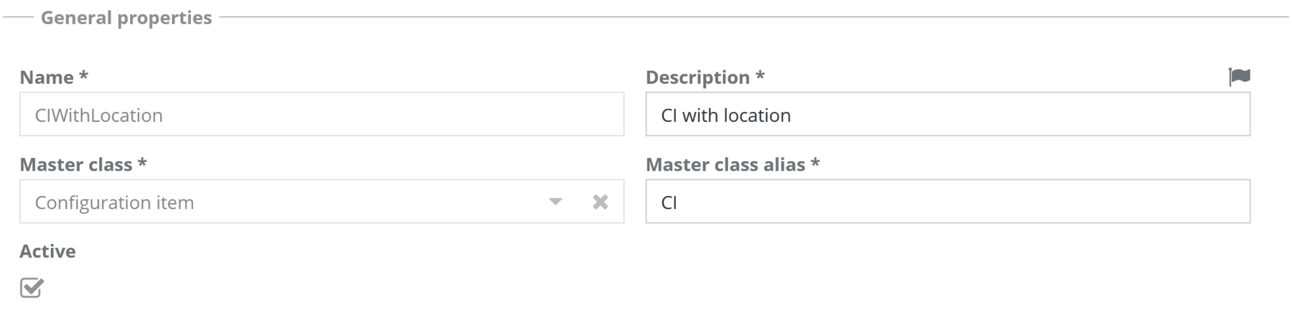
The Property TAB needs some parameters to be compiled through the wizard, as mentioned below.

<u>Step 1</u>

General properties

The following information is required:

- Activity name
- Description
- Master class: original class
- Master class alias: original alias class
- Active: the view is active and can be used



General properties

Name *: CIWithLocation

Description *: CI with location

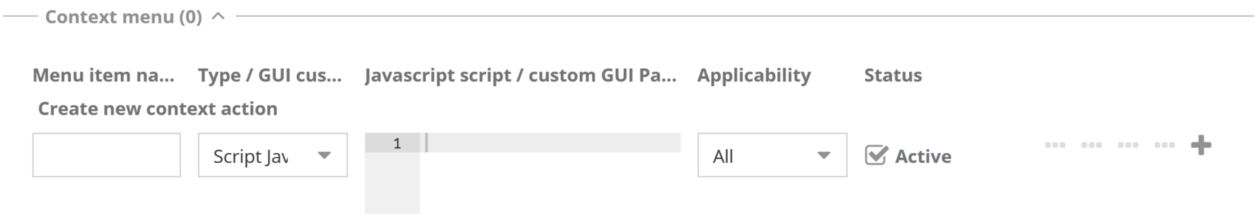
Master class *: Configuration item

Master class alias *: CI

Active: ☑

Contextual menu

You can configure a contextual menu with the same configuration modes of contextual menus on classes.



Context menu (0)

Menu item na...	Type / GUI cus...	Javascript script / custom GUI Pa...	Applicability	Status
Create new context action				
	Script Jav	1	All	☑ Active

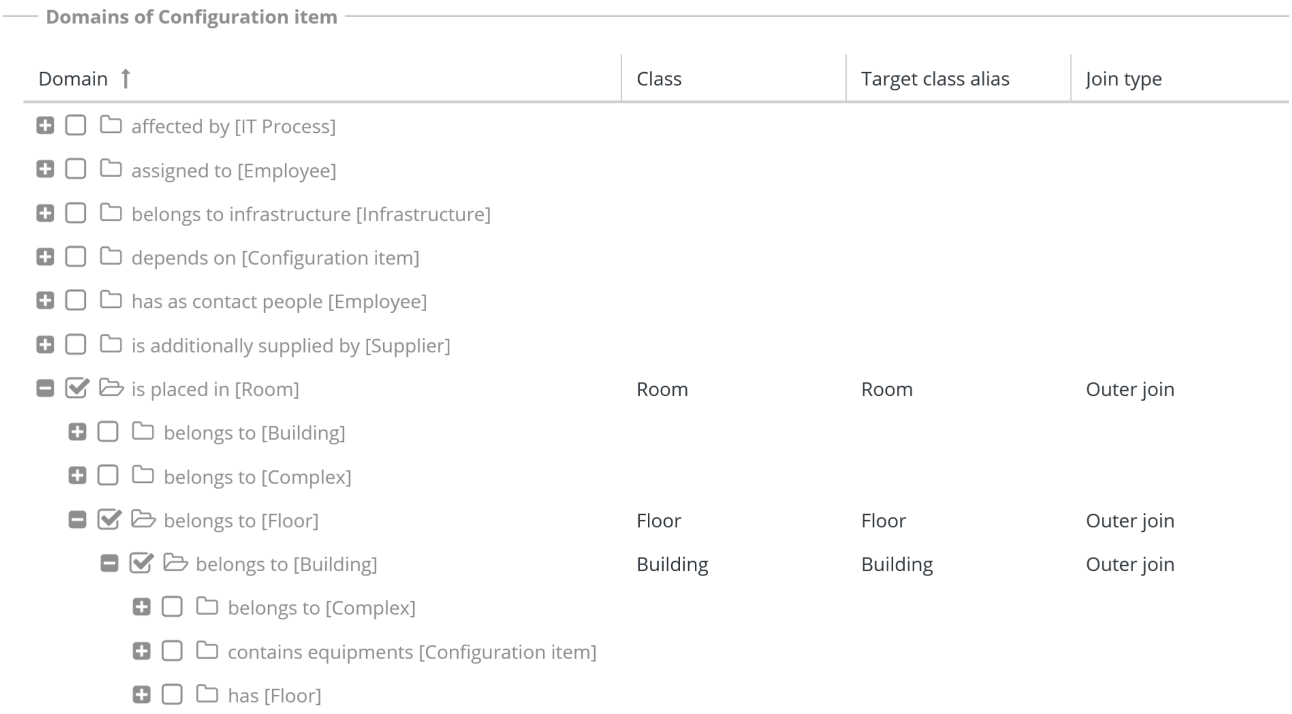
<u>Step 2</u>

Join definition

You can select the join modes among the classes through the class tree reached by "domains" 1:N starting from the root class.

For each linked class you can define an alias and the type of join (inner join or outer join).

In the example, starting from the CI class joins with Room have been defined, and then Floor and Building.



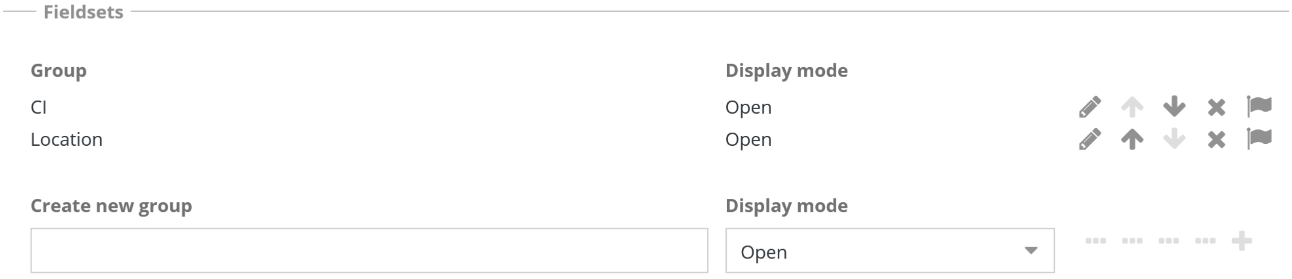
Domains of Configuration item

Domain ↑	Class	Target class alias	Join type
⊞ ☐ affected by [IT Process]			
⊞ ☐ assigned to [Employee]			
⊞ ☐ belongs to infrastructure [Infrastructure]			
⊞ ☐ depends on [Configuration item]			
⊞ ☐ has as contact people [Employee]			
⊞ ☐ is additionally supplied by [Supplier]			
⊟ ☑ is placed in [Room]	Room	Room	Outer join
⊞ ☐ belongs to [Building]			
⊞ ☐ belongs to [Complex]			
⊟ ☑ belongs to [Floor]	Floor	Floor	Outer join
⊟ ☑ belongs to [Building]	Building	Building	Outer join
⊞ ☐ belongs to [Complex]			
⊞ ☐ contains equipments [Configuration item]			
⊞ ☐ has [Floor]			

Step 3

Definition of attributes groups (fieldset)

You can define filedsets to group class attributes in joins.



Fieldsets

Group	Display mode
CI	Open
Location	Open

Create new group:

Display mode: Open

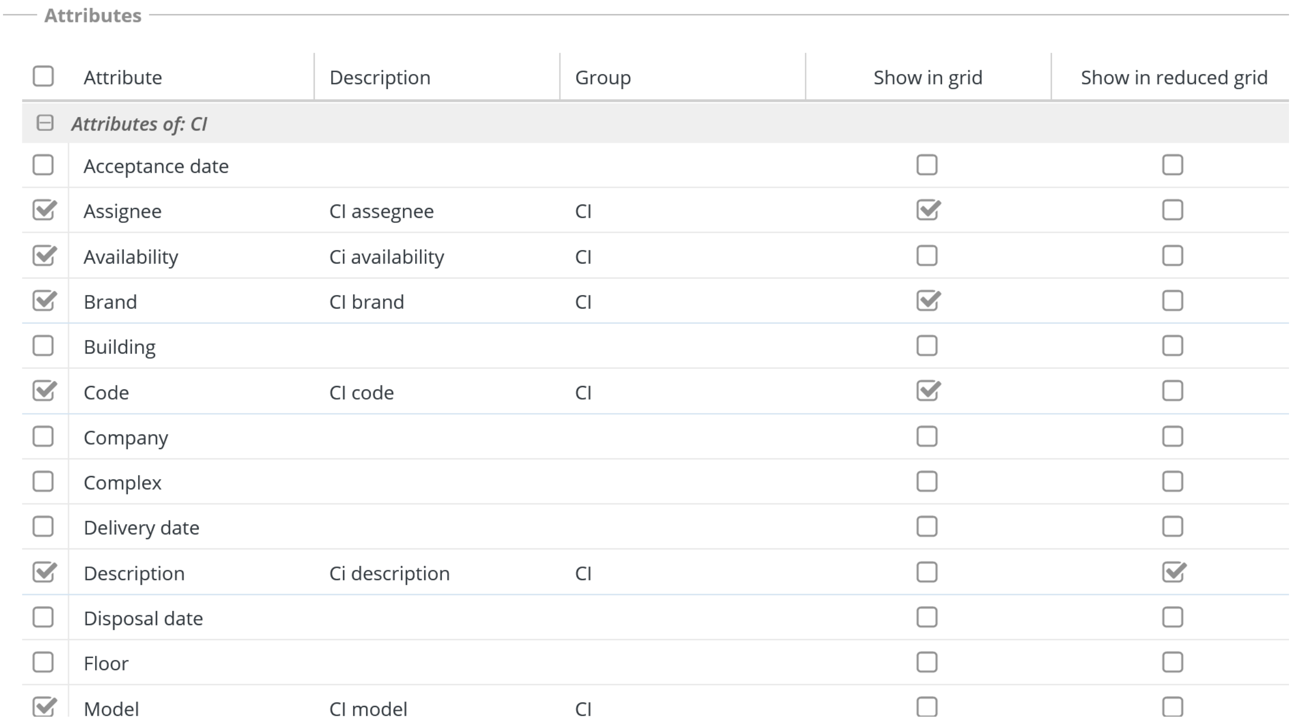
Please refer to the process used for grouping attributes in the classes.

Step 4

Attribute selection

You can select class attributes in joins and make them available in the view.

For each attribute you can set an alias (Description), the relating fieldset, and you can decide if displaying it in the standard grid and if displaying it in the reduced grid (mobile interface).



Attributes

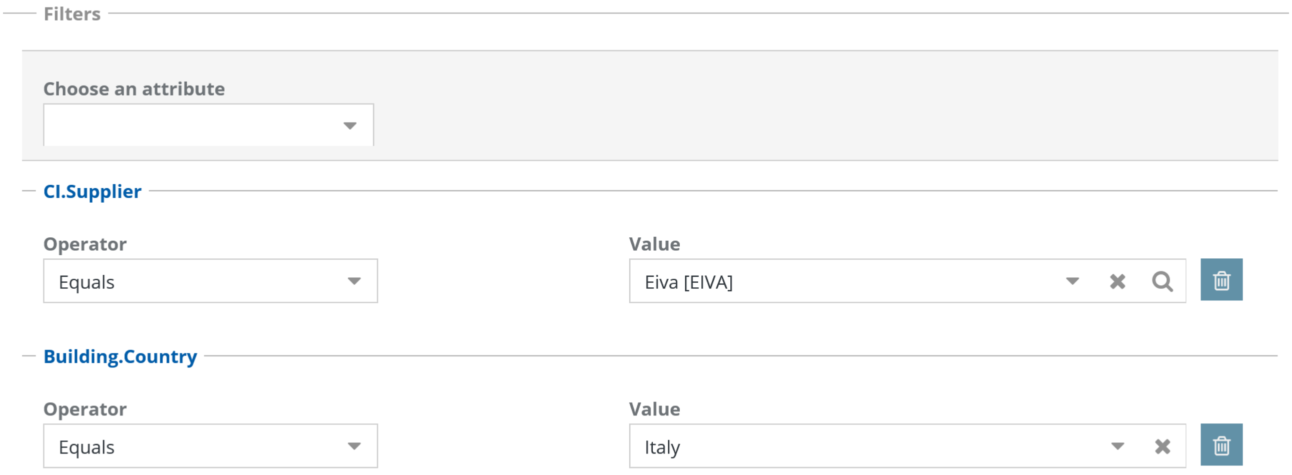
□	Attribute	Description	Group	Show in grid	Show in reduced grid
⊟	*Attributes of: CI*				
□	Acceptance date			□	□
☑	Assignee	CI assegnee	CI	☑	□
☑	Availability	Ci availability	CI	□	□
☑	Brand	CI brand	CI	☑	□
□	Building			□	□
☑	Code	CI code	CI	☑	□
□	Company			□	□
□	Complex			□	□
□	Delivery date			□	□
☑	Description	Ci description	CI	□	☑
□	Disposal date			□	□
□	Floor			□	□
☑	Model	CI model	CI	□	□

Step 5

Filters

You can define filters on selected attributes, in order to restrict the view output.

Please refer to the config of filter views.



Filters

Choose an attribute

CI.Supplier

Operator: Equals

Value: Eiva [EIVA]

Building.Country

Operator: Equals

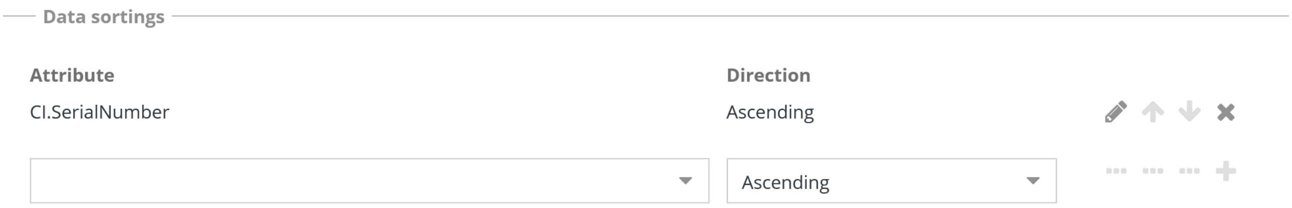
Value: Italy

Step 6

Data organization

You can define the organization of view data.

Please refer to the processes of class data organization.



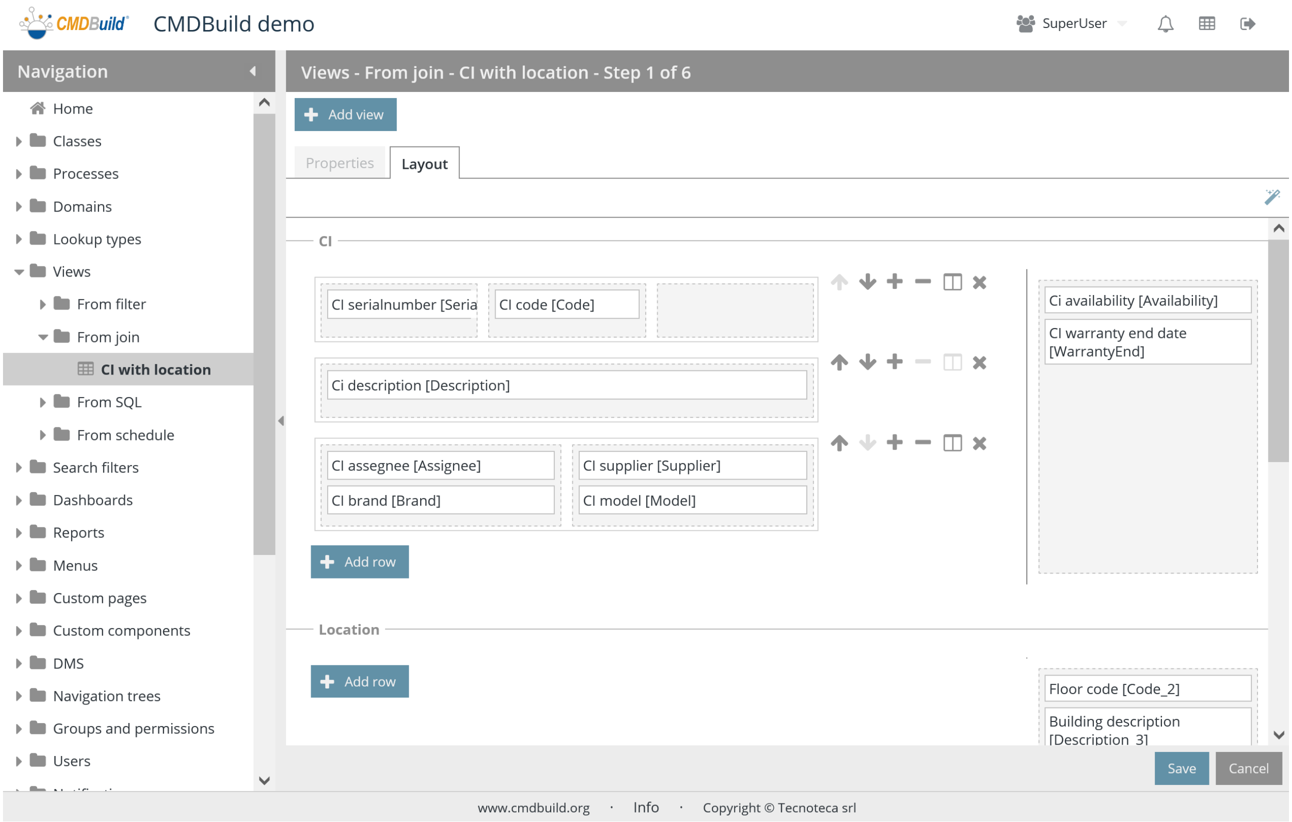
Data sortings

Attribute	Direction
CI.SerialNumber	Ascending
	Ascending

11.2.2. Layout TAB

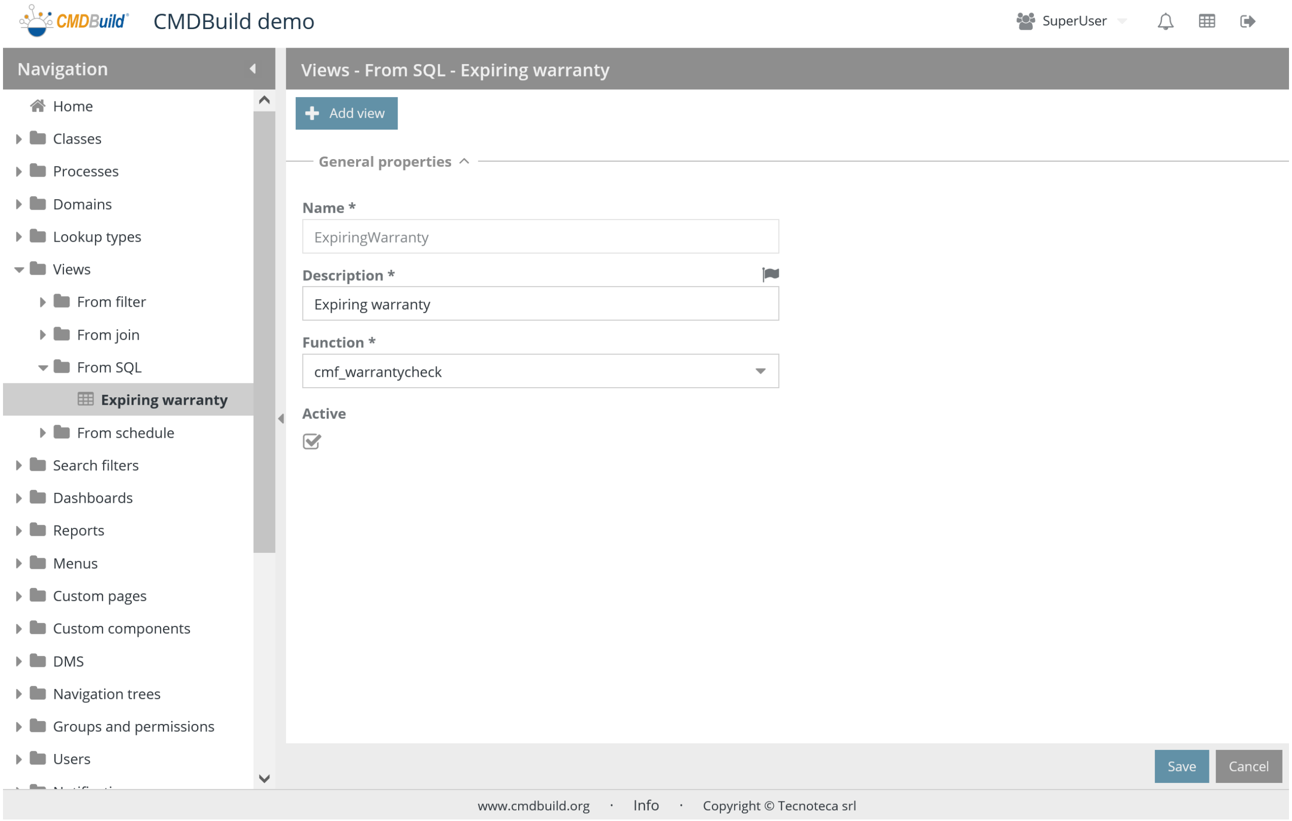
Also for views from join you can define a custom layout, in terms of number of rows, number and width of columns and positioning of attributes in the resulting cells.

Please refer to the process of defining the class layout.



11.3. Views based on SQL

The function allows you to create new views based on SQL and edit the old ones.



Views - From SQL - Expiring warranty

Name * ExpiringWarranty

Description * Expiring warranty

Function * cmf_warrantycheck

Active ☑

There is the possibility of performing the following operations:

• create a new view	+ Add View
• edit the selected view	✎
• delete the selected view	🗑
• disable the selected view (i.e. set the status as non-active)	⊘

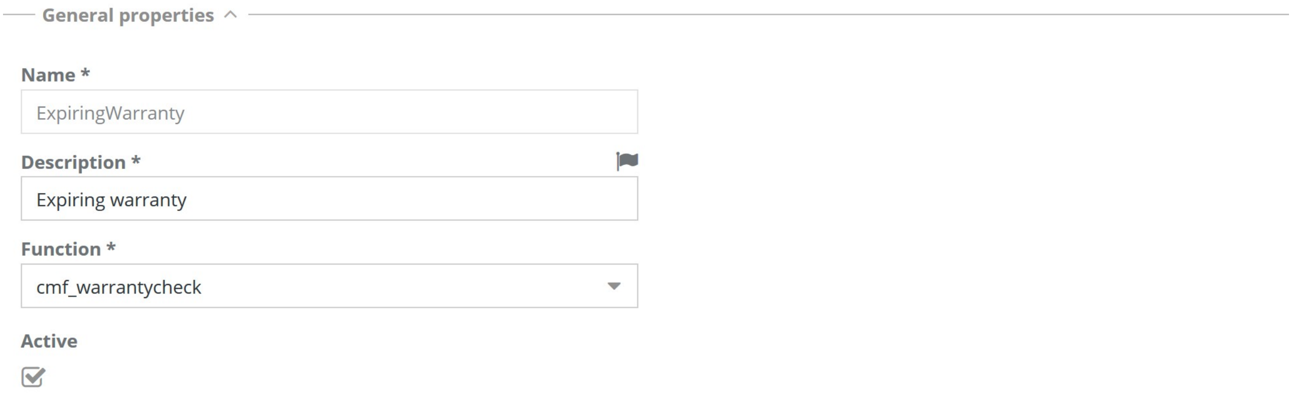
Each view needs some parameters to be compiled (metadata), as mentioned.

11.3.1. General properties

The following information is required:

- Activity name
- Description

- Function: SQL function defined in the PostgreSQL database in order to extract the desired cards from one or more classes (compatibly with the restrictions described at the following paragraph)
- Active: the view is active and can be used



General properties

Name *

ExpiringWarranty

Description *

Expiring warranty

Function *

cmf_warrantycheck

Active

☑

11.3.2. Definition of the data source (PostgreSQL function)

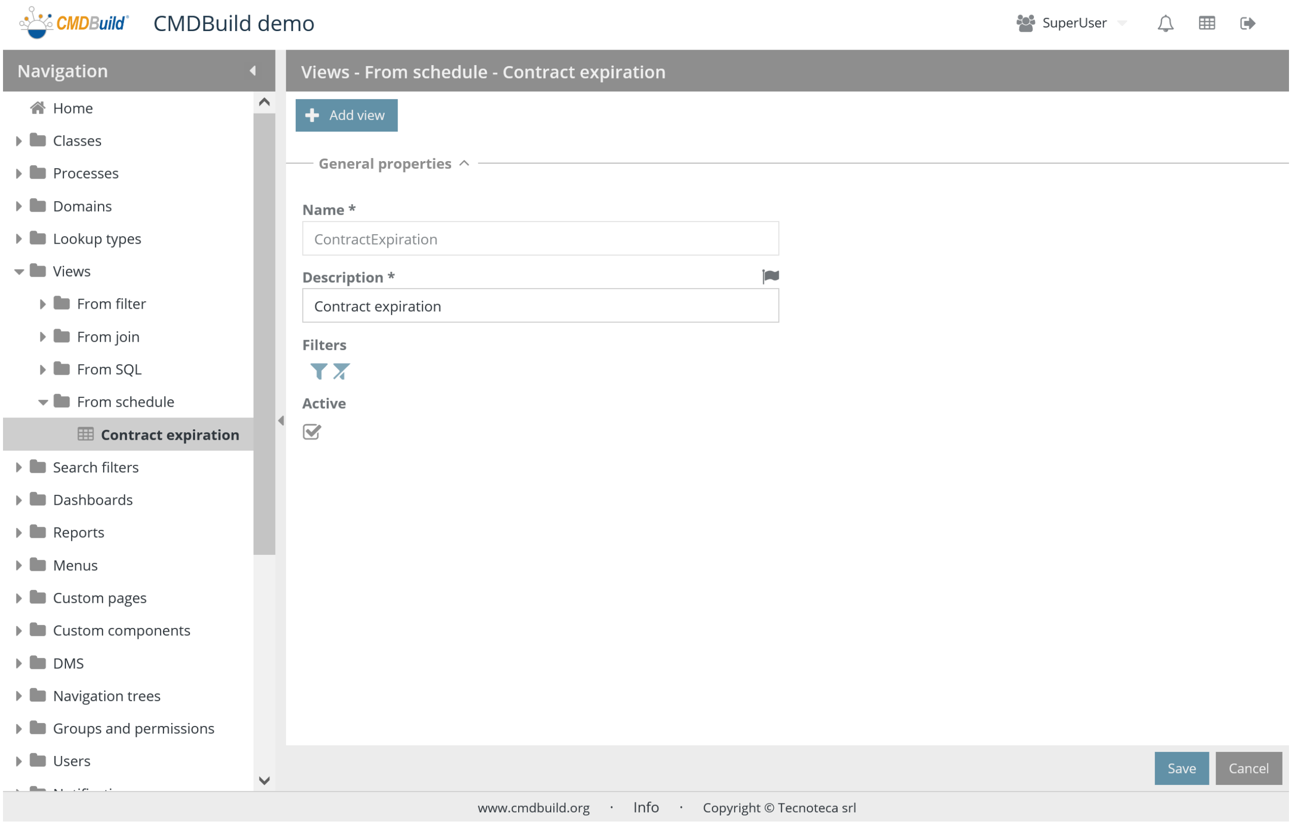
The SQL function, defined in the PostgreSQL database in order to extract the desired cards, has to compatible with the following restrictions:

- the function must report the comment "TYPE: function”
- the definition of the function has to include the input and output parameters, clearly showing the name of each of them
- the input and output parameters have to be chosen among the following ones: “character varying”, “boolean”, “integer”, “bigint”, “numeric”, “double precision”, “date”, “time”, “timestamp”, “text”
- in case that the function becomes more "tuple" than the output values, you should use the syntax "Returns setof record”

At the end of the operation, you have to perform the function "Setup” > Server management > Clear Cache, or to reboot Tomcat.

11.4. Views on the timetable

This feature allows you to create specific views on the Scheduling archive and edit the old ones.



Views - From schedule - Contract expiration

General properties

Name * ContractExpiration

Description * Contract expiration

Filters

Active ☑

The configuration modes are similar to the views defined through the filter, except for the fact that the current file is the Scheduling one.

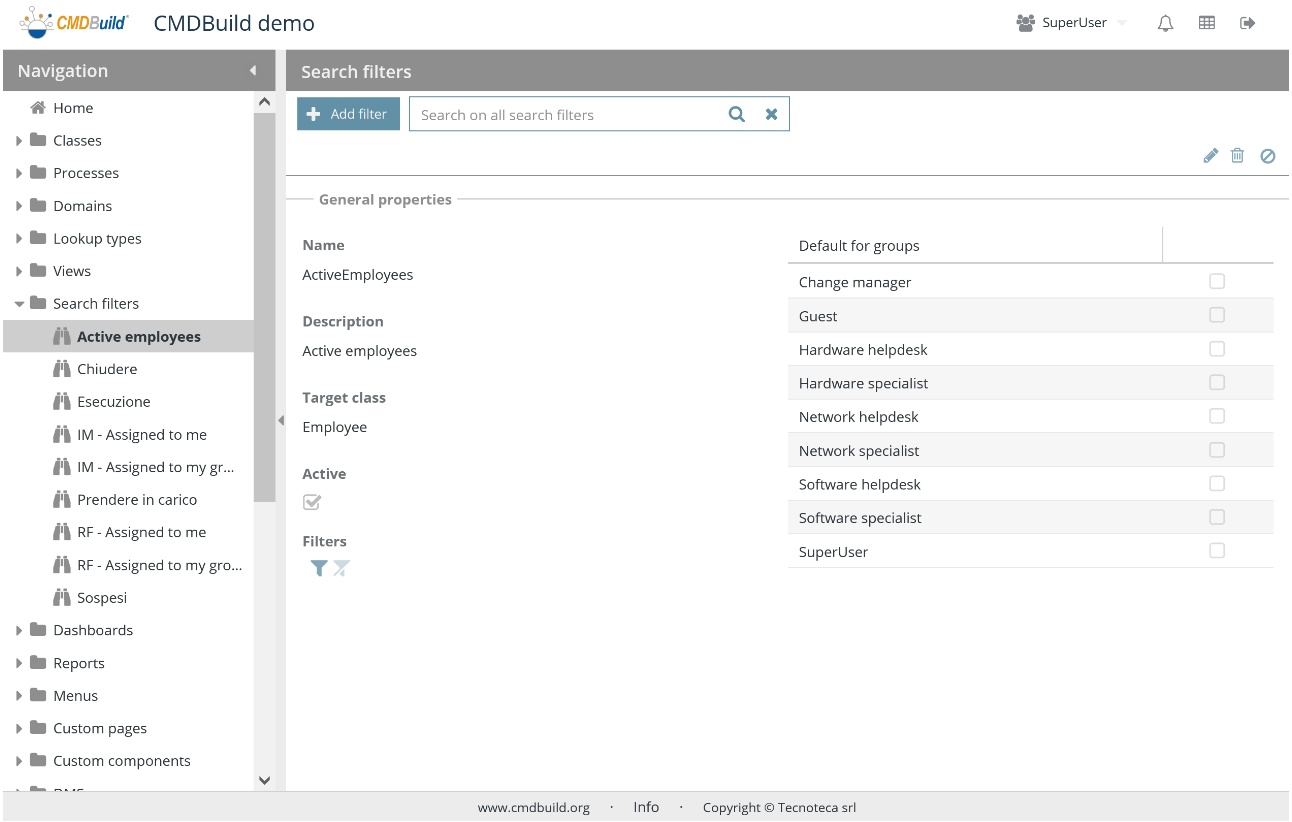
12. Search filters

Search filters can be used by operators in the data Management Module in order to display only the cards of a CMDBuild class meeting the set criteria.

They can be configured by the same operators in the Management Module using them in that moment or saving them in order to find them again at the following steps of the application, as described in the User Manual. Otherwise they can be configured by the administrator in the Administrator Module and made available to operators, as described in the following paragraph.

12.1. Properties

The function allows you to create new filters and edit the properties of the old ones.



CMDBuild demo SuperUser

Navigation

- Home
- Classes
- Processes
- Domains
- Lookup types
- Views
- Search filters
  - Active employees
  - Chiudere
  - Esecuzione
  - IM - Assigned to me
  - IM - Assigned to my gr...
  - Prendere in carico
  - RF - Assigned to me
  - RF - Assigned to my gro...
  - Sospesi
- Dashboards
- Reports
- Menus
- Custom pages
- Custom components

Search filters

+ Add filter | Search on all search filters

General properties

Name: ActiveEmployees

Description: Active employees

Target class: Employee

Active: ☑

Filters

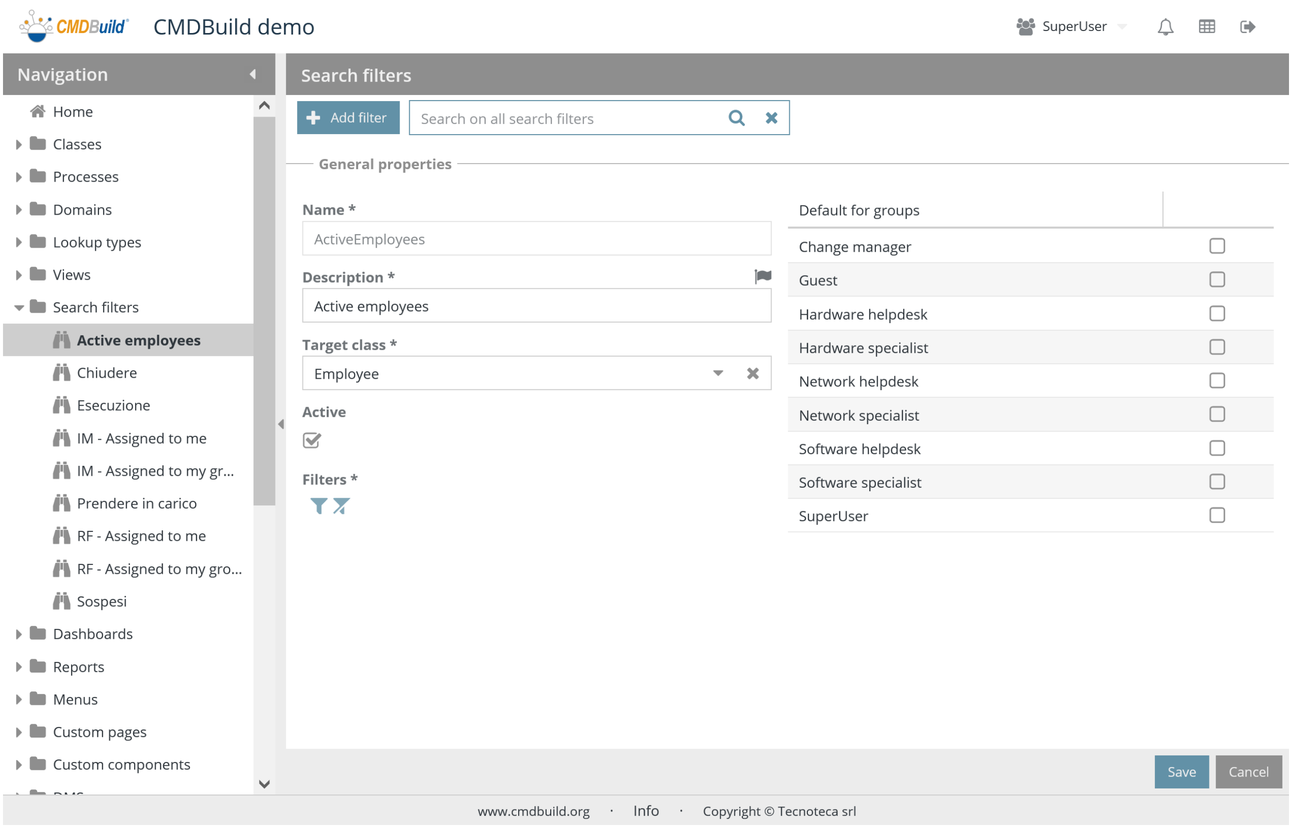
Default for groups	
Change manager	□
Guest	□
Hardware helpdesk	□
Hardware specialist	□
Network helpdesk	□
Network specialist	□
Software helpdesk	□
Software specialist	□
SuperUser	□

www.cmdbuild.org · Info · Copyright © Tecnoteca srl

There is the possibility of performing the following operations:

• create a new filter	+ Add filter
• do searches on the configured filters	Search on all search filters 🔍 ✖
• edit the selected filter	✎

• delete the selected filter	🗑
• disable the selected filter (i.e. set the status as non-active)	⊘



General properties

Name *: ActiveEmployees

Description *: Active employees

Target class *: Employee

Active: ☑

Filters *

Default for groups	
Change manager	□
Guest	□
Hardware helpdesk	□
Hardware specialist	□
Network helpdesk	□
Network specialist	□
Software helpdesk	□
Software specialist	□
SuperUser	□

A new search filter can not be used if the access permissions are not defined, specifying which user group(s) can use them.

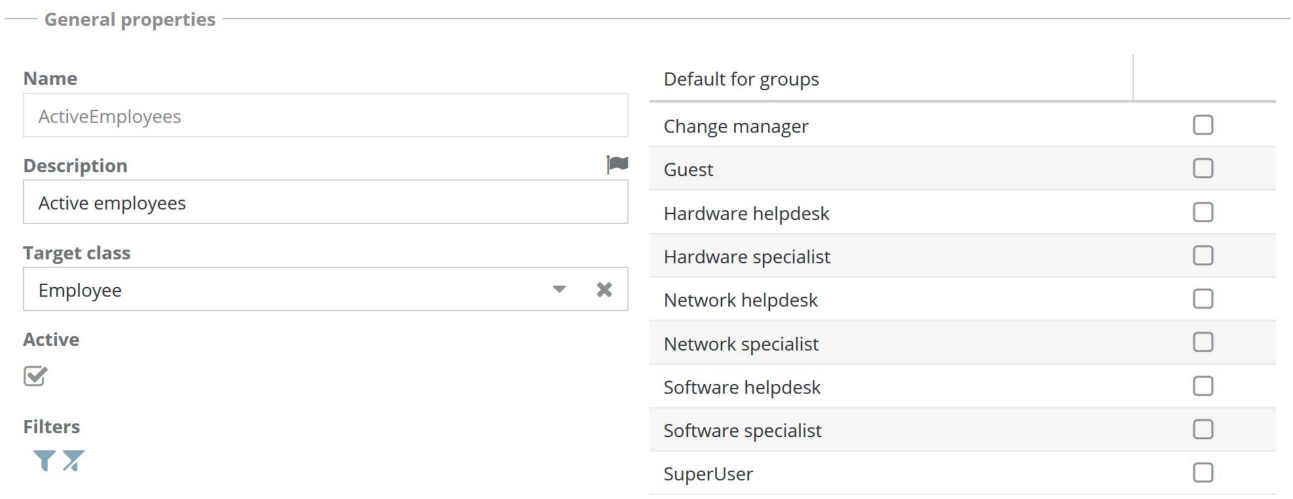
Such operation should be carried out by using the function "Users and Groups" / “Permissions” / “Search filters”.

Each filter needs some parameters to be compiled (metadata), as mentioned.

12.1.1. General properties

The following information is required:

- Name: name of the filter
- Description: description of the filter
- Target class: class which the filter can be applied to
- “Default for groups”: it allows you to specify that the current filter is applied as default filter onto the above mentioned class for the selected groups (see also the TAB "Default filters" of Users and Groups)
- Active: the filter is active and can be used



General properties

Name

ActiveEmployees

Description

Active employees

Target class

Employee

Active

☑

Filters

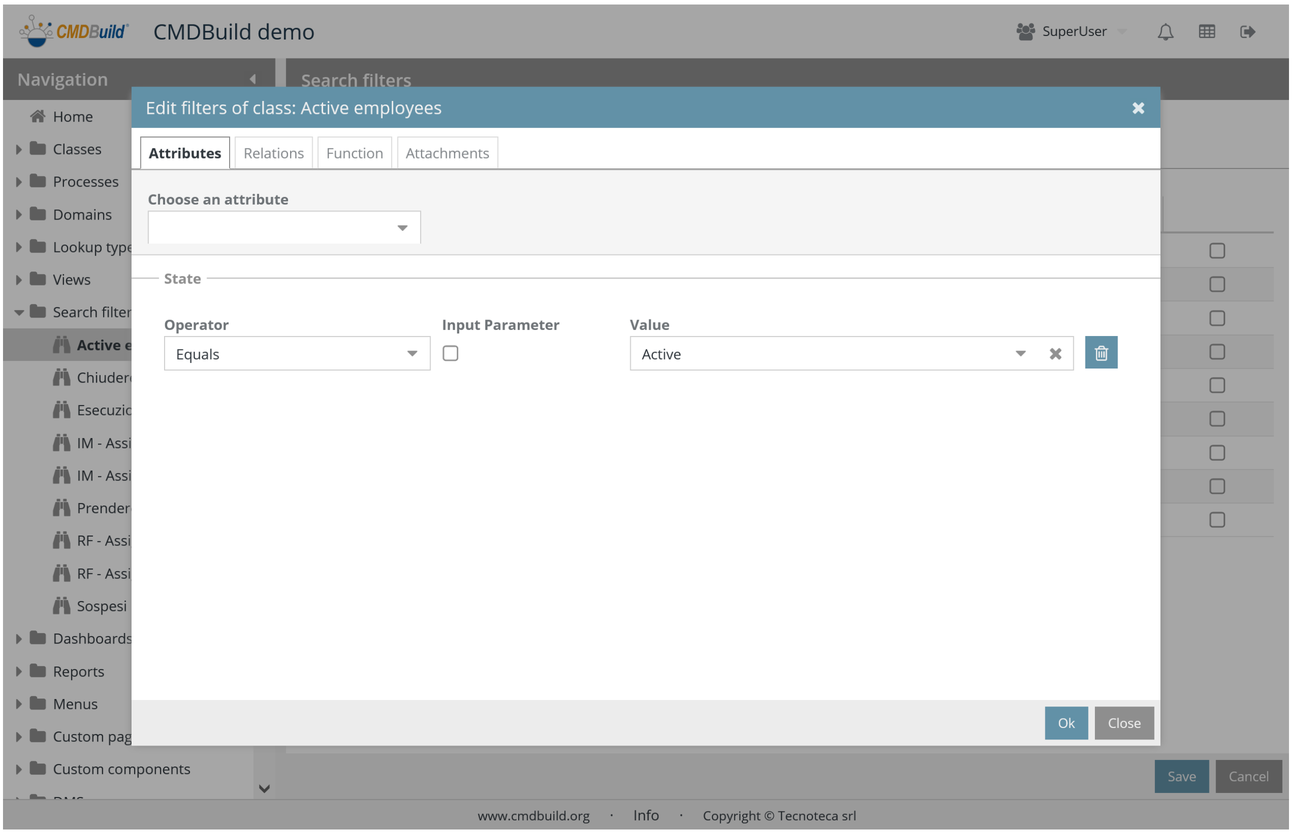
Default for groups	
Change manager	□
Guest	□
Hardware helpdesk	□
Hardware specialist	□
Network helpdesk	□
Network specialist	□
Software helpdesk	□
Software specialist	□
SuperUser	□

This form shares the following icons:

• configuration of the current filter criteria	▼
• filter cancellation	✕

Below you will find four screenshots that describe the user interface for the configuration of search filters regarding to:

- filter on attributes
- filter on relations
- filter through SQL function
- filter on attachments through search into the document (if allowed) and through search on metadata of documents

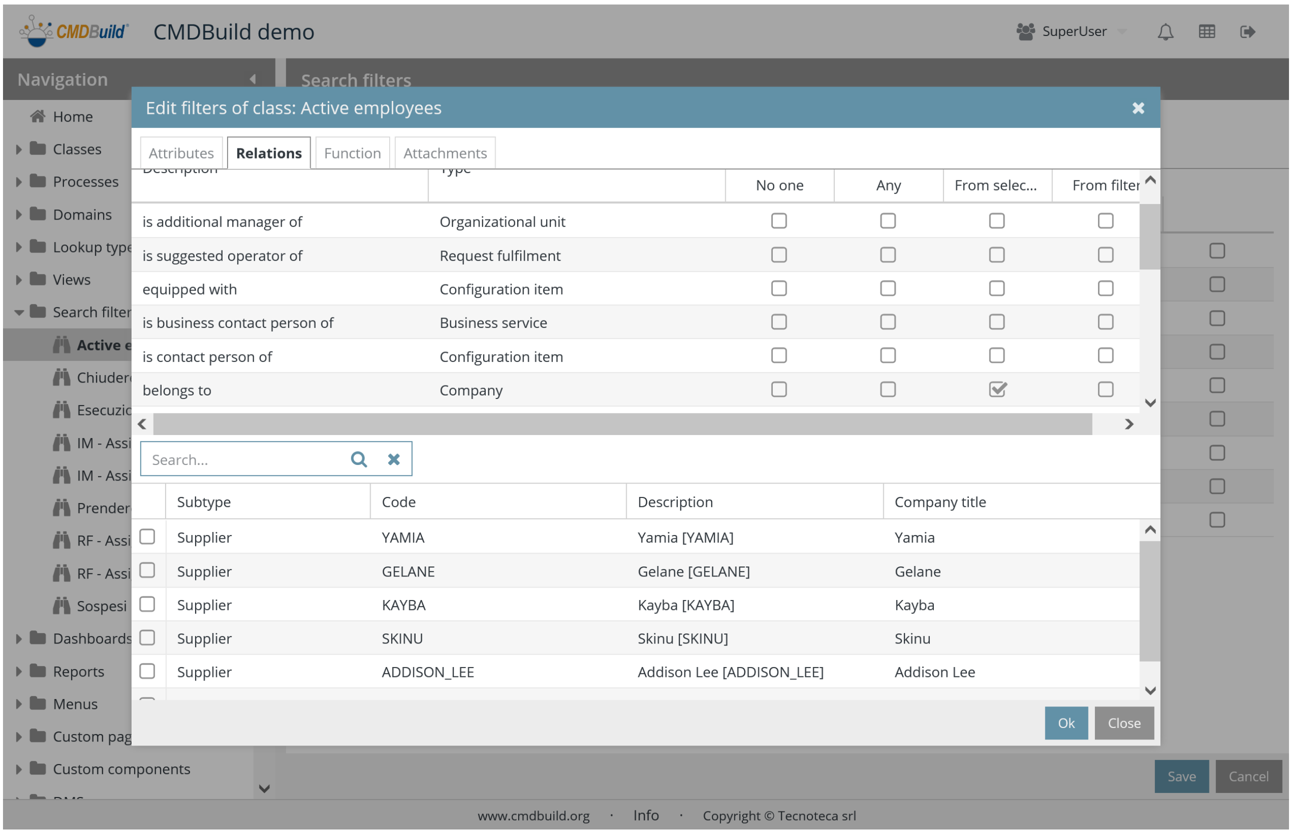


Edit filters of class: Active employees

Choose an attribute

State

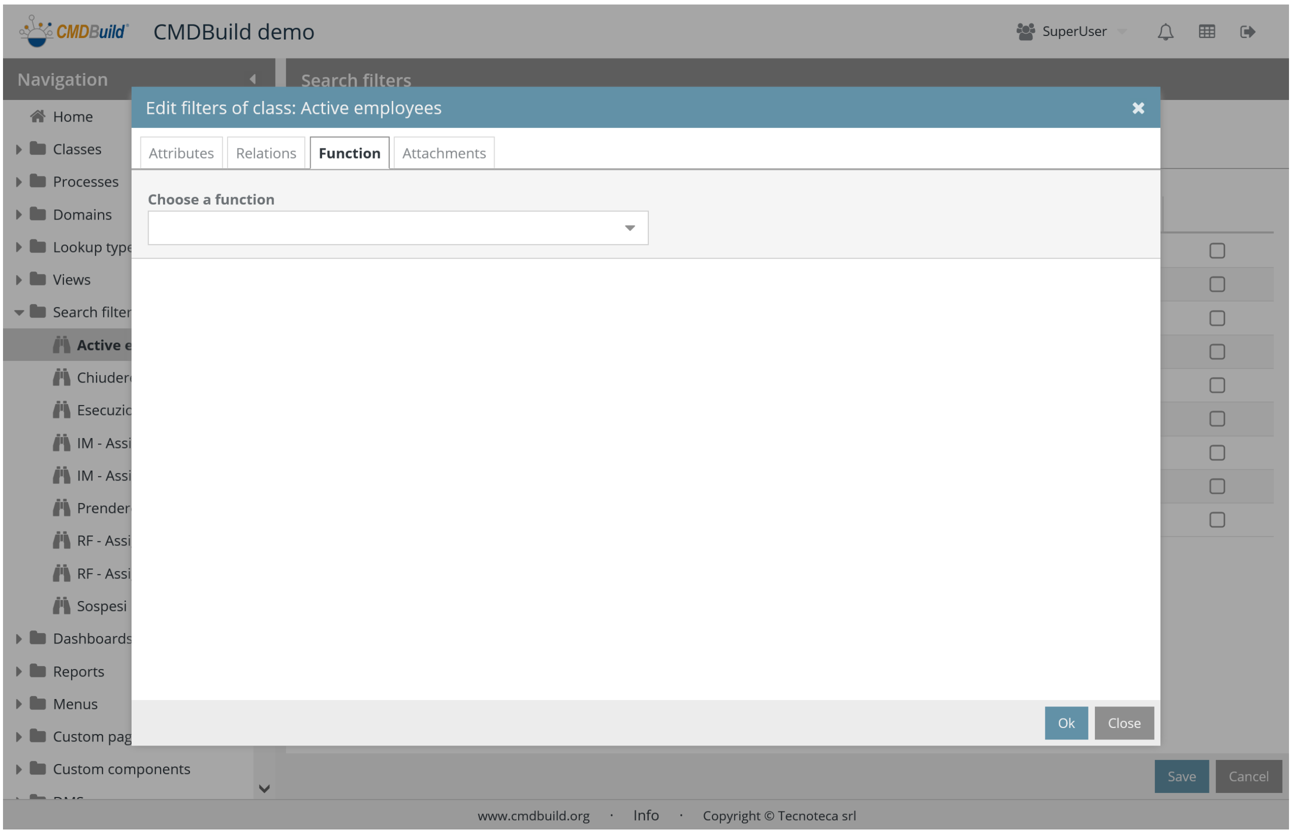
Operator	Input Parameter	Value
Equals	□	Active



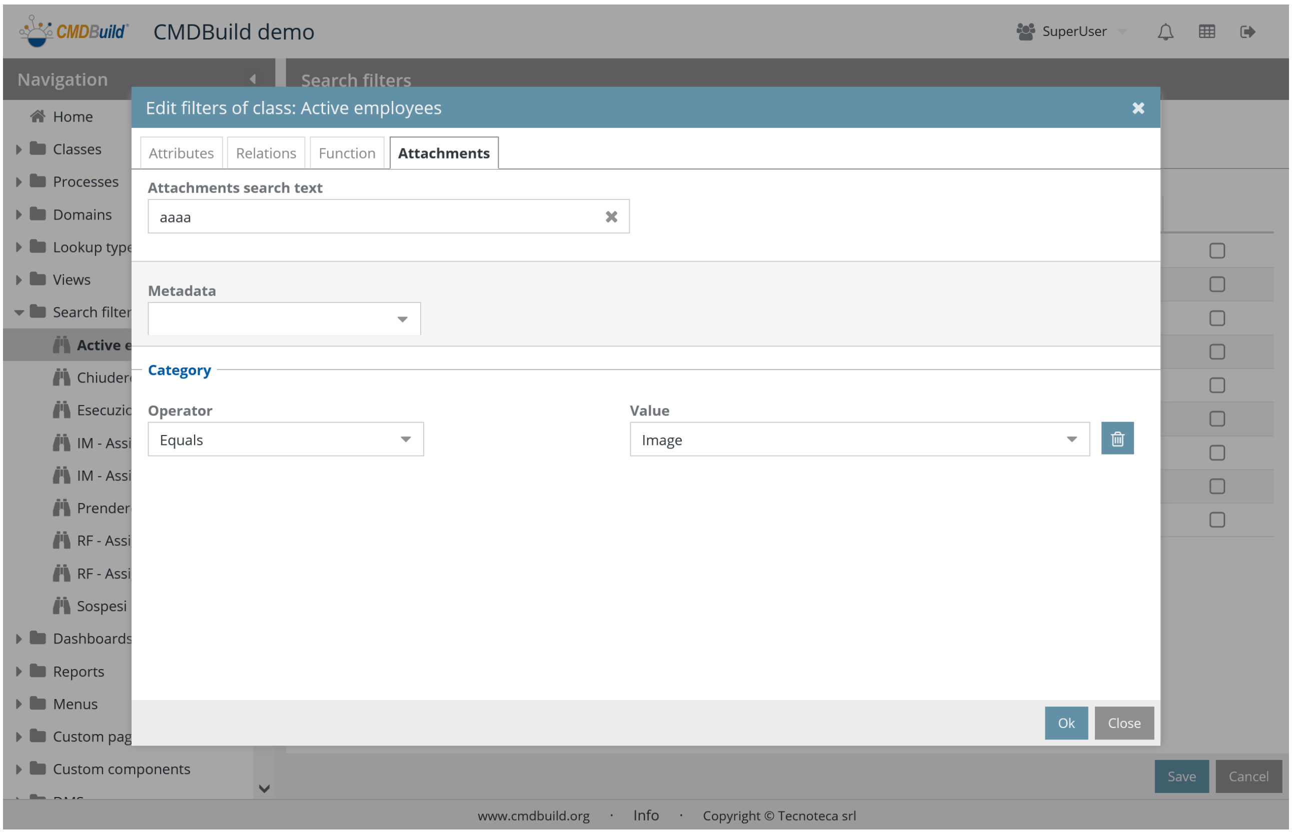
Edit filters of class: Active employees

Description	Type	No one	Any	From selec...	From filter
is additional manager of	Organizational unit	□	□	□	□
is suggested operator of	Request fulfilment	□	□	□	□
equipped with	Configuration item	□	□	□	□
is business contact person of	Business service	□	□	□	□
is contact person of	Configuration item	□	□	□	□
belongs to	Company	□	□	☑	□

	Subtype	Code	Description	Company title
□	Supplier	YAMIA	Yamia [YAMIA]	Yamia
□	Supplier	GELANE	Gelane [GELANE]	Gelane
□	Supplier	KAYBA	Kayba [KAYBA]	Kayba
□	Supplier	SKINU	Skinu [SKINU]	Skinu
□	Supplier	ADDISON_LEE	Addison Lee [ADDISON_LEE]	Addison Lee



Choose a function



Attachments search text: aaaa

Metadata

Category

Operator: Equals

Value: Image

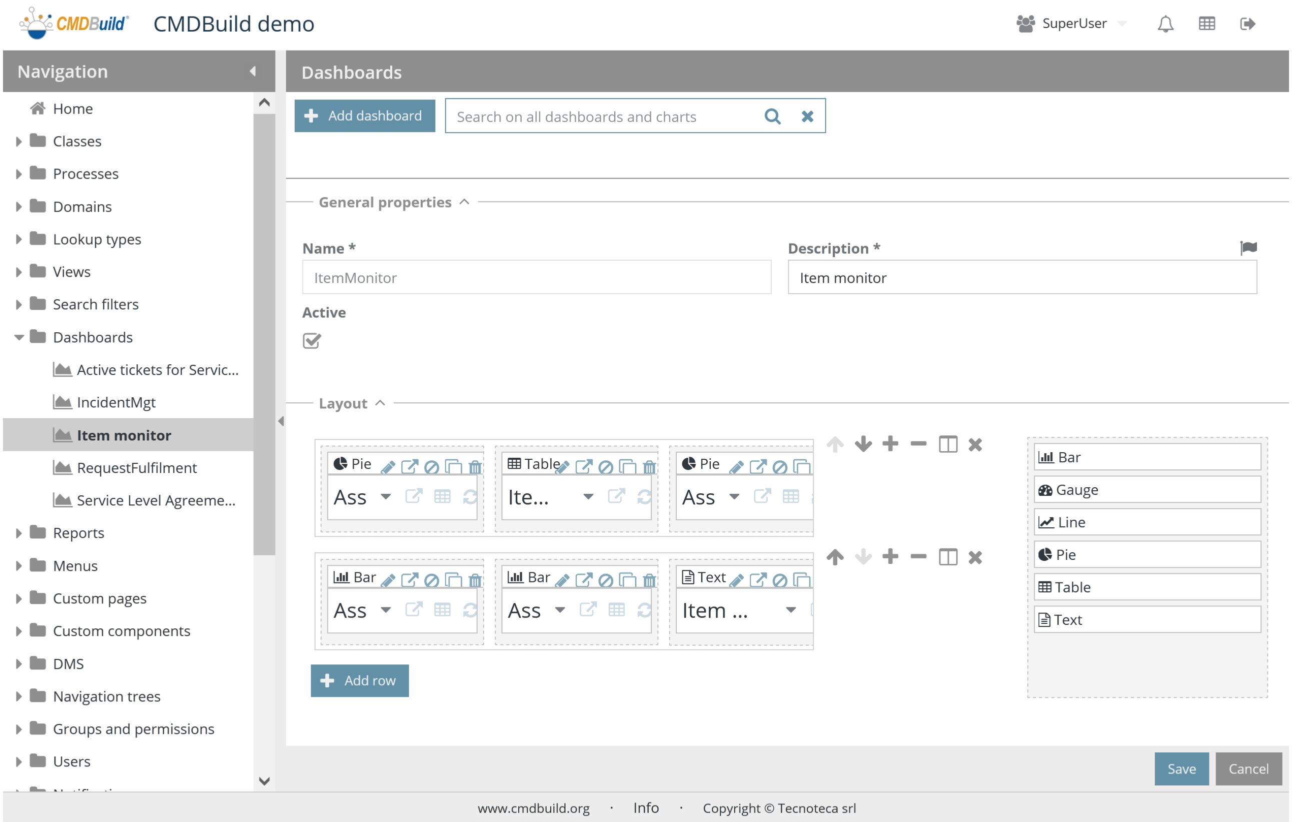
13. Dashboard

CMDBuild allows you to configure dashboard pages, which singularly contain charts of different typologies (bar, gauge, line, pie), data and texts tables; then they will be consulted in the Management Module.

Every dashboard can be addressed to a different typology of aspects which should be controlled: asset classifications (age, frequency, breakdown, use), service desk performances (SLA), economic aspects, etc.

For each dashboard you can define a custom layout by specifying the number of rows and columns for the page subdivision and the width of each resulting cell.

The data shown in the charts are extracted from the system performing suitable PostgreSQL functions defined by the administrator according to a particular template described below.



General properties

Name *: ItemMonitor

Description *: Item monitor

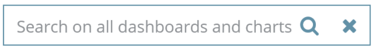
Active: ☑

13.1. Properties

Using this function you can configure new dashboard or edit the properties of the previous ones.

There is the possibility of performing the following operations:

• create a new custom page	✚ Add dashboard

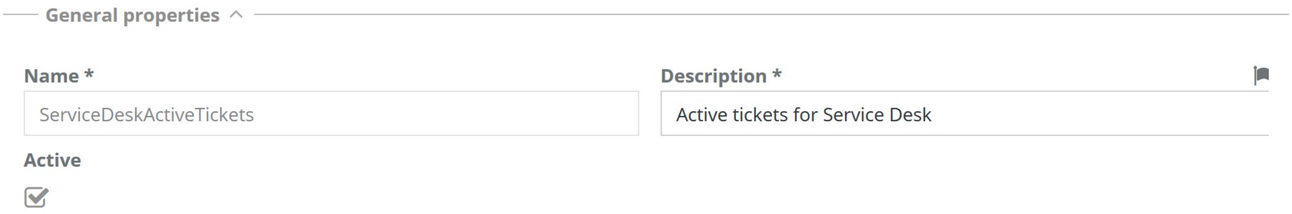
• do searched on the configured dashboard	 Search on all dashboards and charts 🔍 ✖
• edit the selected dashboard	✎
• delete the selected dashboard	🗑
• disable the selected dashboard (i.e. set the status as non-active)	⊘

The configuration of each dashboard needs some parameters to be compiled (metadata), the layout definition and the configuration of each element of the dashboard, as mentioned below.

13.1.1. General properties

The following information is required:

- Name: dashboard name
- Description: dashboard description
- Active: the dashboard is active and can be used



General properties ^

Name *
ServiceDeskActiveTickets

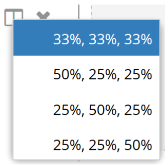
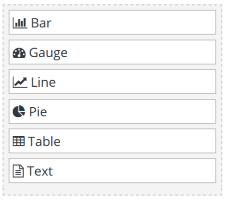
Description *
Active tickets for Service Desk

Active
☑

13.1.2. Layout

To define a custom layout and the configuration of each element of the dashboard, there are the following possibilities:

• create a new row	✚ Add row
• remove a row	✖
• move a row up	↑
• move a row down	↓
• add a new column	+
• remove a column	−

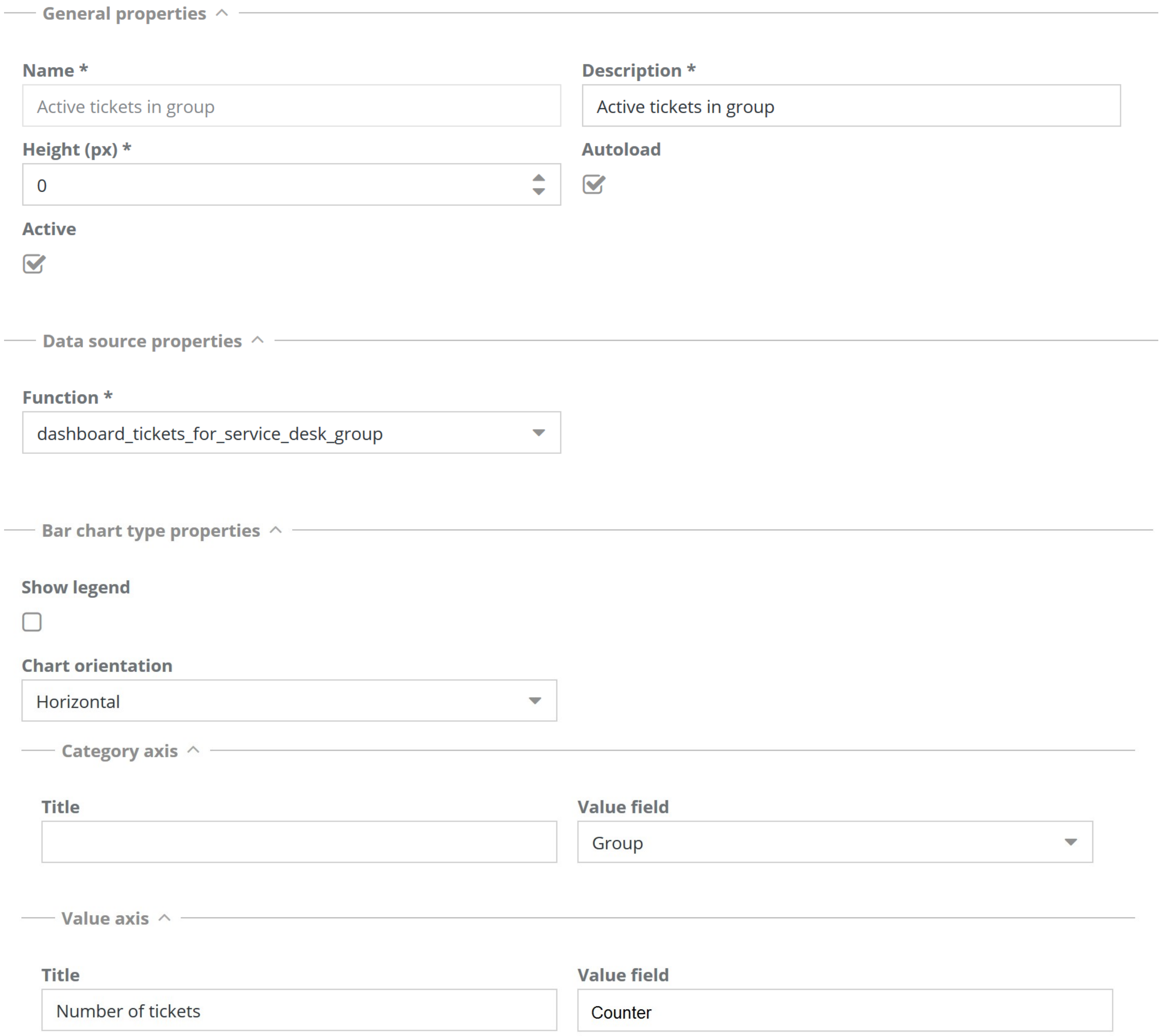
• edit the proportion among the configured columns	 33%, 33%, 33% 50%, 25%, 25% 25%, 50%, 25% 25%, 25%, 50%
• position one of the available types of elements into the layout, with the drag & drop function	

13.1.3. Configuration of Bar chart type properties

Below the properties that can be configured in a Bar chart type property:

- General properties
  - Chart name
  - Chart description
  - Height, in pixel
  - Autoloading when opening the page
  - Active
- Data source: a PostgreSQL function that has to be created in the database according to the instructions mentioned below
- List of input parameters provided by the data source that populates the chart, specifying for each one:
  - Mandatory
  - Data interpretation modes:
    - in case of decimal or data parameters, no further information is required
    - in case of string parameters, you are asked if it is a free string, if it is the name of a class, if you are the current user or the current group
    - in case of integer parameters, you are asked if it is a free number, the ID of a lookup value (in this case you will choose the lookup and the filter), the ID of a data card (in this case you will choose the class and a filter)
  - Default value
- Chart properties:
  - Show legend
  - Axis orientation: horizontal or vertical
  - Title of category board (label)
  - Value of category board (information represented by each bar)

- Title of value board (label)
- Value / values value board (multiselect field for the bar dimensioning)



General properties

Name *
Active tickets in group

Description *
Active tickets in group

Height (px) *
0

Autoload
☑

Active
☑

Data source properties

Function *
dashboard_tickets_for_service_desk_group

Bar chart type properties

Show legend
□

Chart orientation
Horizontal

Category axis

Title

Value field
Group

Value axis

Title
Number of tickets

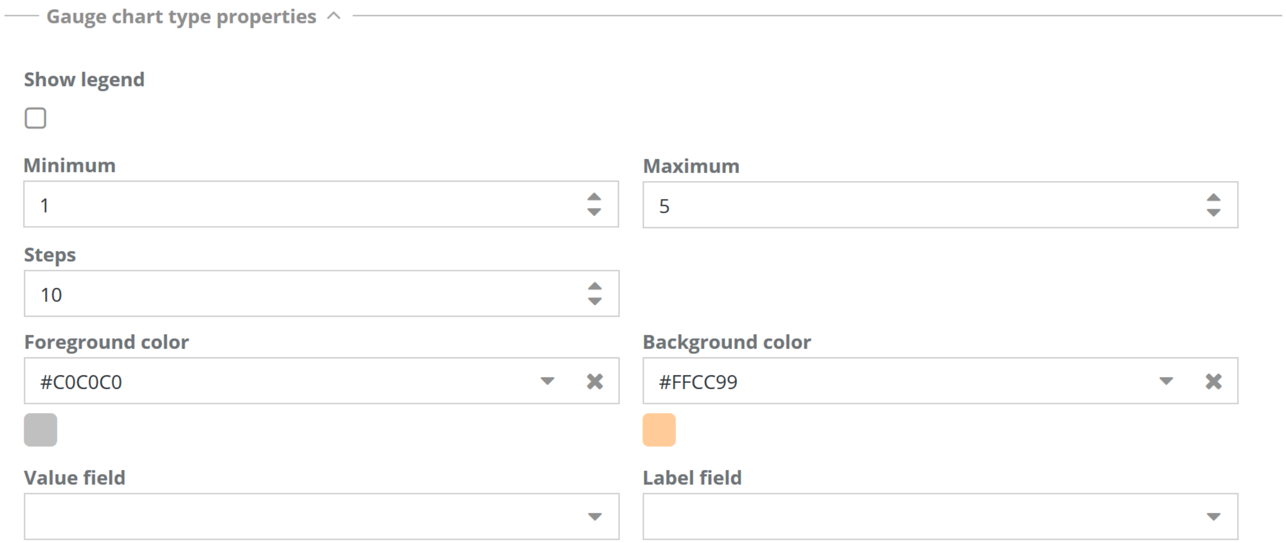
Value field
Counter

13.1.4. Configuration of Gauge chart type properties

The properties that can be configured in a Gauge chart type property are the same described for the Bar charts, except for these specific properties of the chart:

- Show legend
- Maximum scale value
- Minimum scale value (zero if not specified)
- Number of discretization intervals
- Foreground color

- Background color
- Value field (in order to define the indicator shown)



Gauge chart type properties ^

Show legend

□

Minimum: 1

Maximum: 5

Steps: 10

Foreground color: #C0C0C0

Background color: #FFCC99

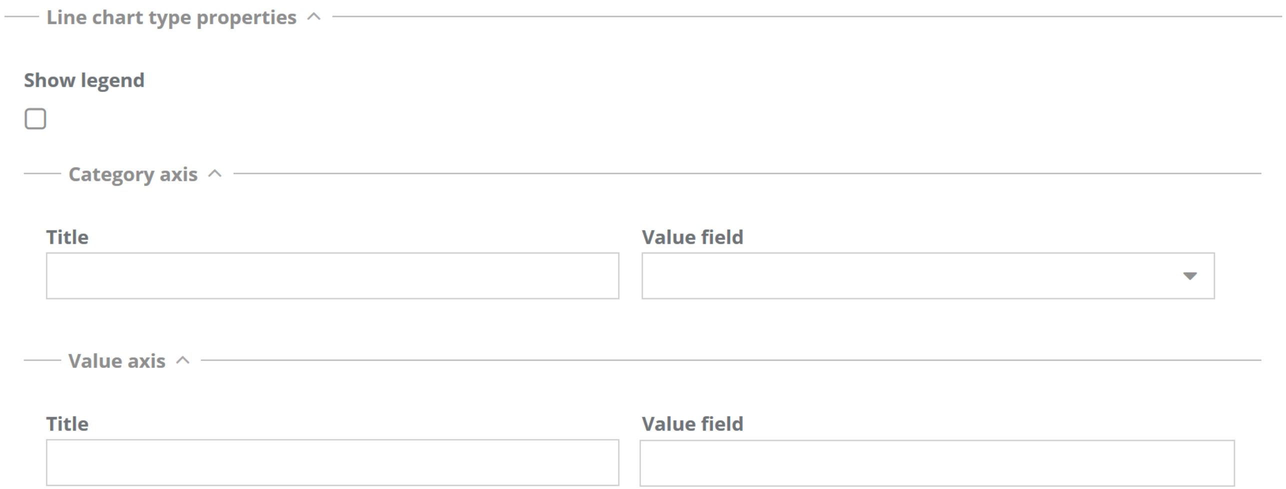
Value field

Label field

13.1.5. Configuration of Line chart type properties

The properties that can be configured in a Line chart type property are the same described for the Bar charts, except for these specific properties of the chart:

- Show legend
- Title of category board (label)
- Value of category board (information represented by each bar)
- Title of value board (label)
- Value / values value board (for the bar dimensioning)



Line chart type properties ^

Show legend

□

Category axis ^

Title

Value field

Value axis ^

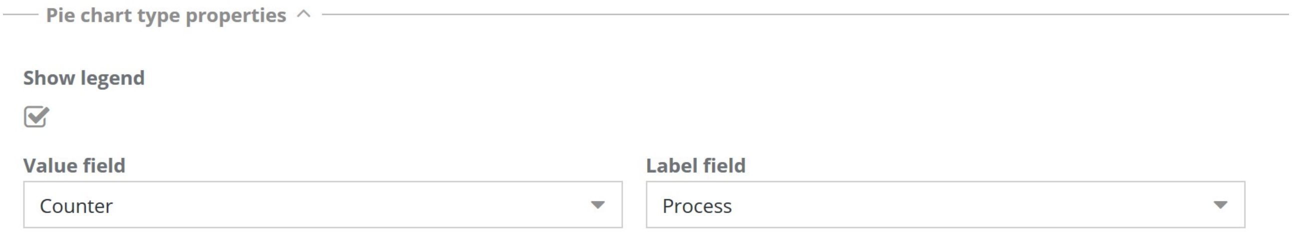
Title

Value field

13.1.6. Configuration of Pie chart type properties

The properties that can be configured in a Pie chart type property are the same described for the Bar charts, except for these specific properties of the chart:

- Show legend
- Value field (for the dimensioning of the areas)
- Label field (cited in the areas)



Pie chart type properties

Show legend

☑

Value field

Counter

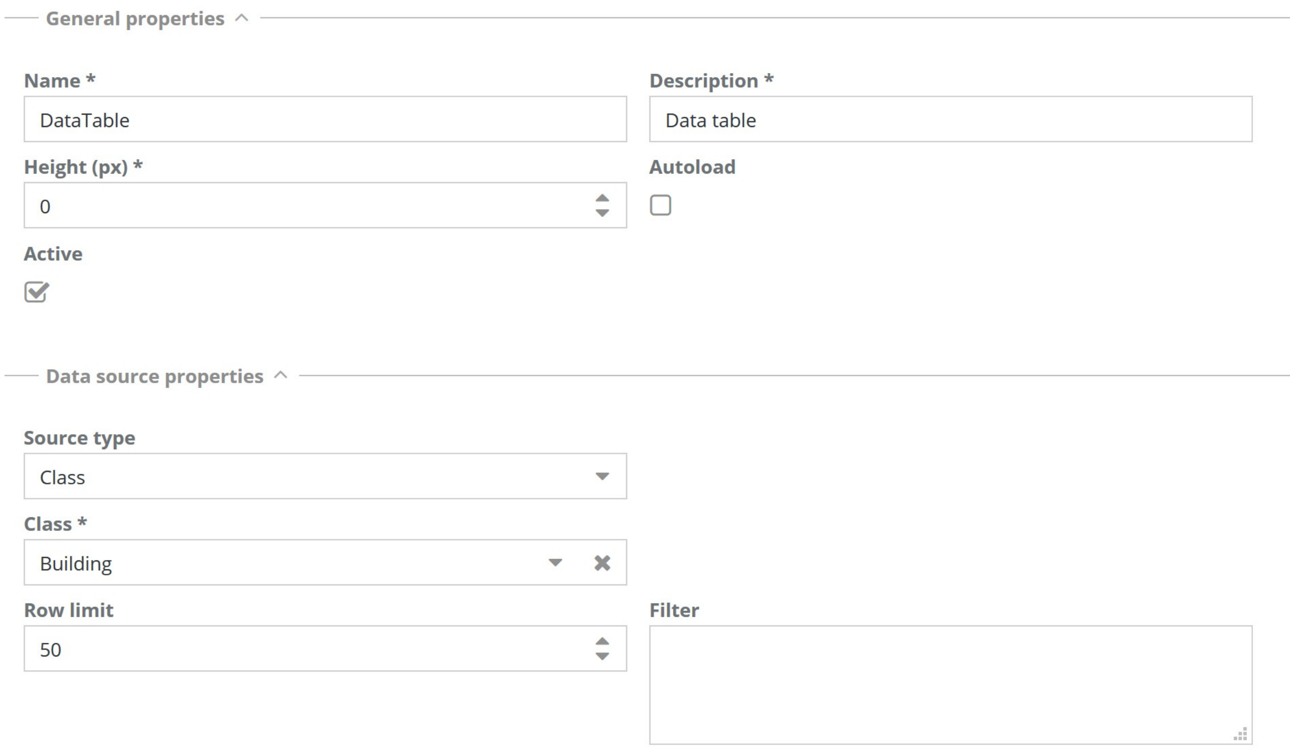
Label field

Process

13.1.7. Configuration of Data table type properties

The properties that can be configured in a Data table property are as follows:

- General properties
  - Activity name
  - Description
  - Height, in pixel
  - Autoloading when opening the page
  - Active
- Properties of the data source
  - Data source type, which takes the following values:
    - Function, in this case you are asked to select the name of the list
    - Class, in this case you are asked to select its name from the list and to specify the number of rows per page and a possible filter
    - View, in this case you are asked to select its name from the list



General properties

Name *: DataTable

Description *: Data table

Height (px) *: 0

Autoload: ☐

Active: ☑

Data source properties

Source type: Class

Class *: Building

Row limit: 50

Filter:

13.1.8. Configuration of Text area type properties

The properties that can be configured in a Text area property are as follows:

- General properties
  - Activity name
  - Description
  - Height, in pixel
  - Autoloading when opening the page
  - Active
- Text area properties:
  - HTML text

13.1.9. Definition of the data source (PostgreSQL function)

In order for the system configuration of charts described above to work accurately, you have to consider in particular the definition of the PostgreSQL function which represents the data source.

In particular:

- the function must report the comment "TYPE: function”
- the definition of the function has to include the input and output parameters, clearly showing the name of each of them

- the input and output parameters have to be chosen among the following ones: “character varying”, “boolean”, “integer”, “bigint”, “numeric”, “double precision”, “date”, “time”, “timestamp”, “text”
- in case that the function becomes more "tuple" than the output values, you should use the syntax "Returns setof record”

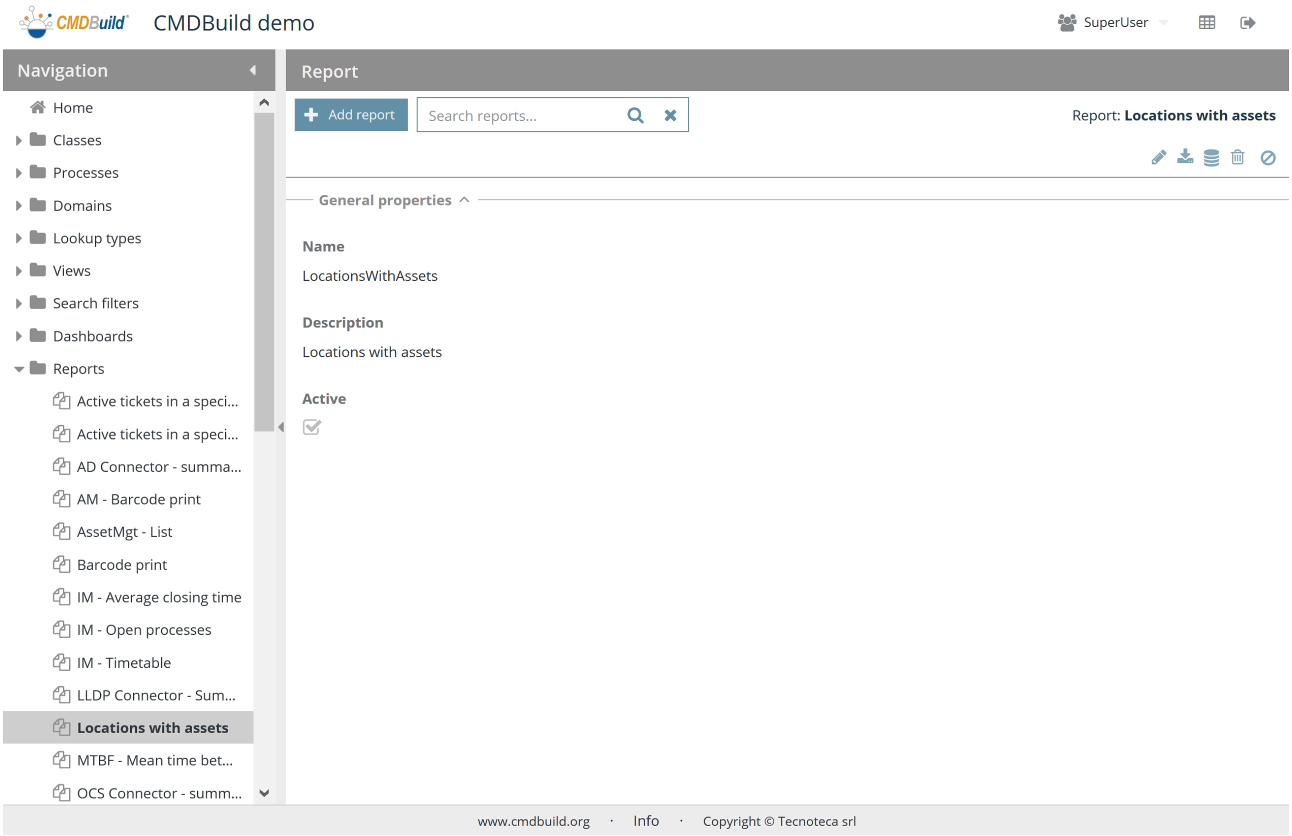
14. Report

CMDBuild allows the administrator to design custom reports, import them into CMDBuild through the Administration Module, insert them into the users' Menu and make them available in the Management Menu.

In order to design reports, CMDBuild uses the external visual editor JasperStudio from the open source suite JasperReports, which allows you to manage intuitively the layout and the formatting of the information extracted from the CMDB and export the template in XML format.

14.1. Properties

The function allows you to create new reports and edit the properties of the old ones.



Report: **Locations with assets**

General properties

Name

LocationsWithAssets

Description

Locations with assets

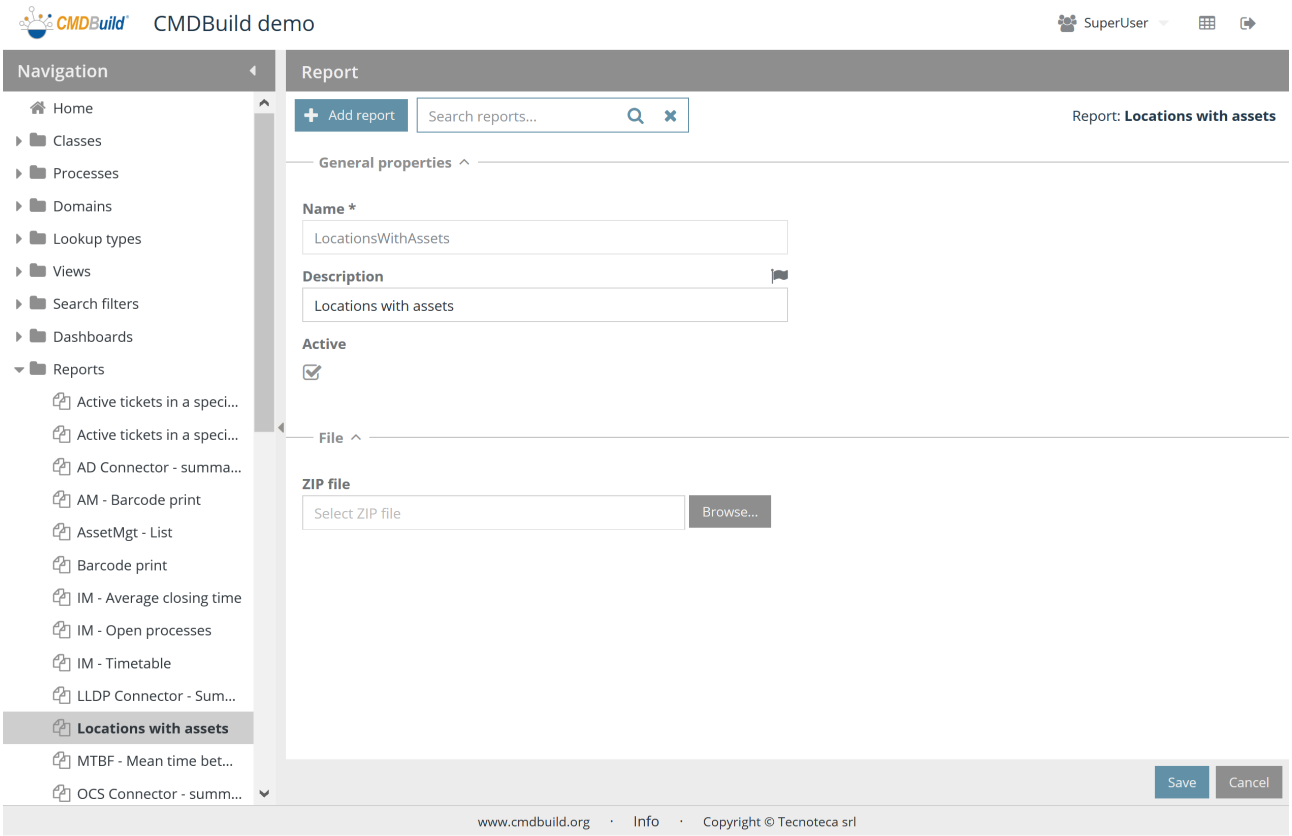
Active

☑

There is the possibility of performing the following operations:

- create a new report	+ Add report
- do searches on the configured reports	Search on all reports 🔍 ✖
- edit the selected report	✎

• download the template of the selected report (in ZIP format, including any subreports and pictures)	⬇
• display the SQL query of the report	🛢
• delete the selected report	🗑
• disable the selected report (i.e. set the status as non-active)	⊘



Report: **Locations with assets**

Name *: LocationsWithAssets

Description: Locations with assets

Active: ☑

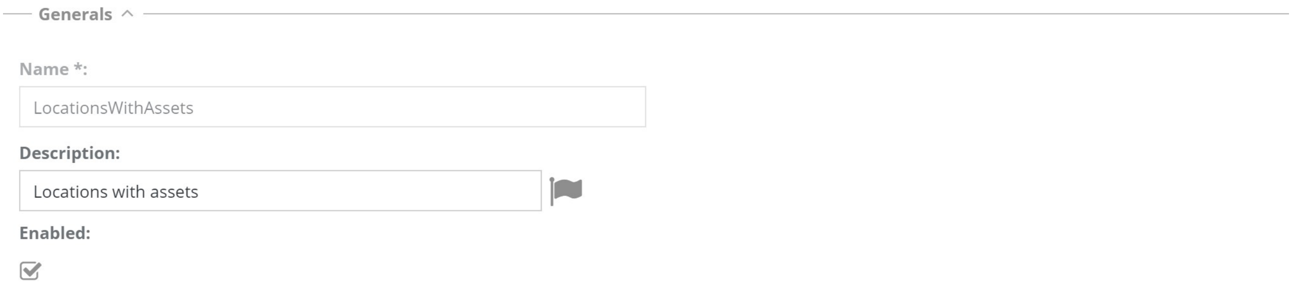
ZIP file: Select ZIP file

Each report needs some parameters to be compiled (metadata), as mentioned.

14.1.1. General properties

The following information is required:

- Name: it is univocal identification of the report
- Description: description of the report
- Active: it states whether the report is active and can be used by an operator



Generals

Name *: LocationsWithAssets

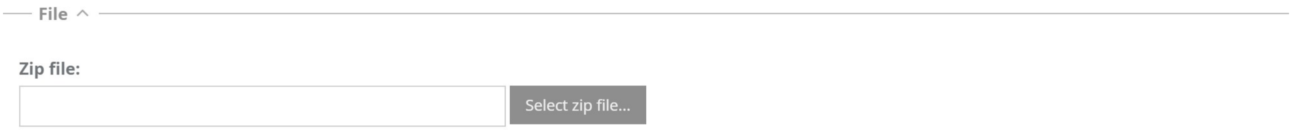
Description: Locations with assets

Enabled: ☑

14.1.2. File

The following information is required:

- ZIP file: upload of the file including the main template (file jrxml), any subreports (jasper files) and any pictures



File

Zip file:

15. Menu

Each user group in CMDBuild has its own custom navigation menu, which can be configured by the administrator.

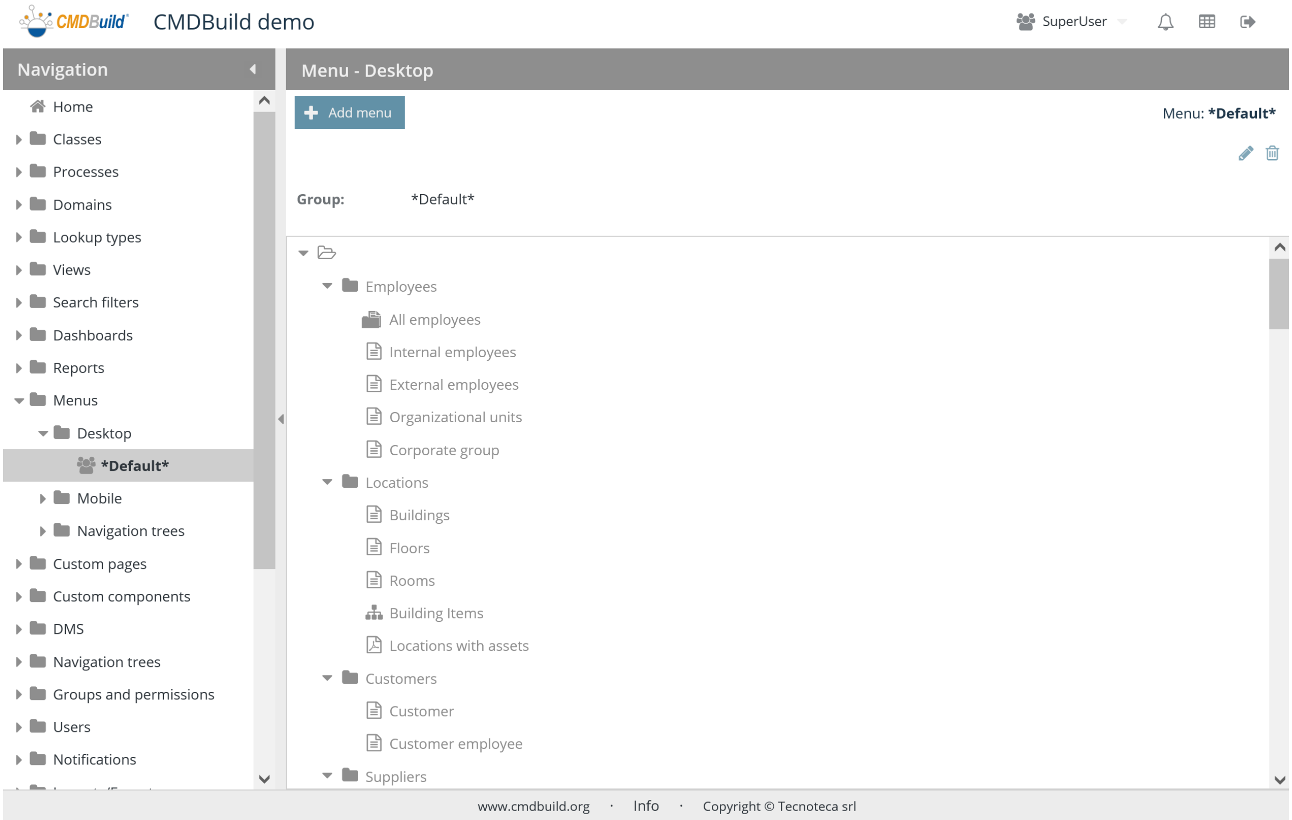
Each menu can include all types of elements created in the system (classes, processes, views, dashboards, reports, custom pages) and custom folders (used to organize entries).

Accessing the Management Module, each operator has a menu configured for his/her own group, or the default menu (if any) filtered according to his/her own needs.

Different menus can be configured for "desktop" and "mobile" interface.

It is also possible to define some class hierarchy, based on relations (specific navigation trees) to "implement" in any position of the menu, in order to navigate in the information by a hierarchic logic (for example from a building to its floors, from a floor to its rooms, from a room to the assets inside it).

15.1. Desktop menu



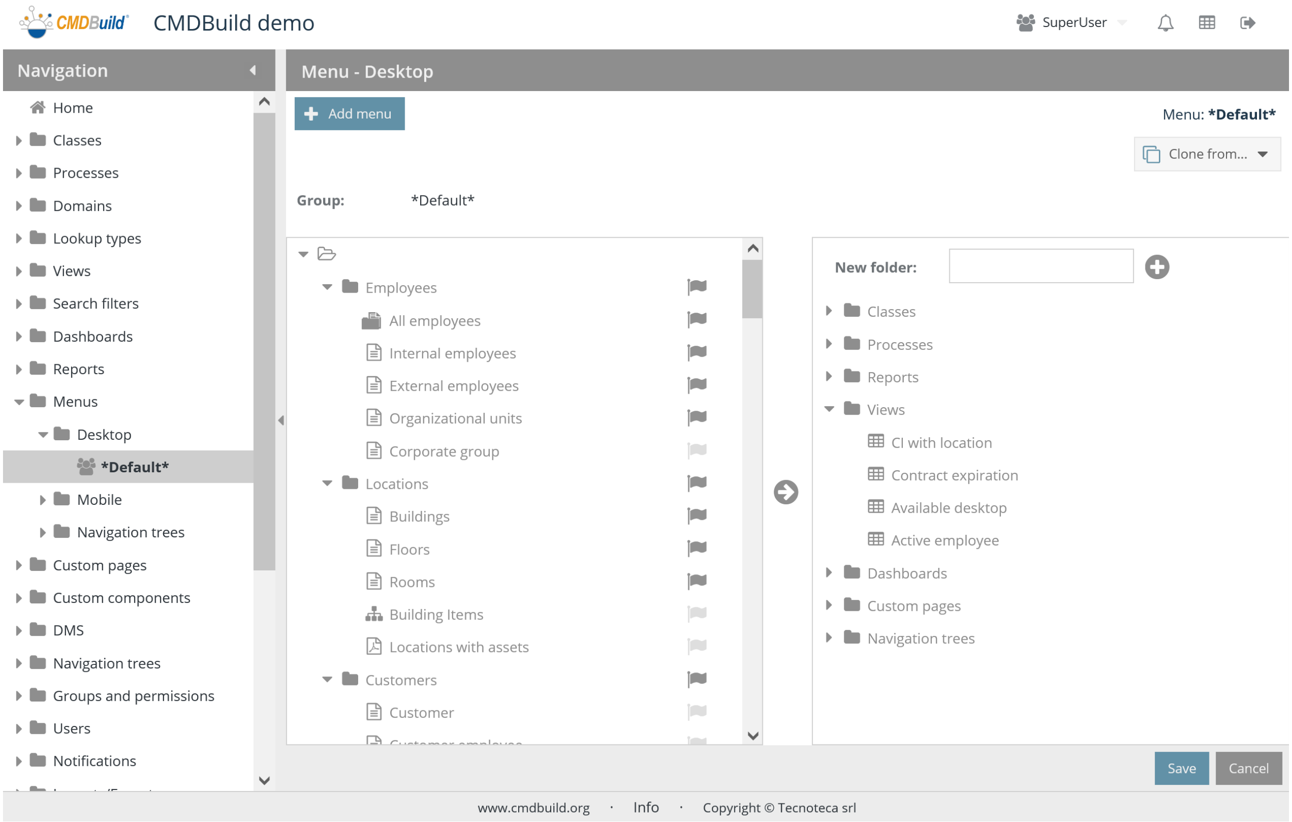
This function allows to create new menus for CMDBuild desktop interface. You can then assign it to different user groups and edit the properties of the old ones.

It is possible to configure a "Default" menu type which will apply to all groups without a specifically configured menu.

There is the possibility of performing the following operations:

• add a new menu	+ Add menu
• edit the selected menu	✎
• delete the selected menu	🗑

The creation or change of a menu can be done as described below.



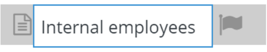
Menu: *Default*

Group: *Default*

15.1.1. Menu configuration

The setup of a menu includes the following operations:

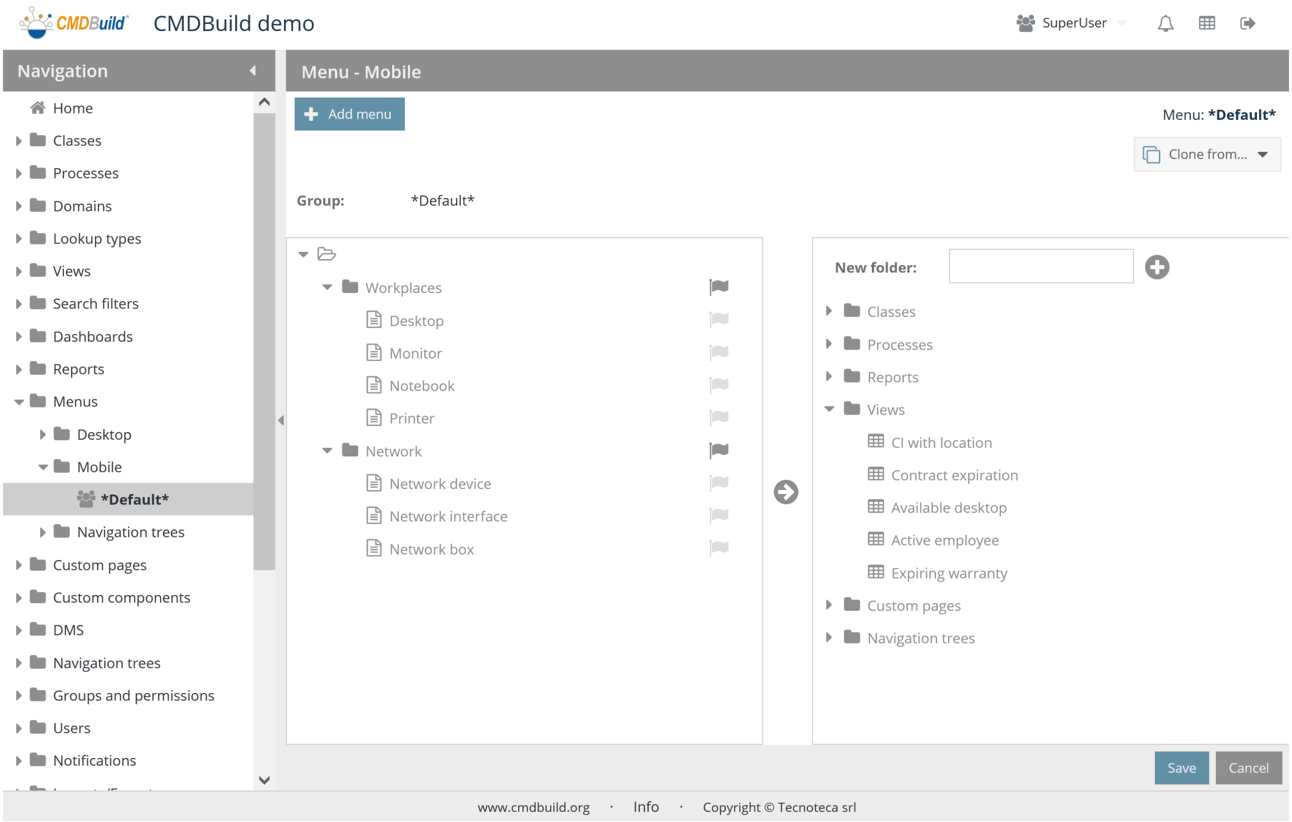
• clone the menu from another configured one	Clone from... ▾ *Default*
• add a new folder	New folder: ⊕
• drag and drop menu items from a list of available elements	🖱
• delete items from the menu	➔

• localization of an element in the menu	⚑
• double-click on the menu to customize the description of the used element	 Internal employees

Once the configuration is confirmed, the menu will instantly be available for the users of the group the menu applies to (or to every group in case of a default menu). They will be able to access the Management Module. If somebody is already connected, he/she will have to clean the browser cache and/or disconnect and reconnect, in order to see the modified menu.

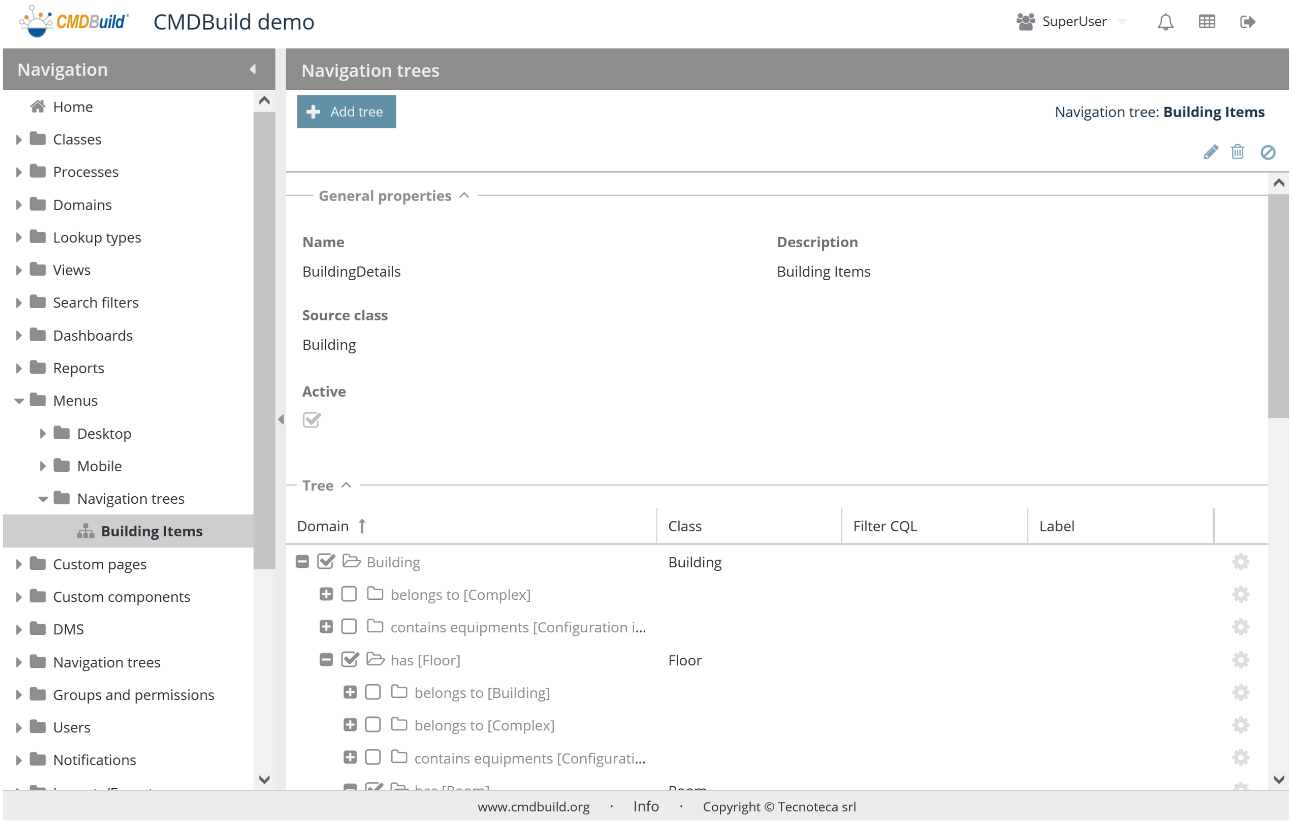
15.2. Mobile menu

For the creation and modification of menus dedicated to the mobile APP, the same indications valid for the desktop interface apply.



15.3. Navigation trees for menus

In addition to the classic navigation through data, moving from a class to another, it is possible to "implement" in any position of the menu a class hierarchy, based on relations 1:N, in order to navigate through information by a hierarchic logic, for example to view the floors inside a building, the rooms inside a floor, the assets inside a room.



Navigation tree: **Building Items**

Name: BuildingDetails

Description: Building Items

Source class: Building

Active: ☑

Domain	Class	Filter CQL	Label
☑ Building	Building		
☐ belongs to [Complex]			
☐ contains equipments [Configuration i...			
☑ has [Floor]	Floor		
☐ belongs to [Building]			
☐ belongs to [Complex]			
☐ contains equipments [Configurati...			

The configuration modalities are the same of generic navigation trees, apart from two more features:

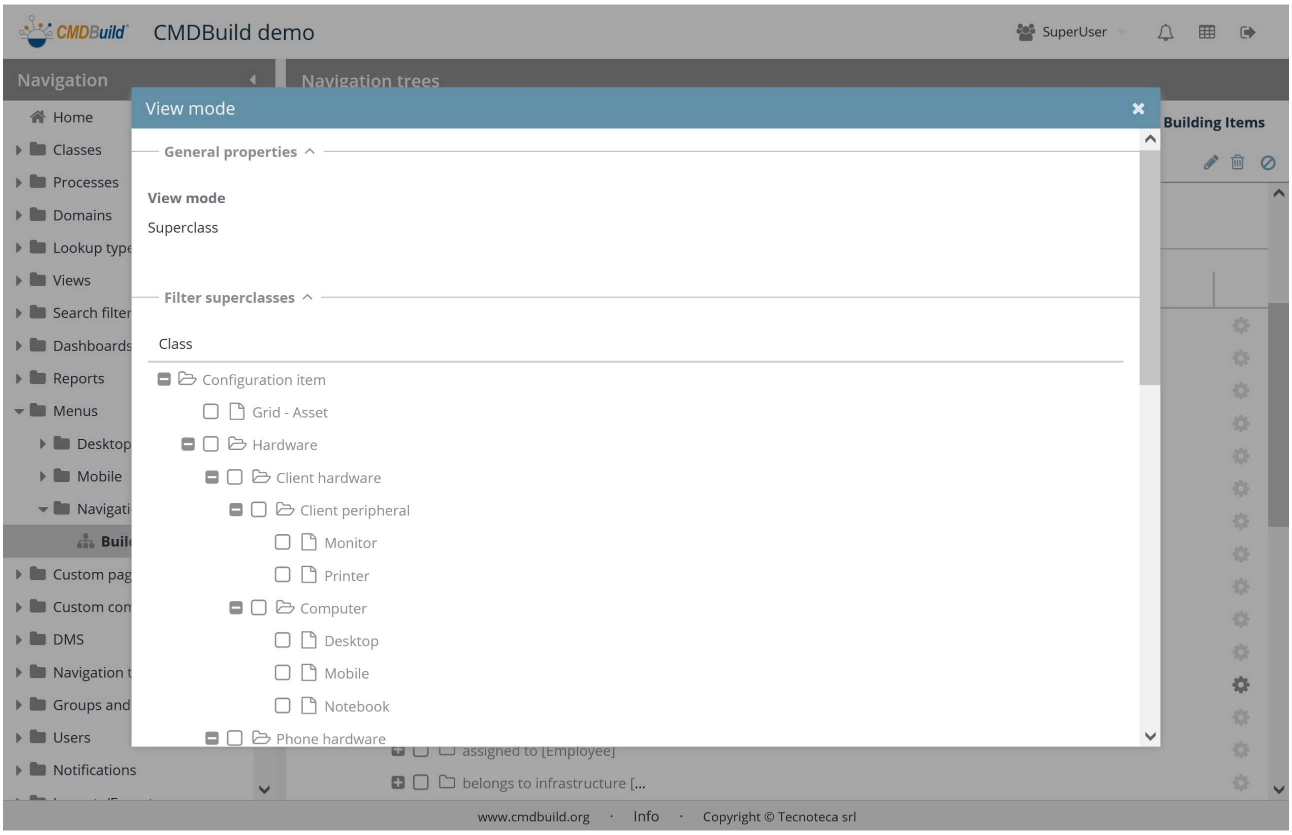
- label: it allows you to assign to the entry of the menu a label that is different from the name of the related class
- view mode: it is active only for selected superclasses, with the options described below

15.3.1. View mode

The view mode includes the following options:

- superclass: cards of all subclasses are shown together in the superclass node
- subclasses: for subclass an entry is inserted into the menu

Through the icon in the column on the right (cogwheel), enabled only for superclasses, you can filter the subclasses you want to include in the menu and manage a custom label from this function.



View mode

General properties

View mode
Superclass

Filter superclasses

Class

- ⊟ Configuration item
  - □ Grid - Asset
  - ⊟ □ Hardware
    - ⊟ □ Client hardware
      - ⊟ □ Client peripheral
        - □ Monitor
        - □ Printer
      - ⊟ □ Computer
        - □ Desktop
        - □ Mobile
        - □ Notebook
    - ⊟ □ Phone hardware

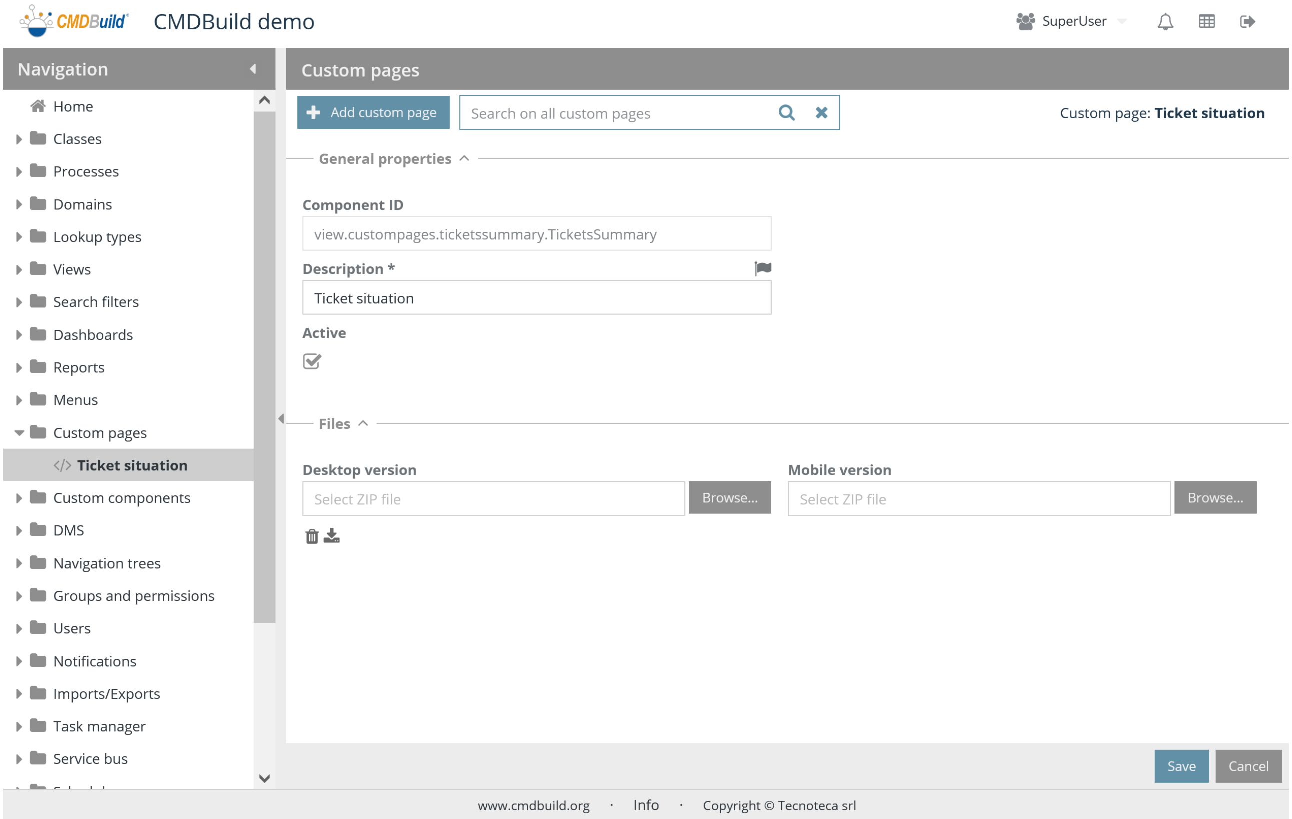
16. Custom page

In the user interface of the Management Module you can include data of custom pages, subject to the standard security mechanisms of CMDBuild.

The Custom pages are programmed in javascript though the same mechanisms of Sencha ExtJs, the framework used for the standard implementation of CMDBuild.

Pages with different layouts can be configured for "desktop" and "mobile" interface.

16.1. Custom pages



Custom page: **Ticket situation**

General properties

Component ID: view.custompages.ticketssummary.TicketsSummary

Description *: Ticket situation

Active: ☑

Files

Desktop version: Select ZIP file Browse...

Mobile version: Select ZIP file Browse...

Using this function you can configure new custom pages or edit the properties of the previous ones.

There is the possibility of performing the following operations:

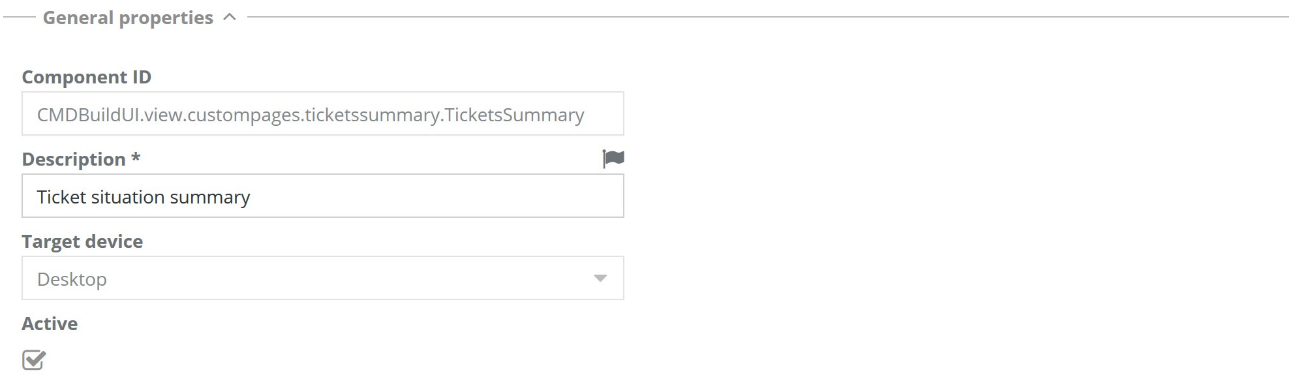
• create a new custom page	+ Add custom page
• do searches on configured custom pages	Search on all custom pages 🔍 ✖
• edit the selected custom page	✎

• download the code of the selected custom page (in ZIP format)	⤓
• delete the selected custom page	🗑
• disable the selected custom page (i.e. set the status as non-active)	⊘

16.1.1. General properties

The following information is managed:

- ID component: it is assigned by the system during its creation and it is available only in view-mode in the modification form of a custom page
- Description: it is the unique name assigned to the custom page
- Active: it states whether the custom page is active and can be used by an operator



General properties ^

Component ID
CMDBuildUI.view.custompages.ticketssummary.TicketsSummary

Description *
Ticket situation summary

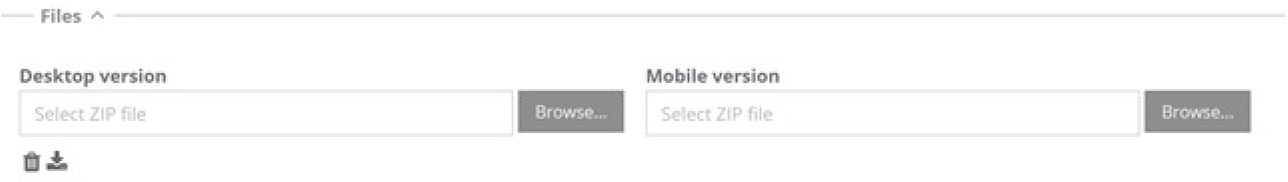
Target device
Desktop

Active
☑

16.1.2. File

The following information is required:

- Desktop ZIP file: it includes all files related to the implementation of the custom page on the desktop user interface
- Mobile ZIP file: it includes all files related to the implementation of the custom page on the mobile user interface



Files ^

Desktop version
Select ZIP file

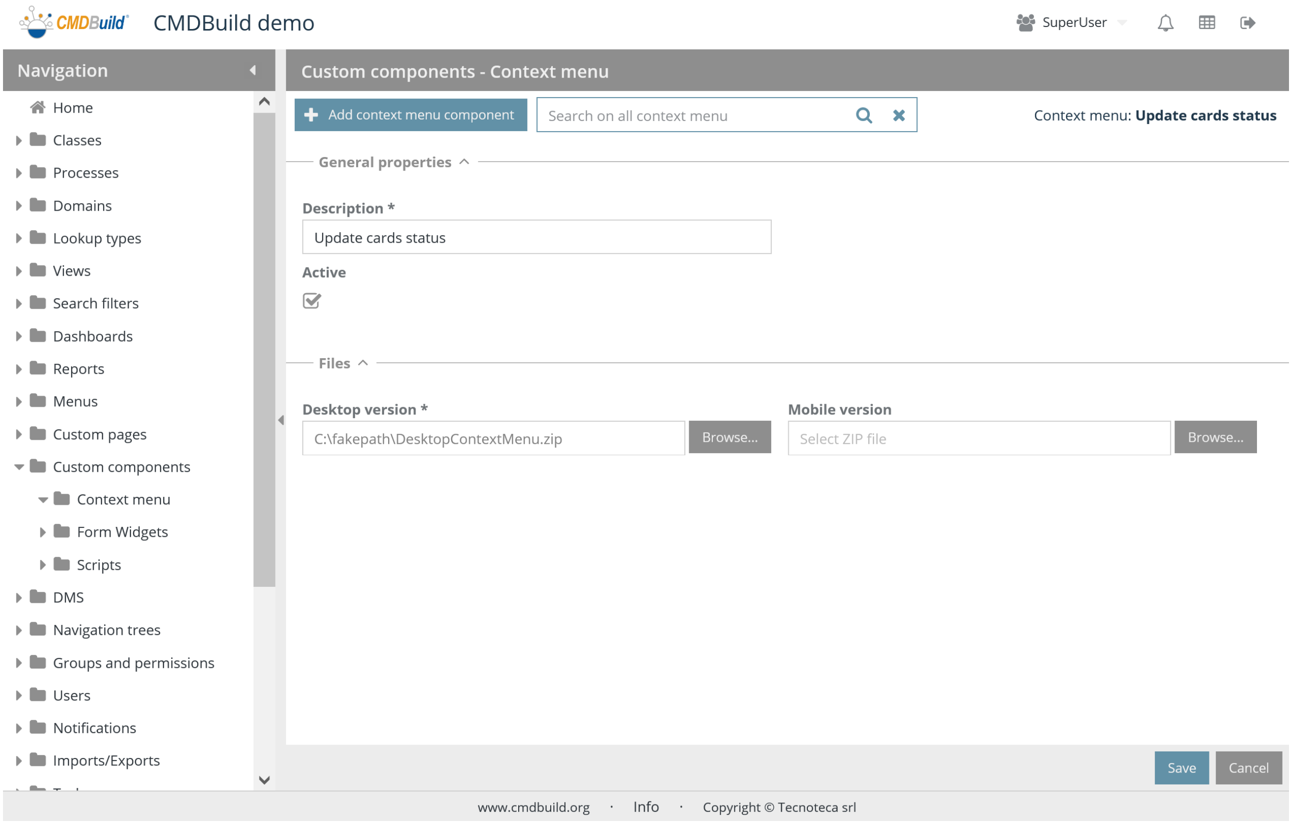
Mobile version
Select ZIP file

17. Custom component

CMDBuild allows the administrator to create custom UI components, import them into CMDBuild through the Administration Module and use them when creating contextual menus and widgets.

17.1. Contextual menus

This function allows you to define the custom components for the creation of contextual menus used both in the desktop interface and in the mobile interface and import their implementation javascript code.



Context menu: **Update cards status**

Description *: Update cards status

Active: ☑

Desktop version *: C:\fakepath\DesktopContextMenu.zip

Mobile version: Select ZIP file

There is the possibility of performing the following operations:

• create a new custom component	✚ Add context menu component
• do searches on configured custom components	Search on all context menu 🔍 ✖
• edit the selected custom component	✎
• delete the selected custom component	🗑

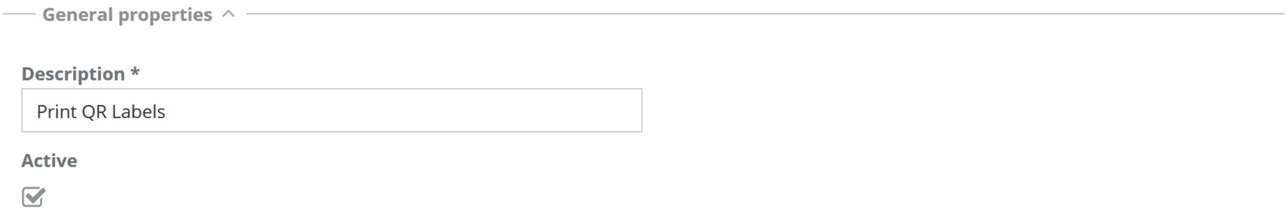
• disable the selected custom component (i.e. set the status as non-active)	⊘

Each custom component needs some parameters to be compiled (metadata), as mentioned below.

17.1.1. General properties

The following information is required:

- Description: description of the custom component
- Active: it states whether the custom component is active and can be used



General properties ^

Description *

Print QR Labels

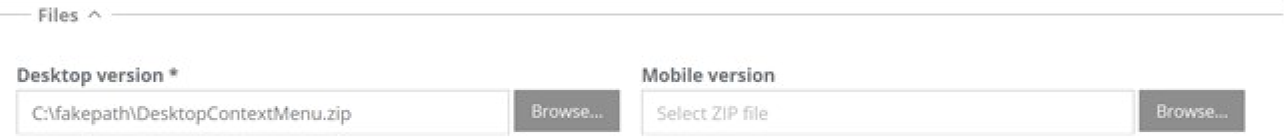
Active

☑

17.1.2. File

The following information is required:

- Desktop ZIP file: it includes all files related to the implementation of the custom menu on the desktop user interface
- Mobile ZIP file: it includes all files related to the implementation of the custom menu on the mobile user interface



Files ^

Desktop version *

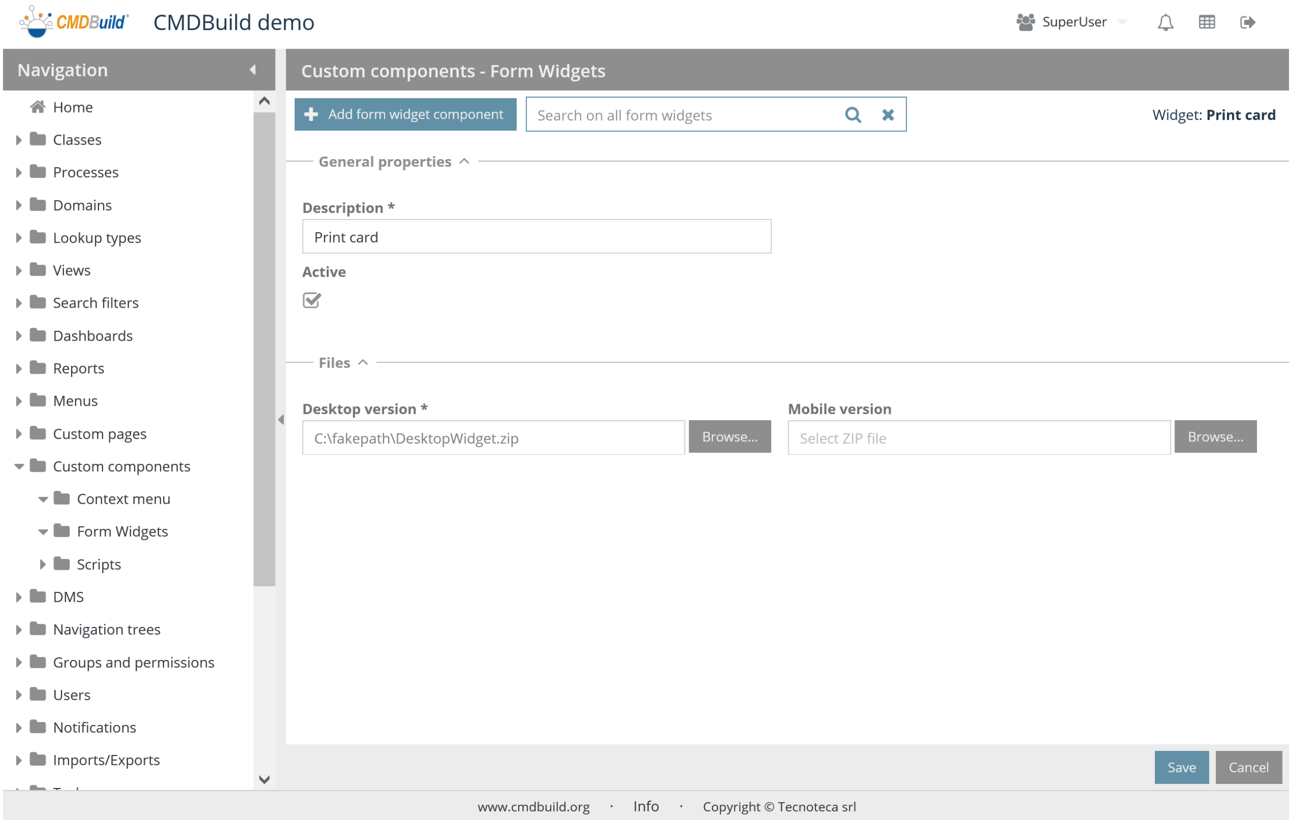
C:\fakepath\DesktopContextMenu.zip

Mobile version

Select ZIP file

17.2. Form widget

This function allows you to define the custom components for the creation of widgets to be shown in data cards, both in the desktop and mobile interface, and import their implementation javascript code.



Widget: Print card

Description *: Print card

Active: ☑

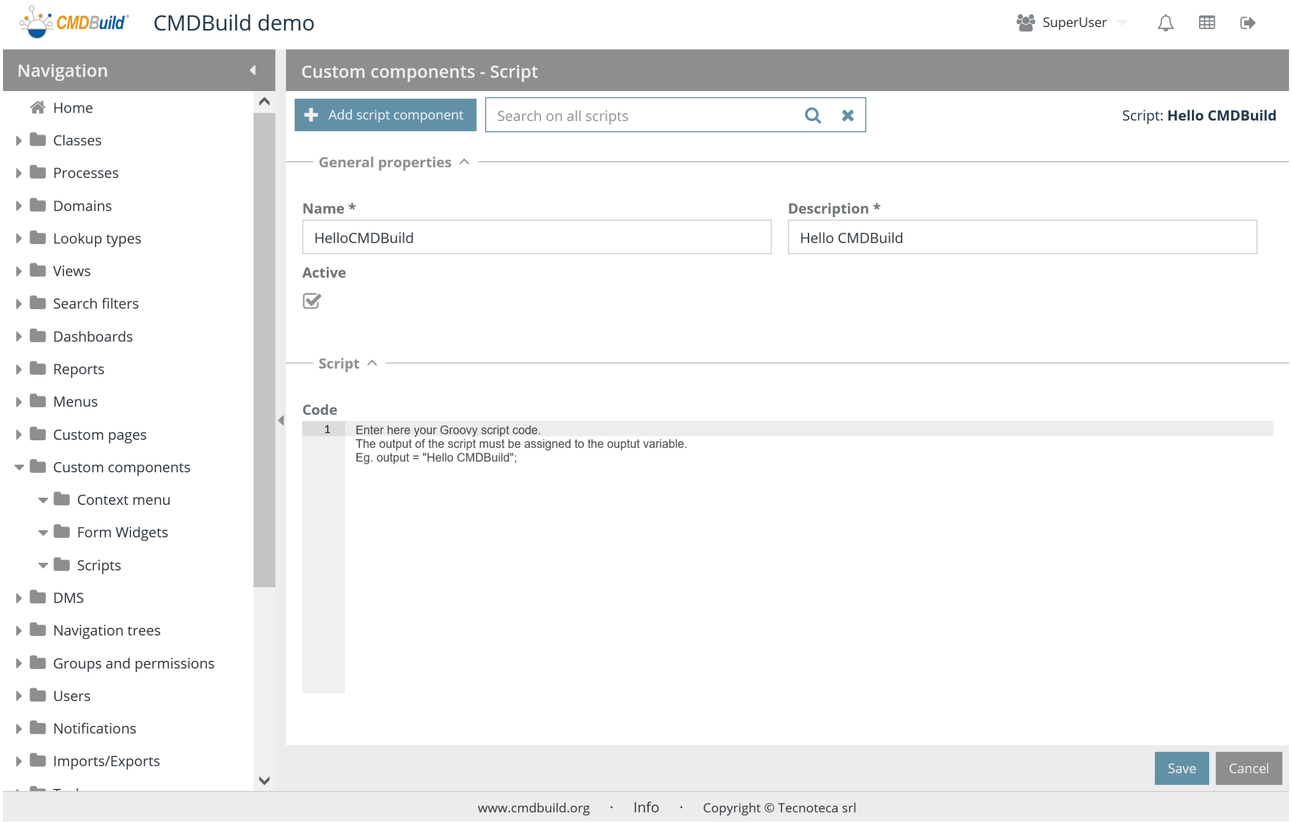
Desktop version *: C:\fakepath\DesktopWidget.zip

Mobile version: Select ZIP file

The management modes are similar to the components of contextual menus described above.

17.3. Script

This function allows you to define the scripts used in cards for the definition of custom behaviors, both in the desktop and mobile interface.



Script: **Hello CMDBuild**

Name * HelloCMDBuild

Description * Hello CMDBuild

Active ☑

Code

```
Enter here your Groovy script code.
The output of the script must be assigned to the ouptut variable.
Eg. output = "Hello CMDBuild";
```

The management modes are similar to the ones of other custom components, except for the scripts that are inserted from user interface and are not imported as file.

18. DMS

CMDBuild includes the integration with document systems (DMS Document Management System) for filing and consulting attachments of Configuration Items.

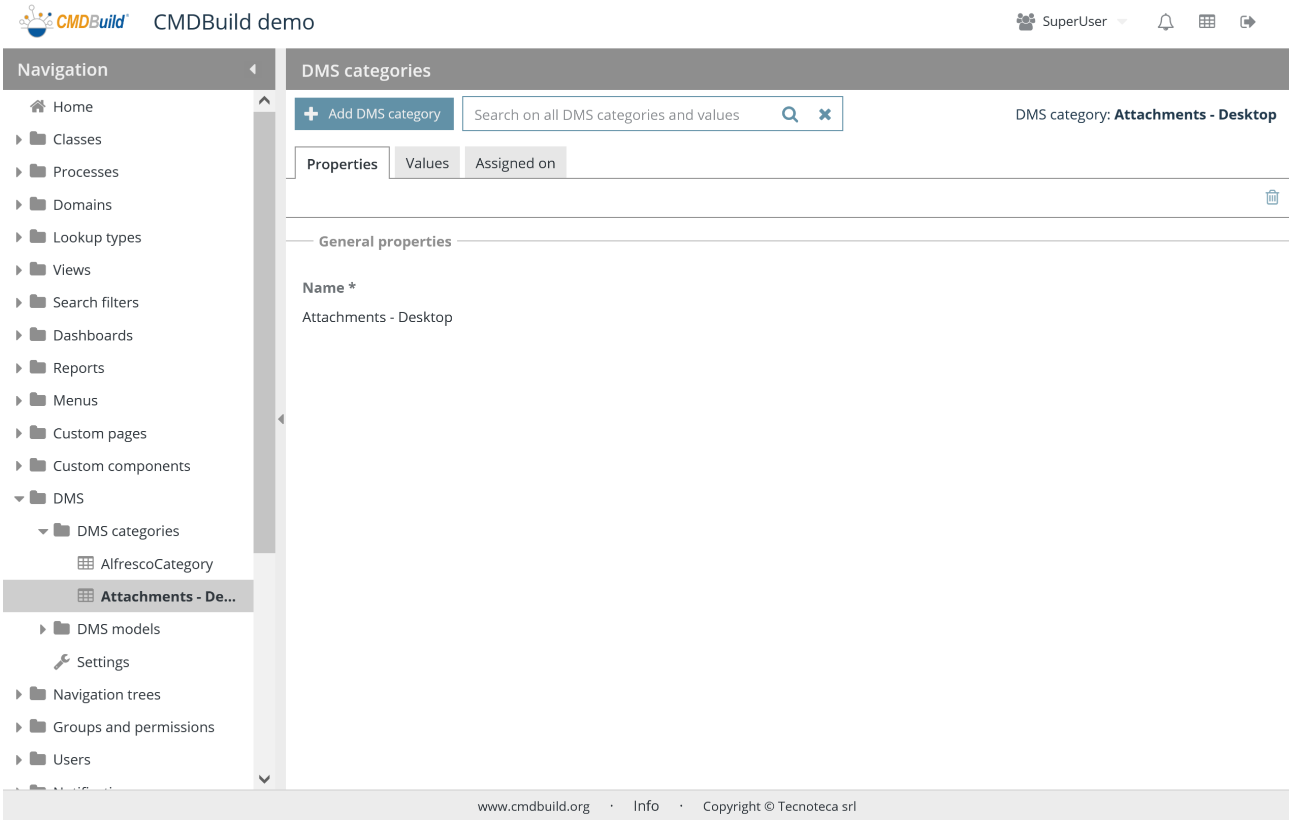
The integration is based on the standard protocol CMIS (Content Management Interoperability Services), which has to be supported by the chosen DMS. CMDBuild recommends the DMS open source Alfresco Community version, set as default solution. . The CMIS solution related to Microsoft Share Point online is also supported.

DMS systems are used as repository for documents, while CMDBuild remains the user interface to load and download documents.

The attachments can be grouped in document categories which can be defined in a diversified way for each class. Each category can be associated to a document model for the pre-configuration of specific behaviours for that type of documents; in particular the activation of additional attributes for managing specific information (metadata).

18.1. DMS categories

This function allows the creation of new document categories, which can be associated to one or more CMDBuild classes and customized in terms of attributes (document metadata) and other behaviours through document models described in the following paragraph.



DMS category: **Attachments - Desktop**

General properties

Name *

Attachments - Desktop

18.1.1. Properties tab

There is the possibility of performing the following operations:

• create a new Document category	+ Add value
• do searches on the configured Document categories	Search... 🔍 ✖
• delete the selected Document category	🗑

Properties

The following information is required for the configuration:

- Name (not editable)

General properties

Name *

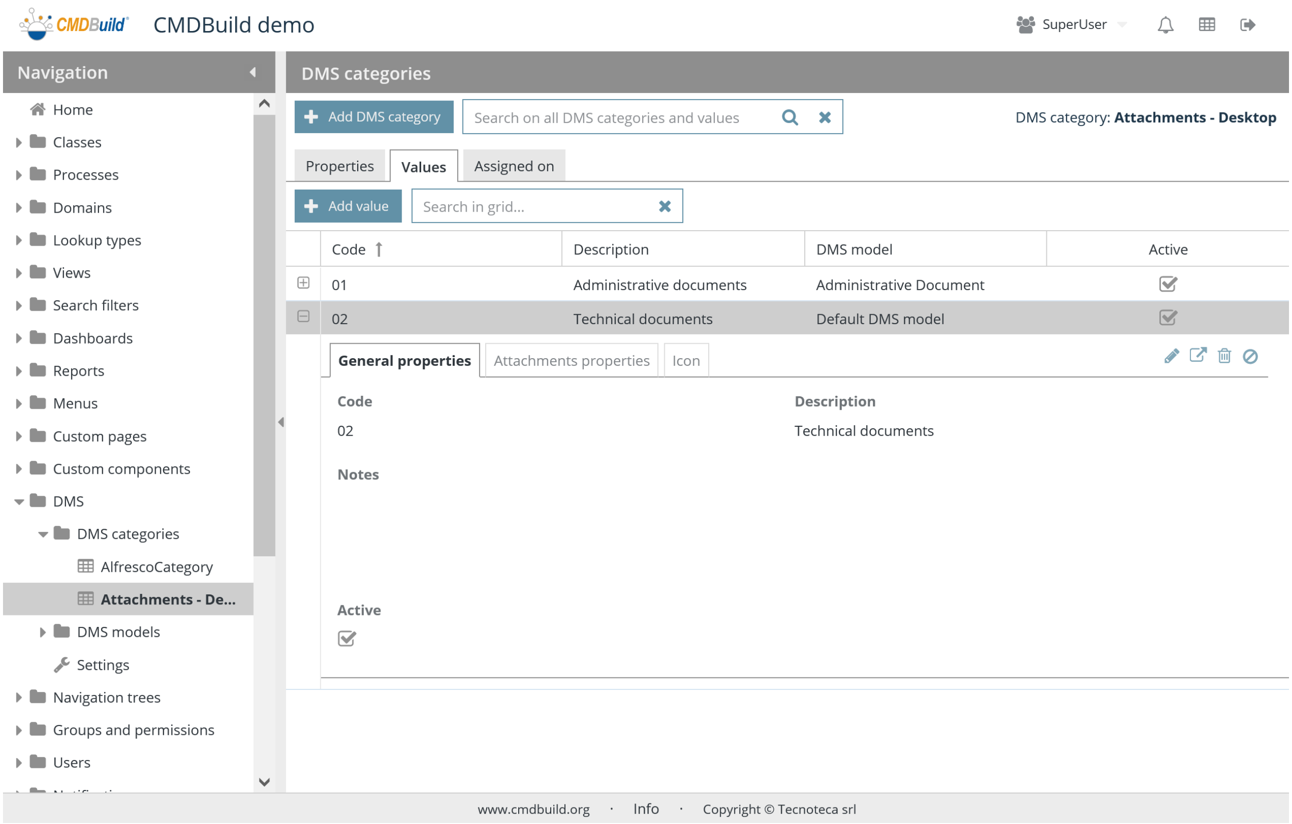
18.1.2. Values TAB

The function allows you to create the values associated to the current document category for classifying and differentiating documents related to CMDBuild cards belonging to the class associated to that category.

There is the possibility of performing the following operations:

• create a new entry in the Document category	+ Add value
• do searches on entries of the document categories	Search in grid... ✖
• expand the current row and display its data	⊞
• edit the selected entry	✎
• display the selected entry	↗
• delete the selected entry	🗑
• disable the selected entry (i.e. set the status as non-active)	⊘

Each value of the list requires some parameters to be compiled, as described below.



DMS category: **Attachments - Desktop**

Code	Description	DMS model	Active
01	Administrative documents	Administrative Document	☑
02	Technical documents	Default DMS model	☑

Code
02

Description
Technical documents

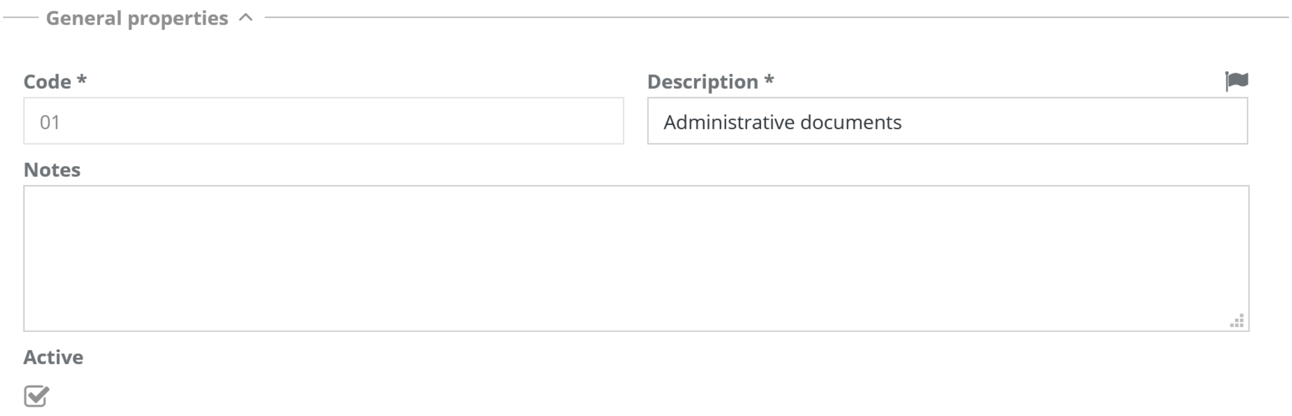
Notes

Active
☑

General properties

The following information is required:

- Code: DMS category code
- Description: DMS category description
- Notes
- Active: the category is active and can be used



Code *
01

Description *
Administrative documents

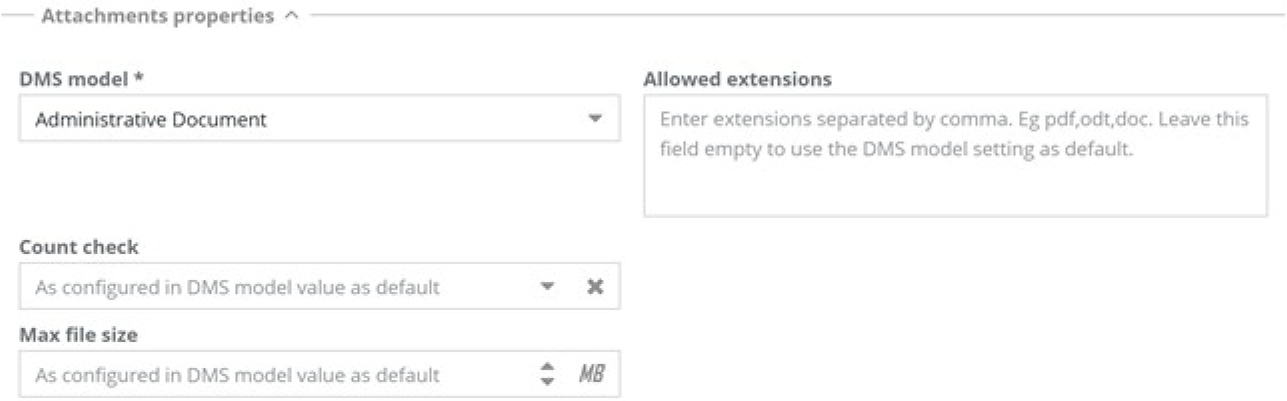
Notes

Active
☑

Attachment properties

The following information is required:

- DMS model: you can select the DMS model (similar to a CMDBuild class) that has to be associated to the document category in order to customize its behaviour (custom attributes and others)
- Allowed extensions: extensions allowed for the files that have to be attached to the current document category
- Count check: it allows you to specify if for each card the current document category will include at least the number of attachments specified, the right number of specified attachments or a maximum number of specified attachments
- Control on the maximal fiel size



Attachments properties

DMS model *: Administrative Document

Allowed extensions: Enter extensions separated by comma. Eg pdf,odt,doc. Leave this field empty to use the DMS model setting as default.

Count check: As configured in DMS model value as default

Max file size: As configured in DMS model value as default MB

Icon

The following information is required:

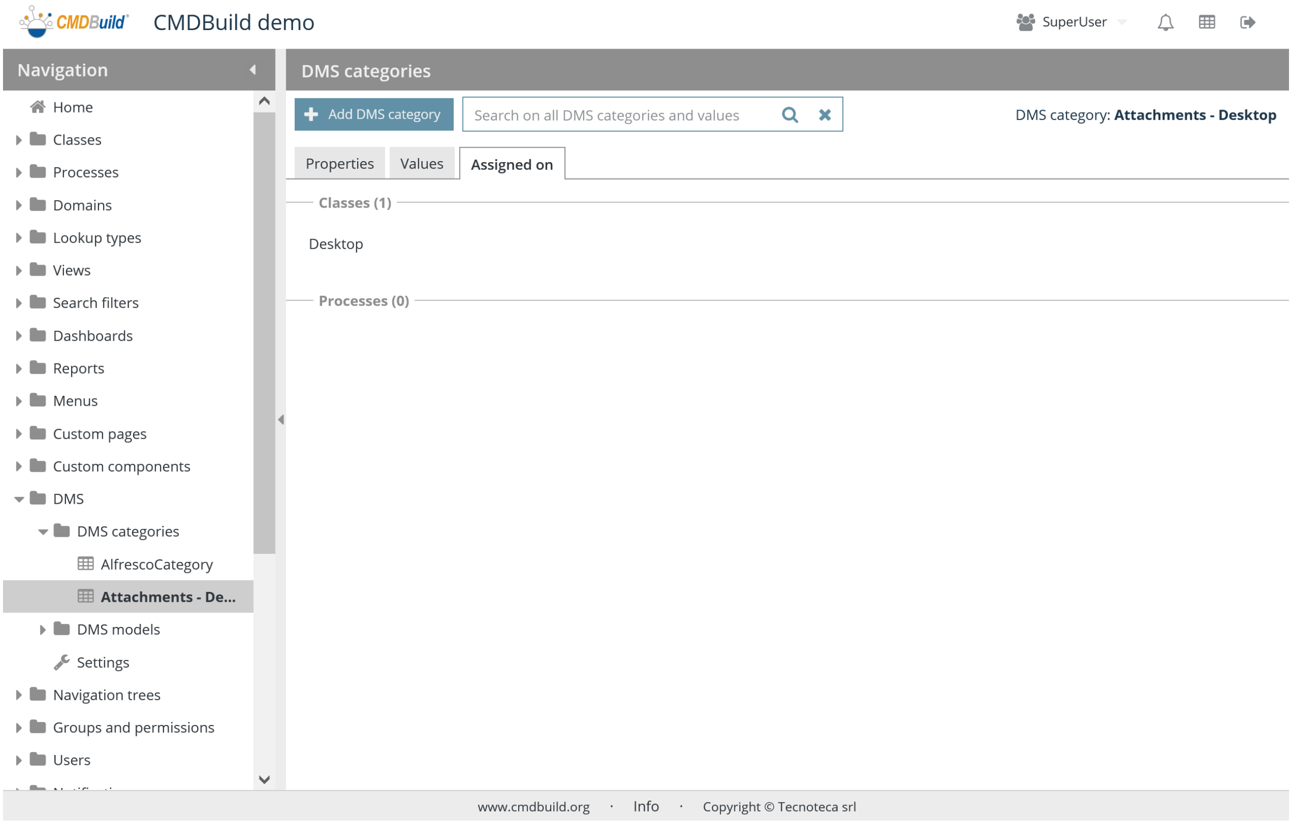
- Icon

Icon

Icon: Select an .png file

18.1.3. TAB Assignment

The function allows you to display the list of classes and processes that use the current document category.


CMDBuild demo
SuperUser
Navigation
Home
Classes
Processes
Domains
Lookup types
Views
Search filters
Dashboards
Reports
Menus
Custom pages
Custom components
DMS
DMS categories
AlfrescoCategory
Attachments - De...
DMS models
Settings
Navigation trees
Groups and permissions
Users
DMS categories
Add DMS category
Search on all DMS categories and values
DMS category: Attachments - Desktop
Properties
Values
Assigned on
Classes (1)
Desktop
Processes (0)
www.cmdbuild.org · Info · Copyright © Tecnoteca srl

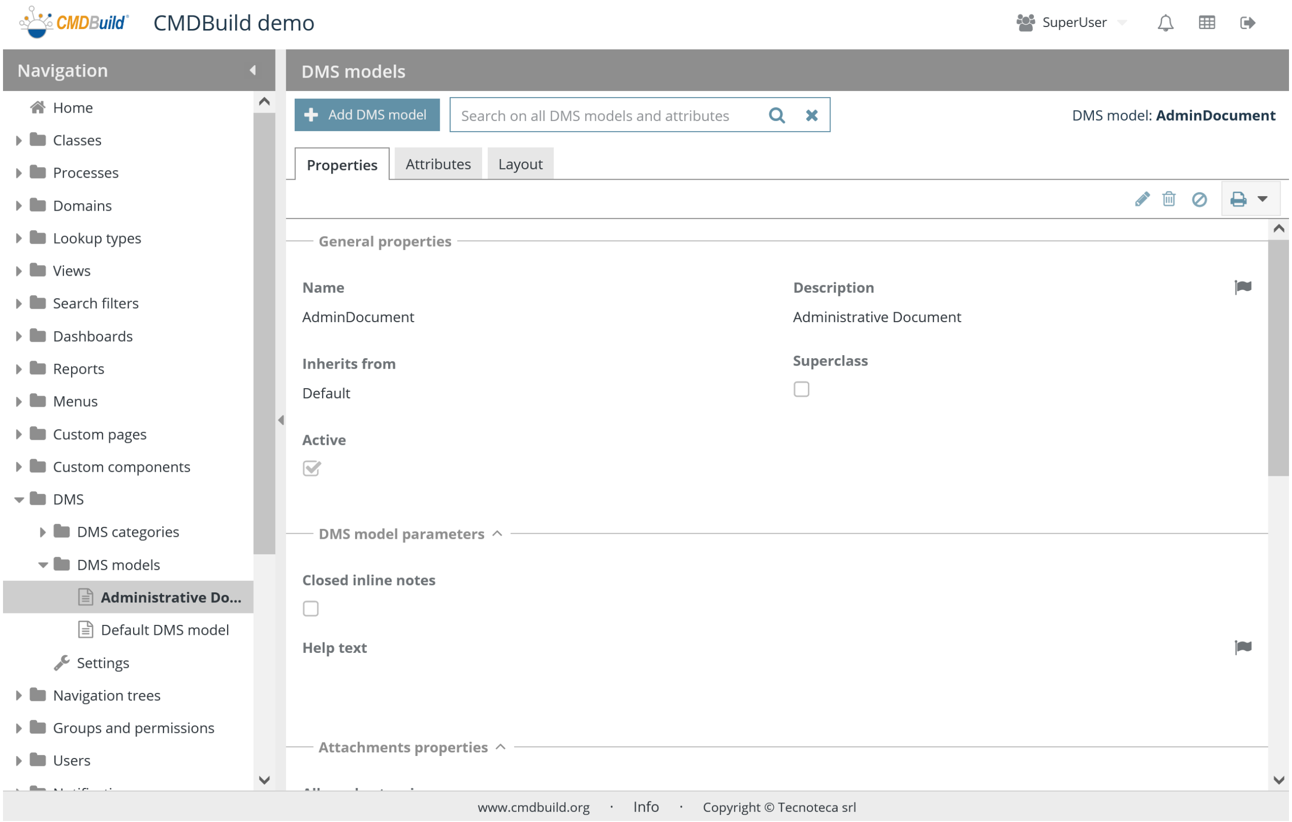
18.2. DMS models

The function allows you to create new document models, which can be associated to a document category in order to customize its behaviour (custom behaviour, etc.).

A DMS model is composed of a particular CMDBuild class, keeping its main features and config possibilities.

18.2.1. Properties tab

The function allows you to manage document models.



DMS model: **AdminDocument**

General properties

Name: AdminDocument

Description: Administrative Document

Inherits from: Default

Superclass: □

Active: ☑

DMS model parameters

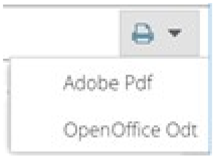
Closed inline notes: □

Help text

Attachments properties

There is the possibility of performing the following operations:

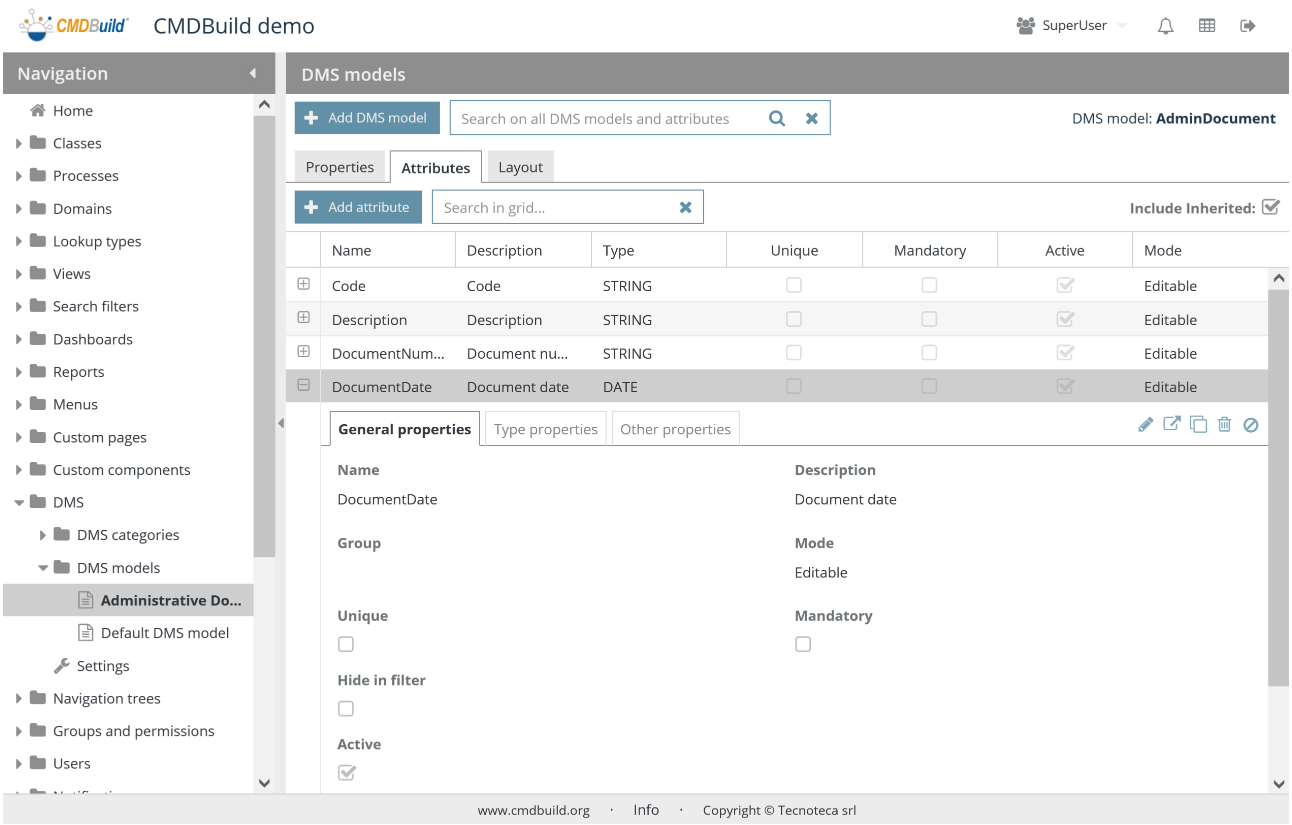
- create a new DMS model	➕ Add DMS model
- do searches on configured DMS models	Search on all DMS models and attributes 🔍 ✖
- edit the selected DMS model	✎
- delete the selected DMS model	🗑
- disable the selected DMS model (i.e. set the status as non-active)	⊘

• print the structure of the current DMS model	

The properties of a document model are the ones available for the common CMDBuild classes. Please read the paragraph related to their meaning and config modes.

18.2.2. Attributes tab

The Attributes tab allows you to manage the attributes of the selected document Model



CMDBuild demo

SuperUser

Navigation

- Home
- Classes
- Processes
- Domains
- Lookup types
- Views
- Search filters
- Dashboards
- Reports
- Menus
- Custom pages
- Custom components
- DMS
  - DMS categories
  - DMS models
    - Administrative Do...
    - Default DMS model
  - Settings
- Navigation trees
- Groups and permissions
- Users

DMS models

Add DMS model | Search on all DMS models and attributes

DMS model: **AdminDocument**

Properties | **Attributes** | Layout

Add attribute | Search in grid...

Include Inherited: ☑

Name	Description	Type	Unique	Mandatory	Active	Mode
Code	Code	STRING	□	□	☑	Editable
Description	Description	STRING	□	□	☑	Editable
DocumentNum...	Document nu...	STRING	□	□	☑	Editable
DocumentDate	Document date	DATE	□	□	☑	Editable

General properties | Type properties | Other properties

Name
DocumentDate

Description
Document date

Group

Mode
Editable

Unique
□

Mandatory
□

Hide in filter
□

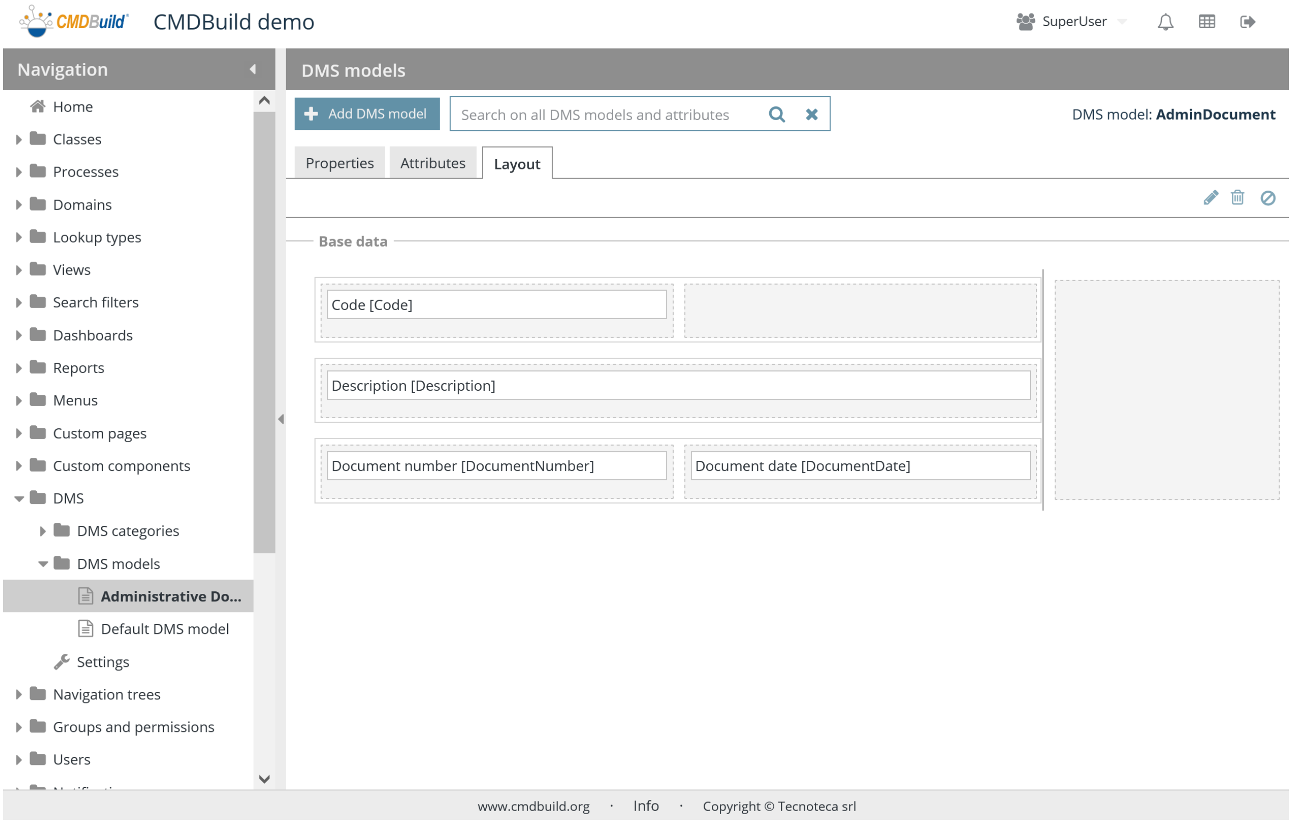
Active
☑

www.cmdbuild.org · Info · Copyright © Tecnoteca srl

The properties of attributes of a document model are the ones available for the attributes of the common CMDBuild classes. Please read the paragraph related to their meaning and config modes.

18.2.3. Layout TAB

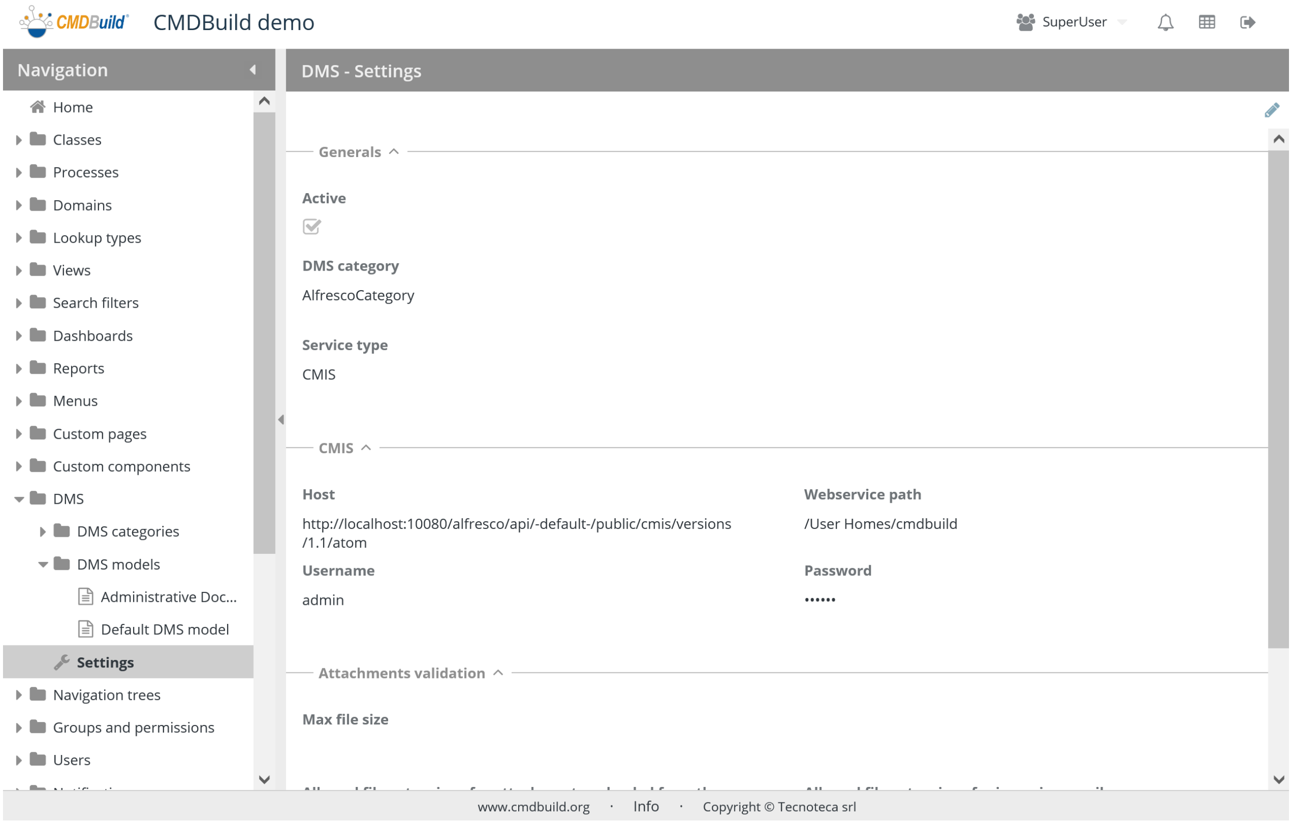
The Layout tab allows you to manage the layout of the selected document model.



The configuration of the layout of a DMS model is similar to the layout of a common CMDBuild class. Please read the related paragraph for the management modes.

18.3. Configurations

The function allows you to set some basic configuration parameters of the document features available in the CMDBuild system.



DMS - Settings

Generals

Active

☑

DMS category

AlfrescoCategory

Service type

CMIS

CMIS

Host

http://localhost:10080/alfresco/api/-default-/public/cmis/versions/1.1/atom

Webservice path

/User Homes/cmdbuild

Username

admin

Password

••••••

Attachments validation

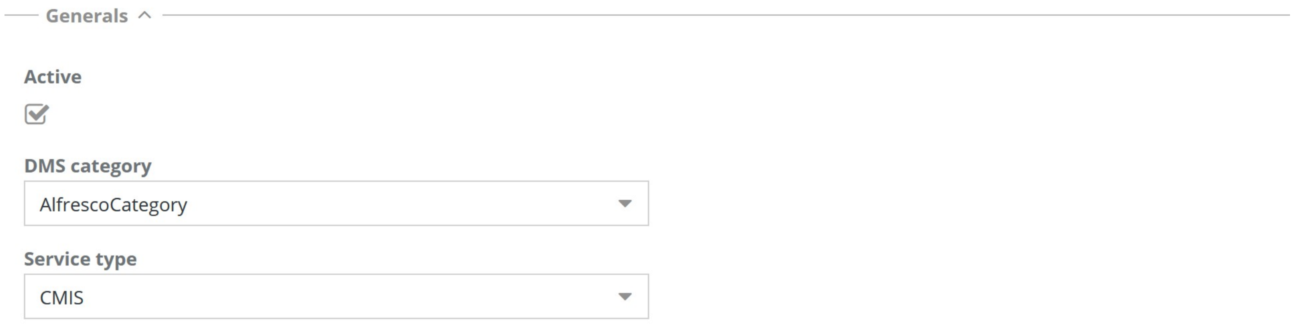
Max file size

Parameters are divided into the three sections described below.

18.3.1. General properties

The following information is required:

- Active: it allows you to activate the interface with the DMS system
- DMS category: default document category, unless stated otherwise for the single classes and processes
- Service type:
  - CMIS (Content Management Interoperability Services), suggested option
  - Microsoft SharePoint Online



Generals

Active

☑

DMS category

AlfrescoCategory

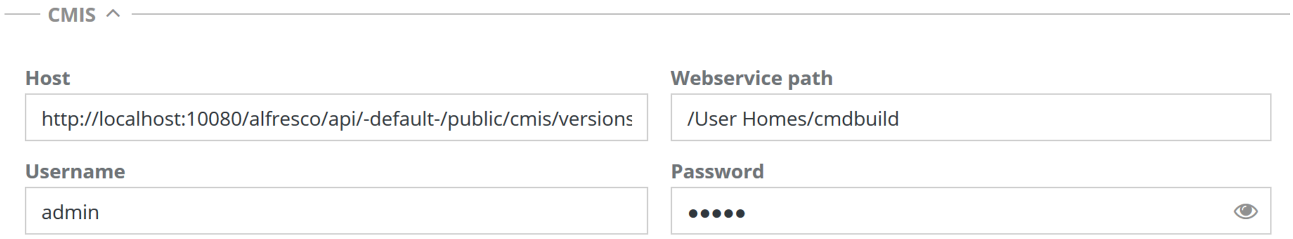
Service type

CMIS

18.3.2. CMIS protocol

The following information is required:

- Host: Alfresco DMS server host
- Webservice path: it corresponds to the path of webservice calls
- Username: DMS username
- Password: user's password



CMIS

Host

http://localhost:10080/alfresco/api/-default-/public/cmis/versions

Webservice path

/User Homes/cmdbuild

Username

admin

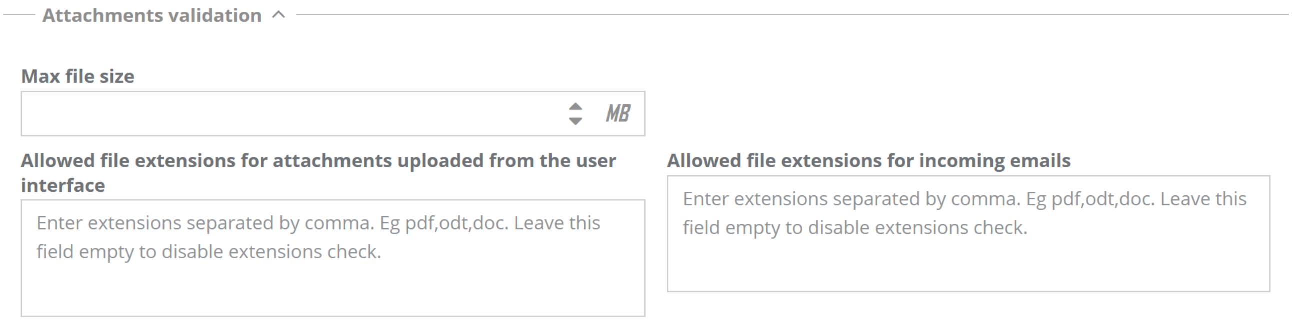
Password

●●●●●

18.3.3. Attachment validation

The following information is required:

- Maximal file seize
- Types of files admitted for loading from user interface: the list of admitted extensions can be specified for operators' loading (the configuration can be overwritten on single document categories)
- Types of files admitted for loading from email attachments: the list of extensions can be specified for automatic loading in the document archive when receiving emails with attachments.



Attachments validation

Max file size

Allowed file extensions for attachments uploaded from the user interface

Enter extensions separated by comma. Eg pdf,odt,doc. Leave this field empty to disable extensions check.

Allowed file extensions for incoming emails

Enter extensions separated by comma. Eg pdf,odt,doc. Leave this field empty to disable extensions check.

19. Navigation trees

A navigation tree is a hierarchical representation of CMDBuild cards: each level corresponds to class cards, linked to the cards of the parent class having a domain with cardinality 1:N.

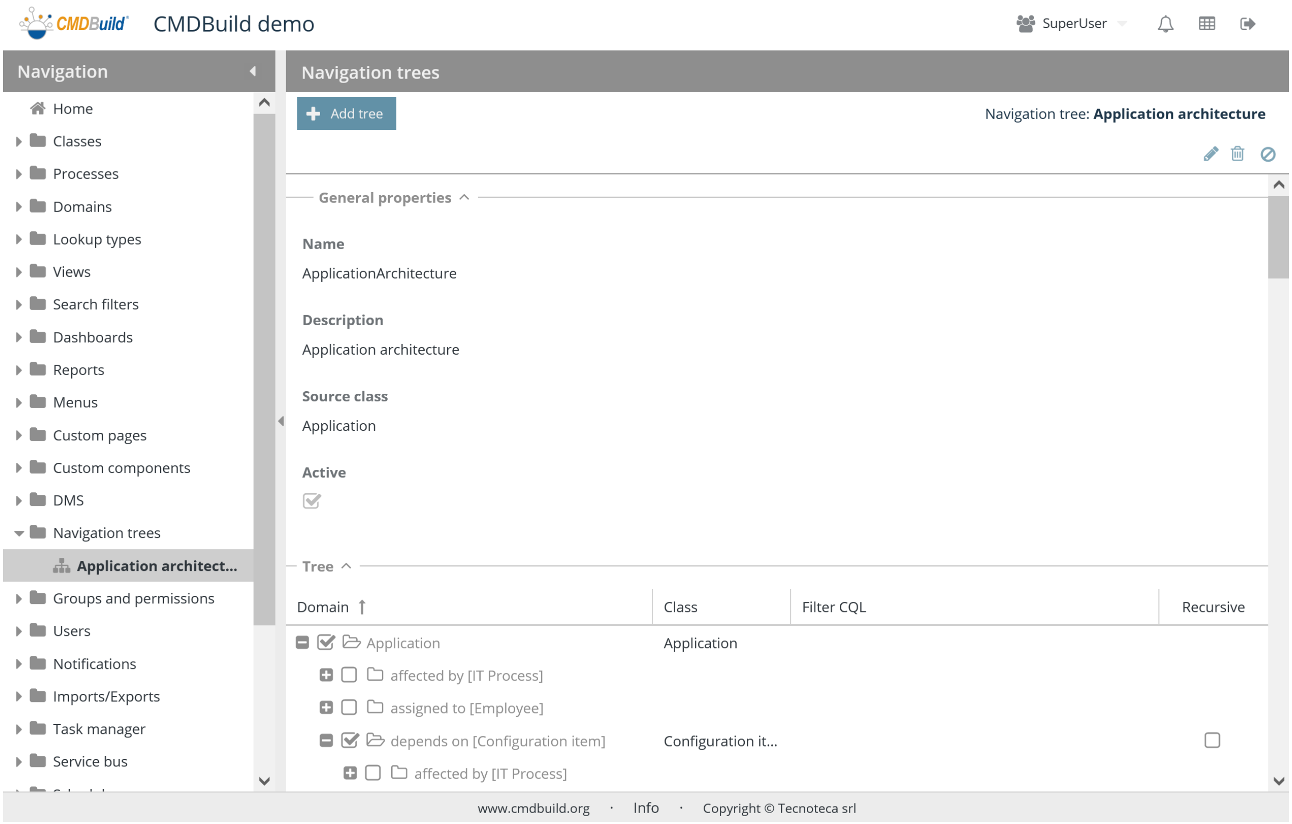
CMDBuild uses navigation trees in various features:

- to define filters that can be applied to the relation graph, excluding some useless relations
- to lead the hierarchical selection of an element through the widget
- to insert a class hierarchy in the navigation menu of the application (can be configured in the menu management)
- to configure the hierarchical navigation menu among items georefered on the map (can be configured in the GIS management)

The use related to the configuration of menus and GIS trees is described in the related chapters of the manual. Below you can read their use in other fields.

19.1. Properties

The function allows you to create new navigation trees and edit the properties of the old ones.



Navigation tree: **Application architecture**

General properties

Name
ApplicationArchitecture

Description
Application architecture

Source class
Application

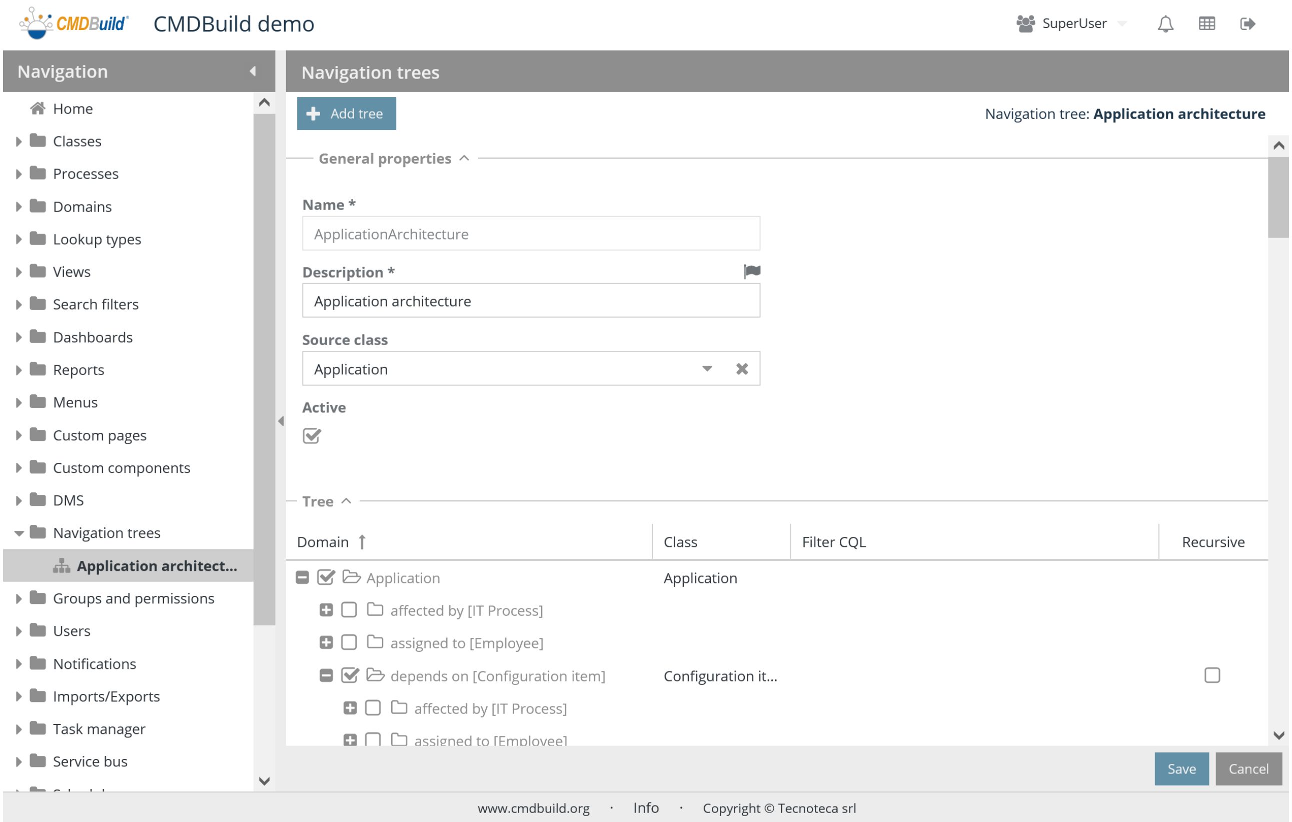
Active
☑

Tree

Domain	Class	Filter CQL	Recursive
⊟ ☑ Application	Application		
⊞ □ affected by [IT Process]			
⊞ □ assigned to [Employee]			
⊟ ☑ depends on [Configuration item]	Configuration it...		□
⊞ □ affected by [IT Process]			

There is the possibility of performing the following operations:

• create a new navigation tree	+ Add tree
• edit the selected navigation tree	✎
• delete the selected navigation tree	🗑
• disable the selected navigation tree (i.e. set the status as non-active)	⊘



Navigation tree: **Application architecture**

General properties

Name *
ApplicationArchitecture

Description *
Application architecture

Source class
Application

Active
☑

Tree

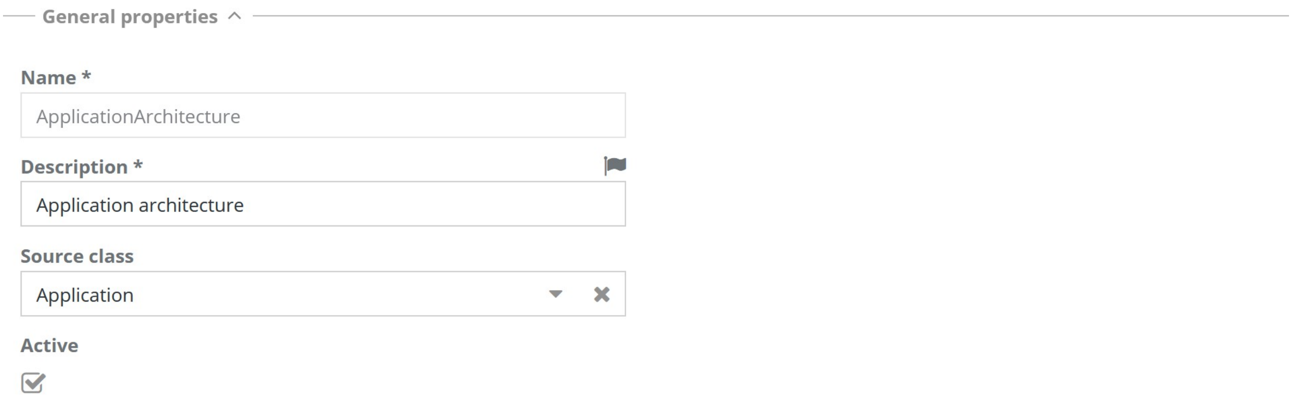
Domain	Class	Filter CQL	Recursive
☑ Application	Application		
☐ affected by [IT Process]			
☐ assigned to [Employee]			
☑ depends on [Configuration item]	Configuration it...		☐
☐ affected by [IT Process]			
☐ assigned to [Employee]			

Each navigation tree needs certain parameters to be compiled (metadata), as described below.

19.1.1. General properties

The following information is required:

- Name: name of navigation tree
- Description: description of the navigation tree
- Original class: root of the tree
- Active: the navigation tree is active and can be used



General properties

Name *
ApplicationArchitecture

Description *
Application architecture

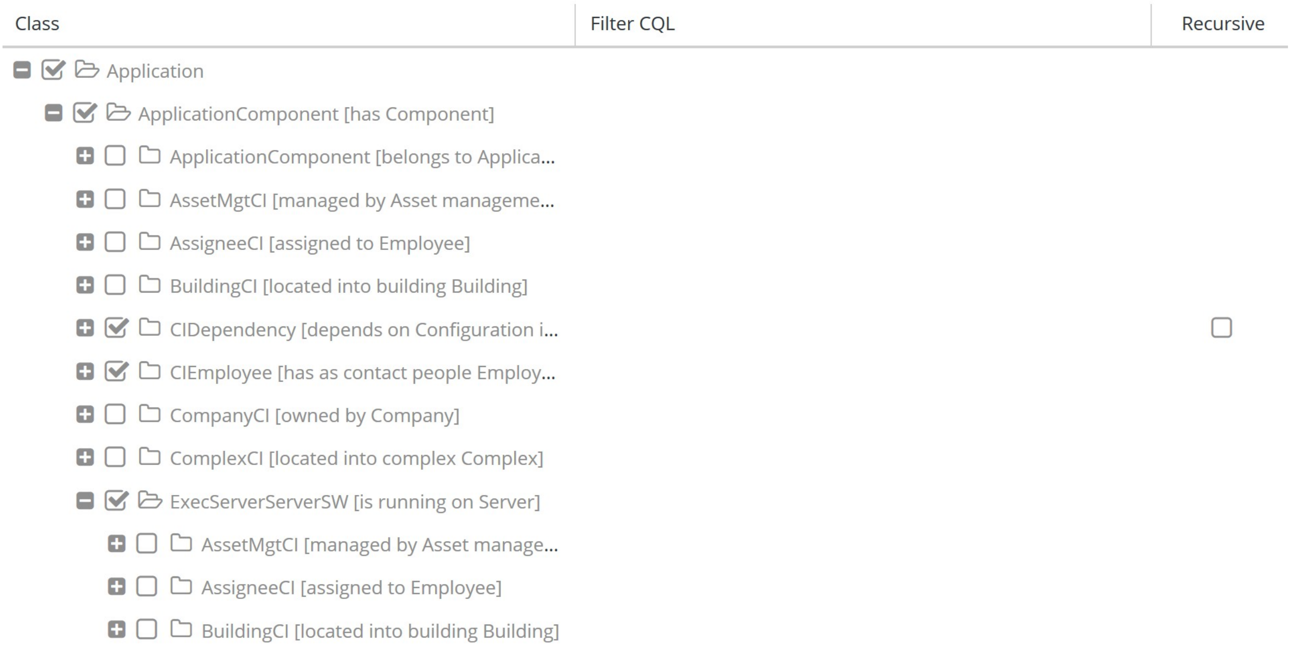
Source class
Application

Active
☑

19.1.2. Navigation tree

Starting from the root class 1:N the class tree is displayed and for each class the following are required:

- Class: it allows you to select the classes of the navigation tree you want to configure
- CQL filter: it allows you to set filters to exclude some elements of the selected classes
- Recursion enabled: it allows you use recursive domains on the same class



Class	Filter CQL	Recursive
⊟ ☑ Application		
⊟ ☑ ApplicationComponent [has Component]		
⊞ □ ApplicationComponent [belongs to Applica...		
⊞ □ AssetMgtCI [managed by Asset manageme...		
⊞ □ AssigneeCI [assigned to Employee]		
⊞ □ BuildingCI [located into building Building]		
⊞ ☑ CIDependency [depends on Configuration i...		□
⊞ ☑ CIEmployee [has as contact people Employ...		
⊞ □ CompanyCI [owned by Company]		
⊞ □ ComplexCI [located into complex Complex]		
⊟ ☑ ExecServerServerSW [is running on Server]		
⊞ □ AssetMgtCI [managed by Asset manage...		
⊞ □ AssigneeCI [assigned to Employee]		
⊞ □ BuildingCI [located into building Building]		

20. Groups and permissions

CMDBuild permissions are based on:

- user groups (roles)
- permissions assigned to a user group on various items: classes (even with restrictions on rows and columns), processes, views, search filters, dashboards, reports, custom pages, import / export templates
- user - group associations

Therefore you can:

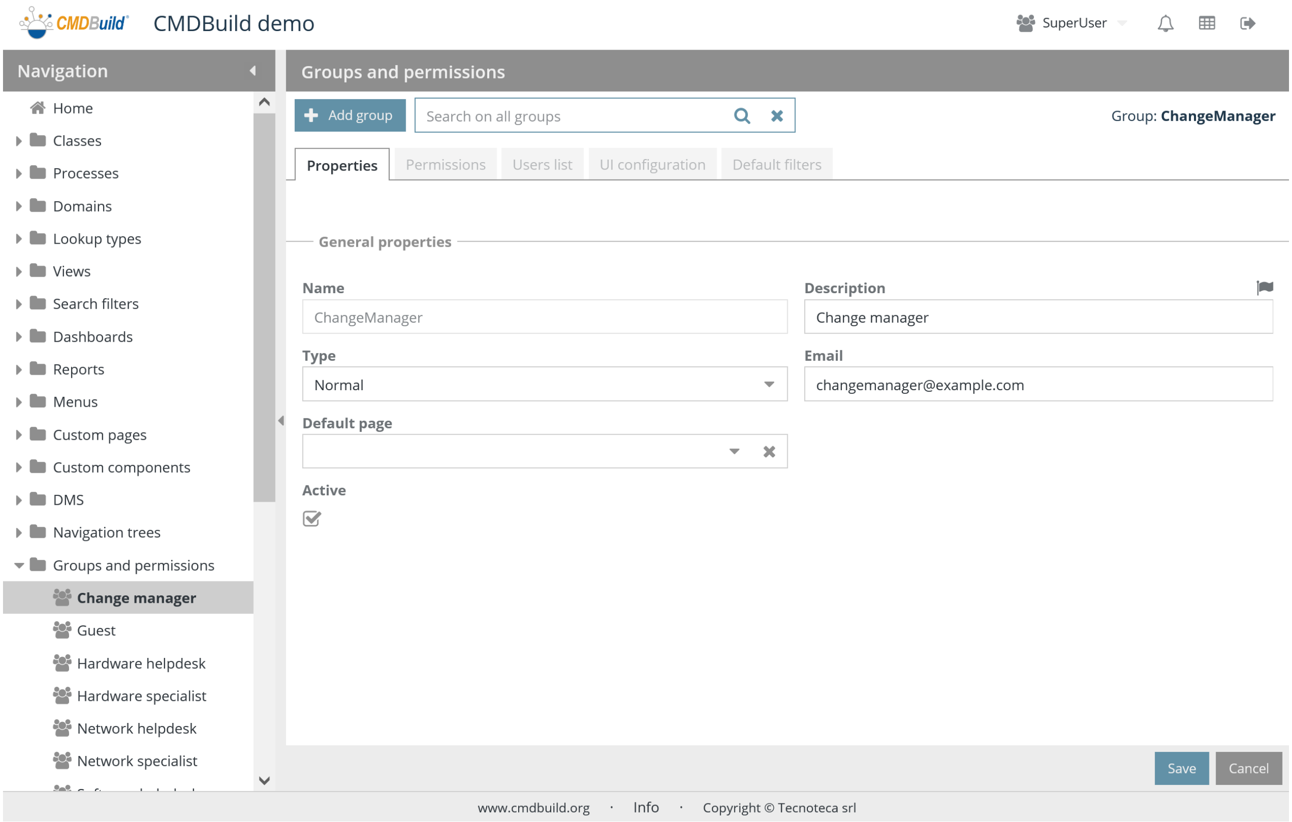
- set user groups with specific permissions for each element configured in the system
- add users to one or more groups; the user inherits permissions from the group

Further functions, gathered in the TAB "UI Setup", allow the definition of additional permissions on elements of the user interface, removing some standard functionalities of CMDBuild and simplifying the interaction interface for some users' groups.

CMDBuild supports also the management of "multitenant", where a set of the CMDB data (section) is reserved to the users belonging to a suborganization of the CMDBuild instance, e.g. a Group Society, a Seat, a Division, etc.

20.1. Properties tab

The function allows you to create new users' groups and edit the properties of the old ones.



Name: ChangeManager

Description: Change manager

Type: Normal

Email: changemanager@example.com

Default page:

Active: ☑

There is the possibility of performing the following operations:

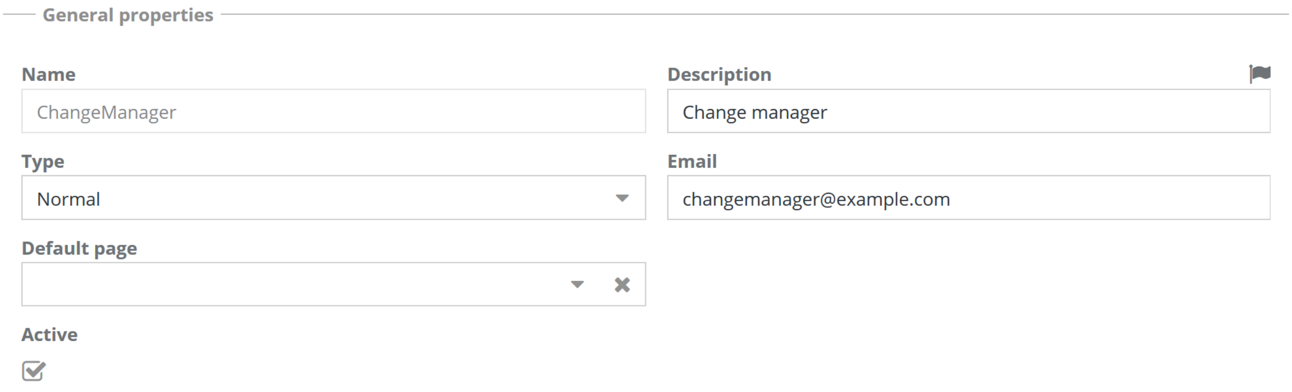
• create a new users' group	+ Add group
• do searches on the configured groups	Search on all groups 🔍 ✖
• edit the selected group	✎
• disable the selected group (i.e. set the status as non-active)	⊘

Each group needs some parameters to be compiled (metadata), as mentioned below.

20.1.1. General properties

The following information is required:

- Activity name
- Description
- Type, which includes the following possibilities:
  - Normal: the most common one. It concerns operators using the application
  - Administrator in read-only mode: read-only access to the Administration Module
  - Limited administrator: possibility of working on the system configuration, except for the ones that edit the data model (classes, processes, domains)
  - User creation administrator: it can create only users and groups, useful especially in case of multitenant, to operate only on the assigned tenants
  - Complete administrator: use of all features of the Administrator Module without limitations
- Email: user's e-mail address, to receive any message provided by the system
- Starting page: it allows you to select the page where the application for the current users' group has to be opened when accessing the Management Module
- Active: the group is active and can be used



General properties

Name: ChangeManager

Description: Change manager

Type: Normal

Email: changemanager@example.com

Default page:

Active ☑

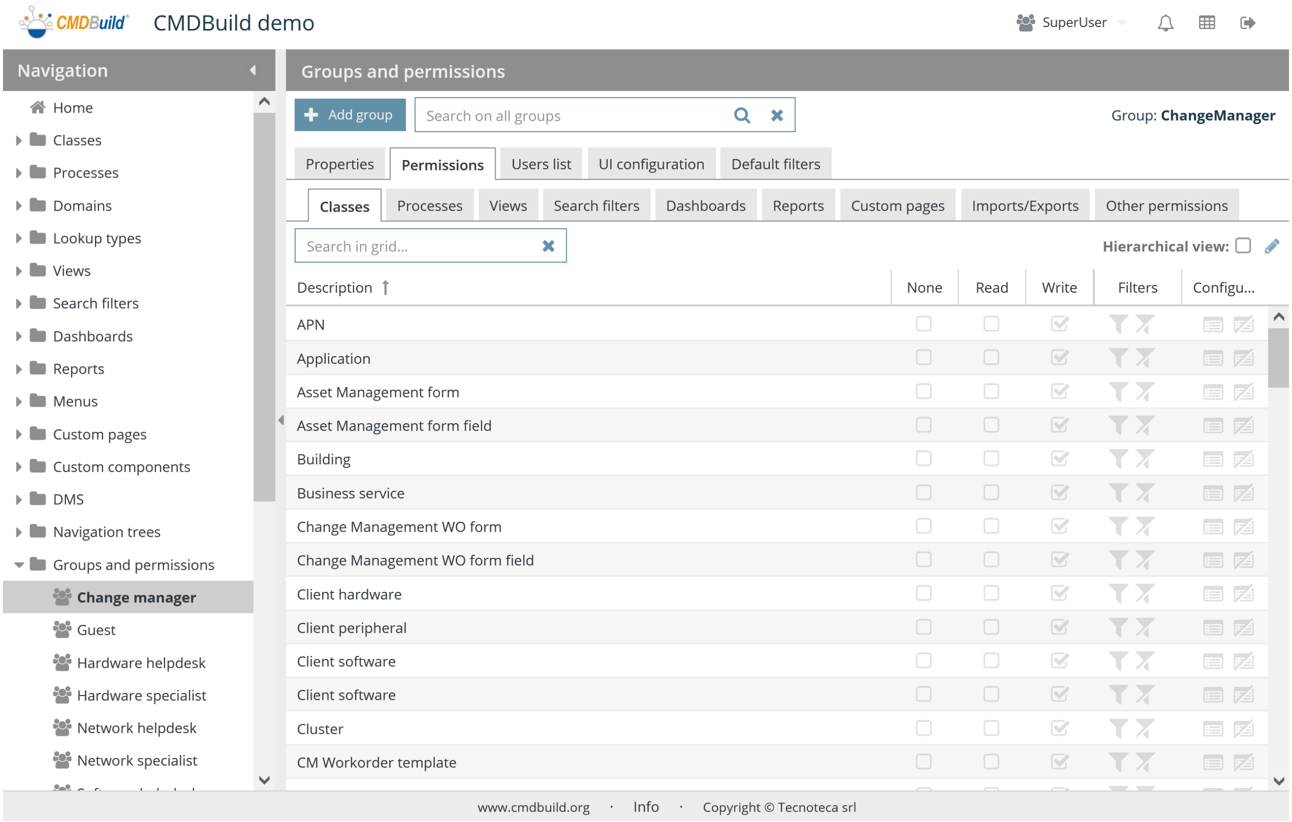
20.2. Permissions tab

The feature allows you to confer access permissions for the current users' group on various items configured in CMDBuild.

Through the second-level tabs described below, permissions can be defined on classes, processes, views, search filters, dashboards, reports, custom pages, import / export templates.

20.2.1. TAB Classes

This tab allows you to define permissions on classes defined in the system.

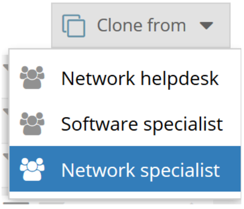


Description	None	Read	Write	Filters	Configu...
APN	□	□	☑		
Application	□	□	☑		
Asset Management form	□	□	☑		
Asset Management form field	□	□	☑		
Building	□	□	☑		
Business service	□	□	☑		
Change Management WO form	□	□	☑		
Change Management WO form field	□	□	☑		
Client hardware	□	□	☑		
Client peripheral	□	□	☑		
Client software	□	□	☑		
Client software	□	□	☑		
Cluster	□	□	☑		
CM Workorder template	□	□	☑		

There is the possibility of performing the following operations:

• do searches on the configured permissions	Search in grid... ✖
• enable the hierarchical view (superclasses, subclasses)	Hierarchical view: □
• edit the list of permissions	✎

If you choose to edit permissions, you can perform the following operations:

• clone permissions from another group	 Clone from Network helpdesk Software specialist Network specialist
• set permissions that are the same for all rows of the list, working on the checks in the columns headers	□
• reduce the permission set on that class, both on rows when a filter is applied and on columns (see below the complete user interface)	
• delete the set restrictions	
• disable some actions on the CMDBuild standard user interface for the selected class (see below the complete user interface)	
• remove the button disabling on the user interface	
• save the changes	Save
• Exit without saving current edits	Cancel

Restriction of permissions on rows and columns

As mentioned above, you can reduce access permissions to the rows or columns of the selected class.

Through the "Filter" icon you can access a popup window that presents two tabs called "Privileges on rows" and "Privileges on columns".

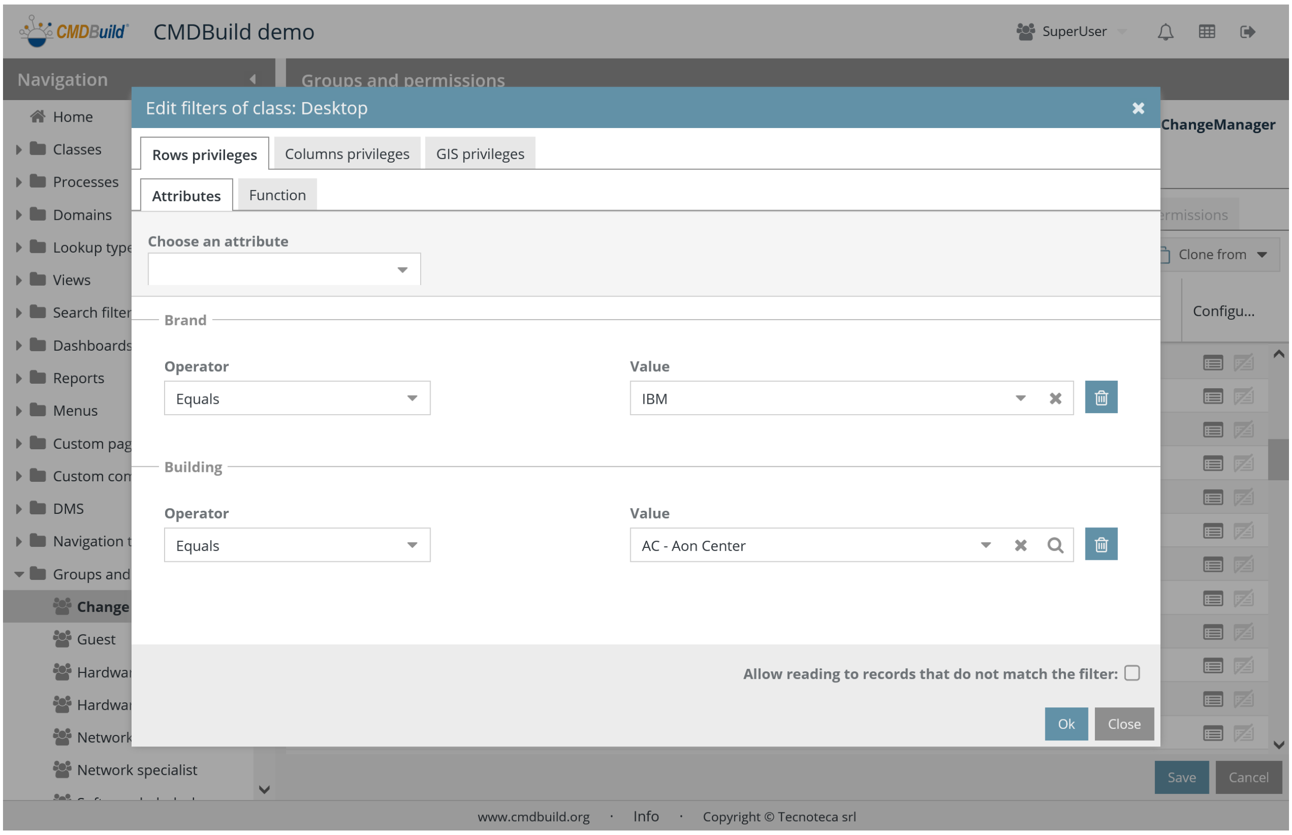
Privileges on rows

The restriction of permissions on rows can be done by defining a filter in two different ways:

- by setting filter criteria on the attributes of the specified source class (Attributes tab), with the same modalities used in the Management Module for the advanced search
- by using a pre-defined PostgreSQL function, that will be created with the following criteria
  - it must report the comment "TYPE: function"
  - it has to include the input and output parameters, clearly showing the name of each of them
  - the input and output parameters have to be chosen among the following ones: “character varying”, “boolean”, “integer”, “numeric”, “double precision”, “date”, “time”, “timestamp”, “text” (not “bigint”)
  - in case that the function becomes more "tuple" than the output values, you should use the syntax "Returns setof record”

Here are two samples of defining the filter.

Through the attribute filter set with the same modalities used in the Management Module for the advanced search:



Edit filters of class: Desktop

Rows privileges | Columns privileges | GIS privileges

Attributes | Function

Choose an attribute

Brand

Operator: Equals

Value: IBM

Building

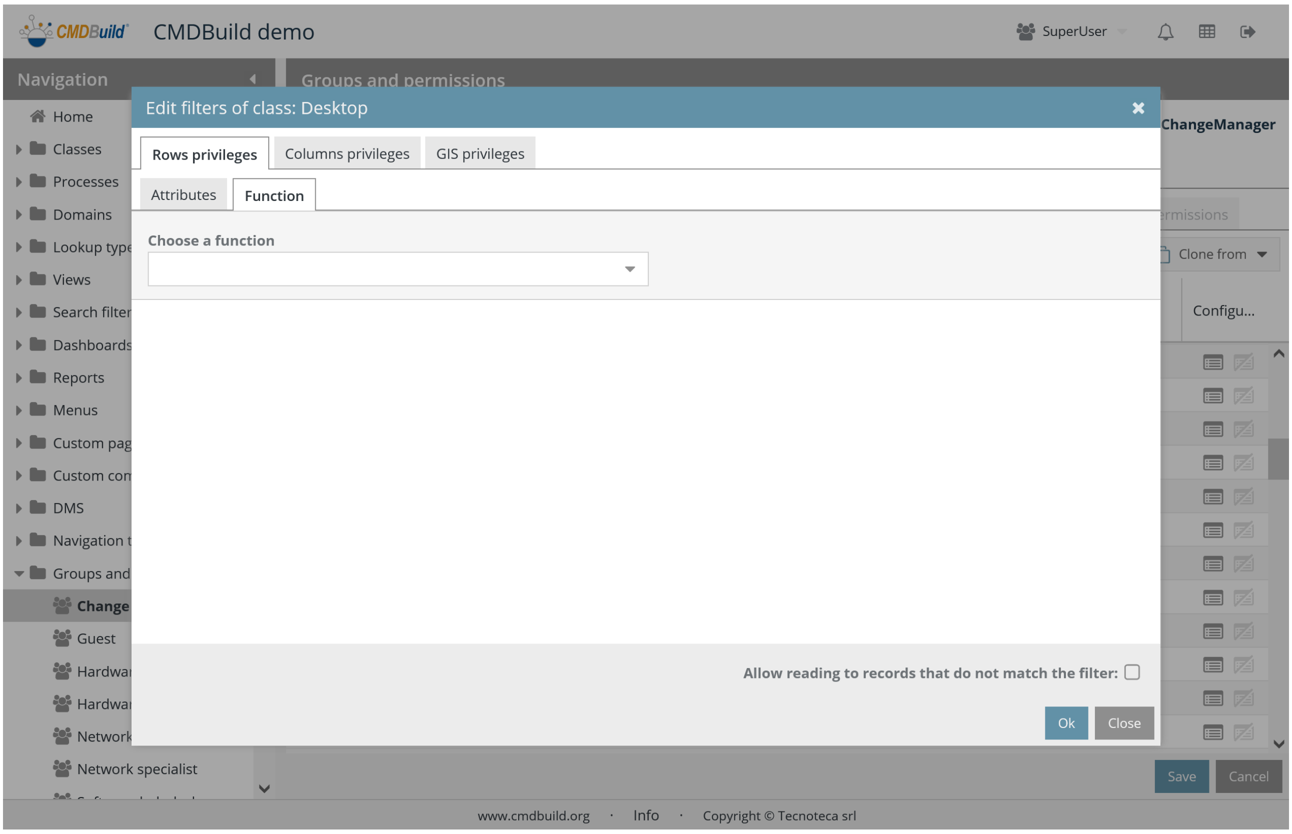
Operator: Equals

Value: AC - Aon Center

Allow reading to records that do not match the filter: □

Ok | Close

Through PostgreSQL function:



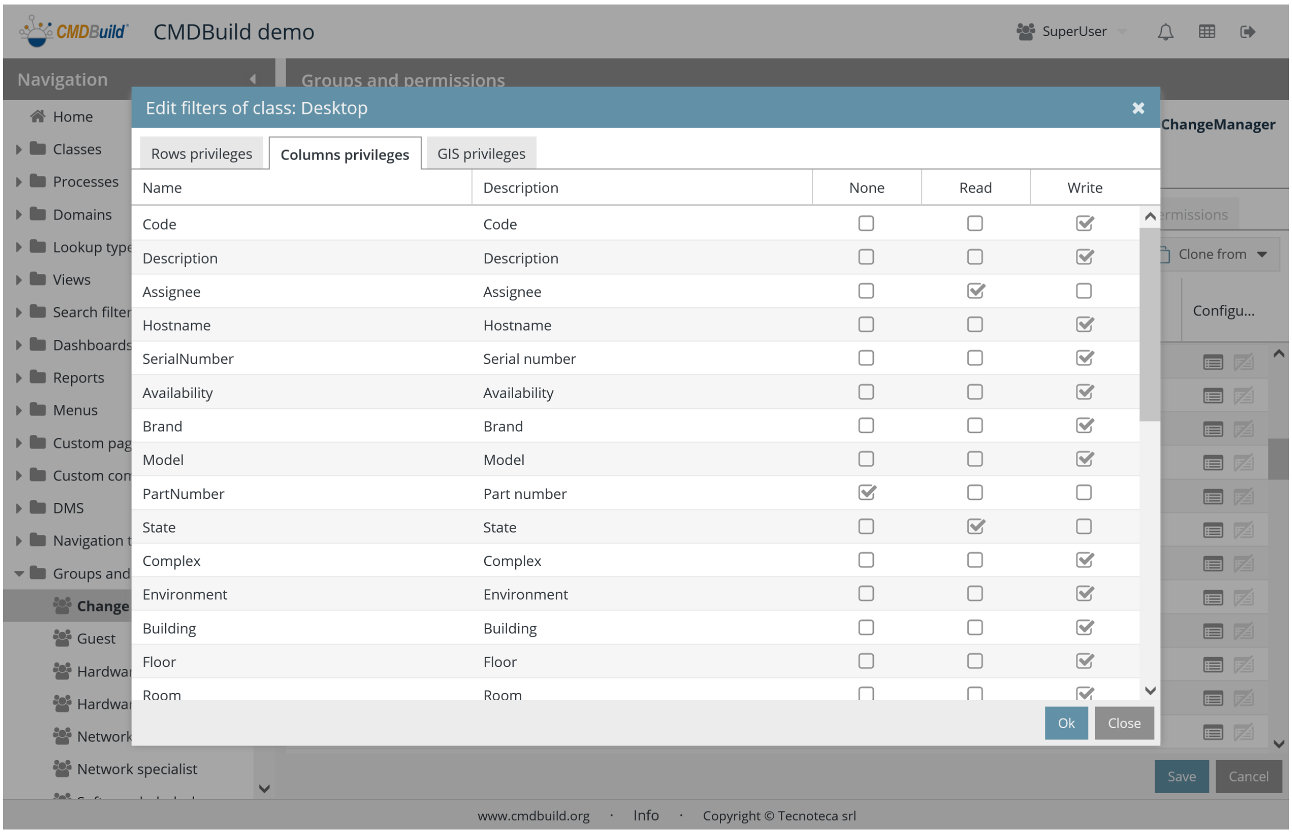
In both cases, if the permission is write-type, you can specify if the rows not included in the filter can be displayed in only-read mode or not displayed at all.

Privileges on columns

The restriction of permissions on columns can be done in the following ways:

- not-visible attribute
- only-visible attribute
- even-editable attribute

Here's an example of screenshot:



Edit filters of class: Desktop

Rows privileges | Columns privileges | GIS privileges

Name	Description	None	Read	Write
Code	Code	☐	☐	☑
Description	Description	☐	☐	☑
Assignee	Assignee	☐	☑	☐
Hostname	Hostname	☐	☐	☑
SerialNumber	Serial number	☐	☐	☑
Availability	Availability	☐	☐	☑
Brand	Brand	☐	☐	☑
Model	Model	☐	☐	☑
PartNumber	Part number	☑	☐	☐
State	State	☐	☑	☐
Complex	Complex	☐	☐	☑
Environment	Environment	☐	☐	☑
Building	Building	☐	☐	☑
Floor	Floor	☐	☐	☑
Room	Room	☐	☐	☑

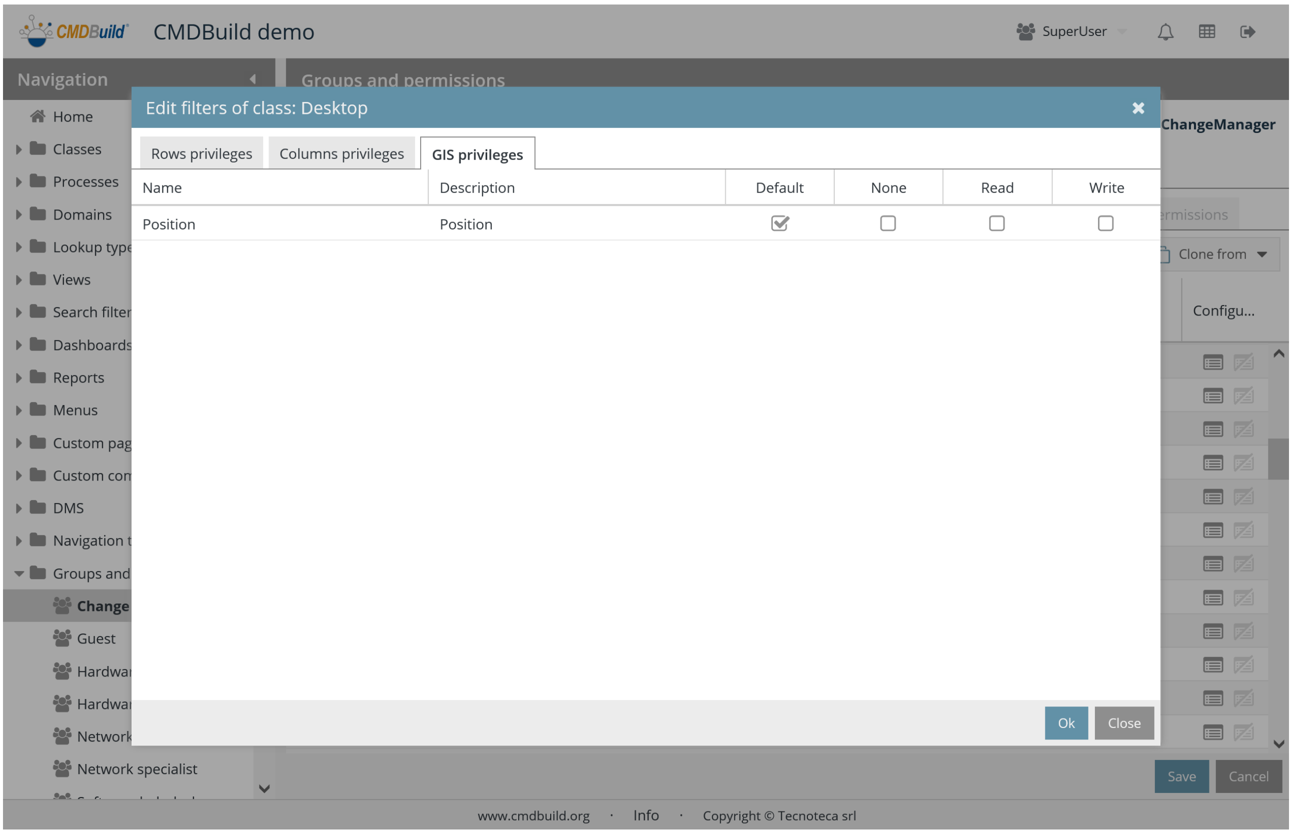
Ok Close

GIS privileges

The restriction of permissions on geographical attributes can be done in the following ways:

- not-visible attribute
- only-visible attribute
- even-editable attribute

Here's an example of screenshot:



Edit filters of class: Desktop

Rows privileges | Columns privileges | GIS privileges

Name	Description	Default	None	Read	Write
Position	Position	☑	□	□	□

GUI buttons control

You can enable (default) or disable all the permissions linked to insert, edit, delete, clone, graph access and print actions for the selected class from the user interface in the CMDBuild data Management Module.

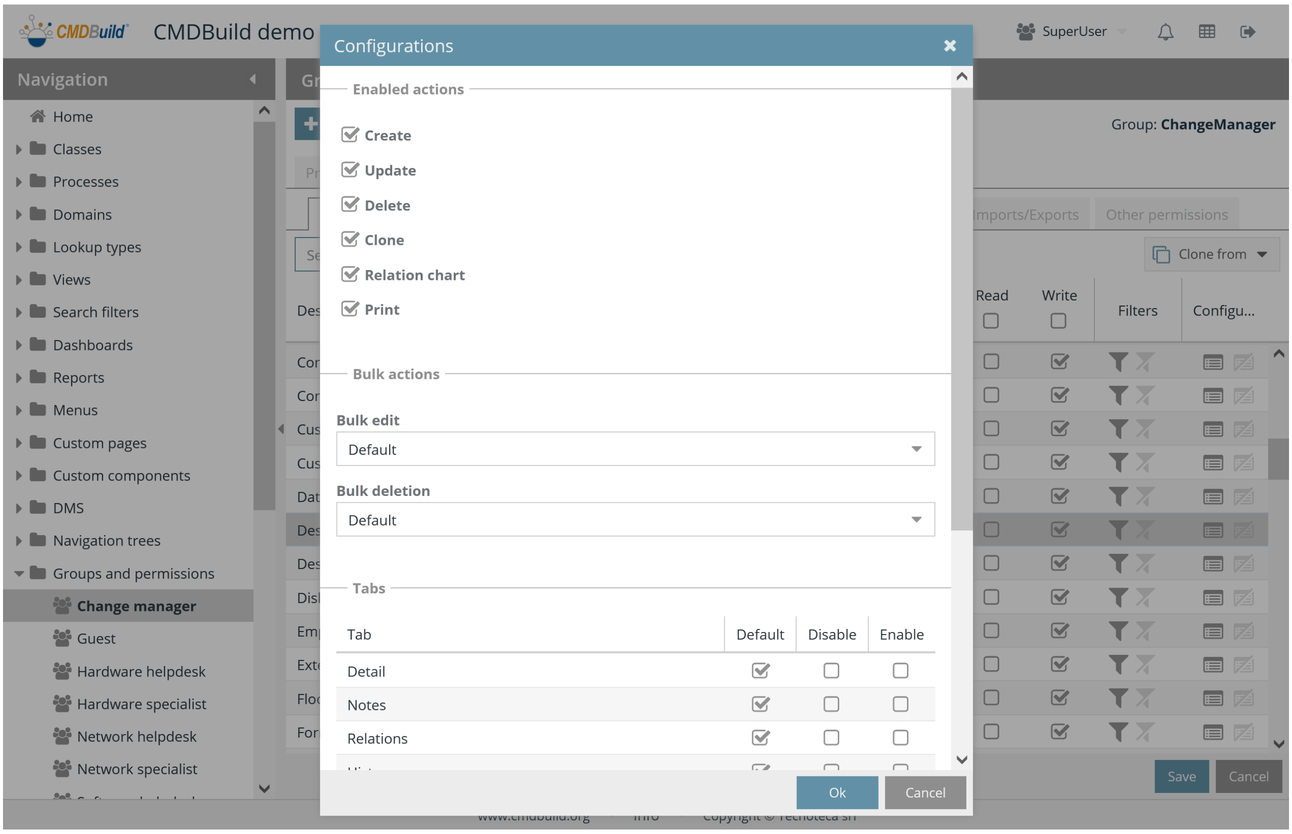
It is also possible to enable (default) or disable all the permissions linked to the massive edit and massive delete operations on the selected class from the user interface in the CMDBuild data Management Module.

In conclusion you can also enable (default) or disable all the permissions linked to the Details, Notes, Relations, History, Email, Attachments, Scheduling TABs for the selected class from the user interface in the CMDBuild data Management Module.

Finally, you can define permissions on the attachments available in the class, according to the DMS category.

This option is only related to the CMDBuild standard GUI and has no consequences on the operations performed through other user interfaces or webservices.

Here's the screenshot of the pop-up window provided by the system.



Configurations

Enabled actions

☑ Create

☑ Update

☑ Delete

☑ Clone

☑ Relation chart

☑ Print

Bulk actions

Bulk edit: Default

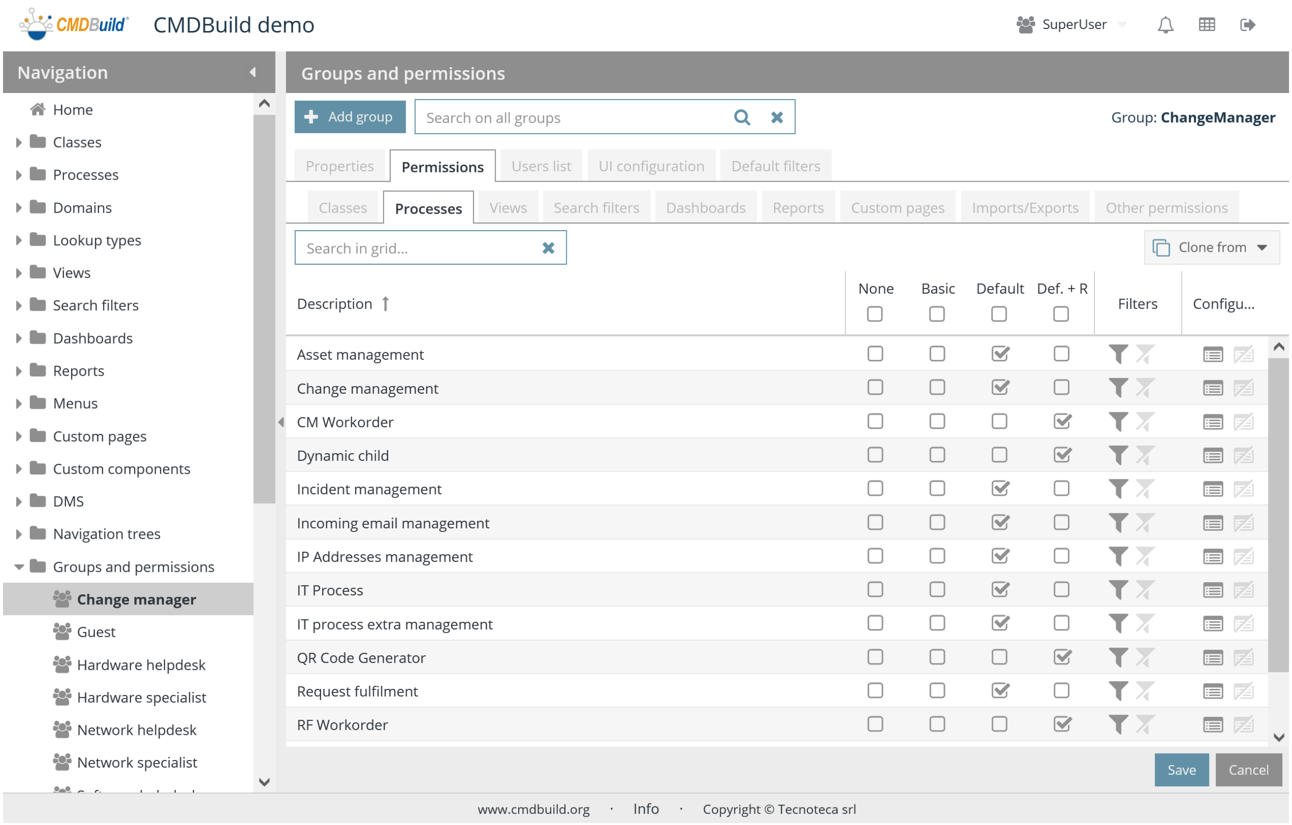
Bulk deletion: Default

Tabs

Tab	Default	Disable	Enable
Detail	☑	□	□
Notes	☑	□	□
Relations	☑	□	□

20.2.2. Processes TAB

This tab allows you to define permissions defined in the system.



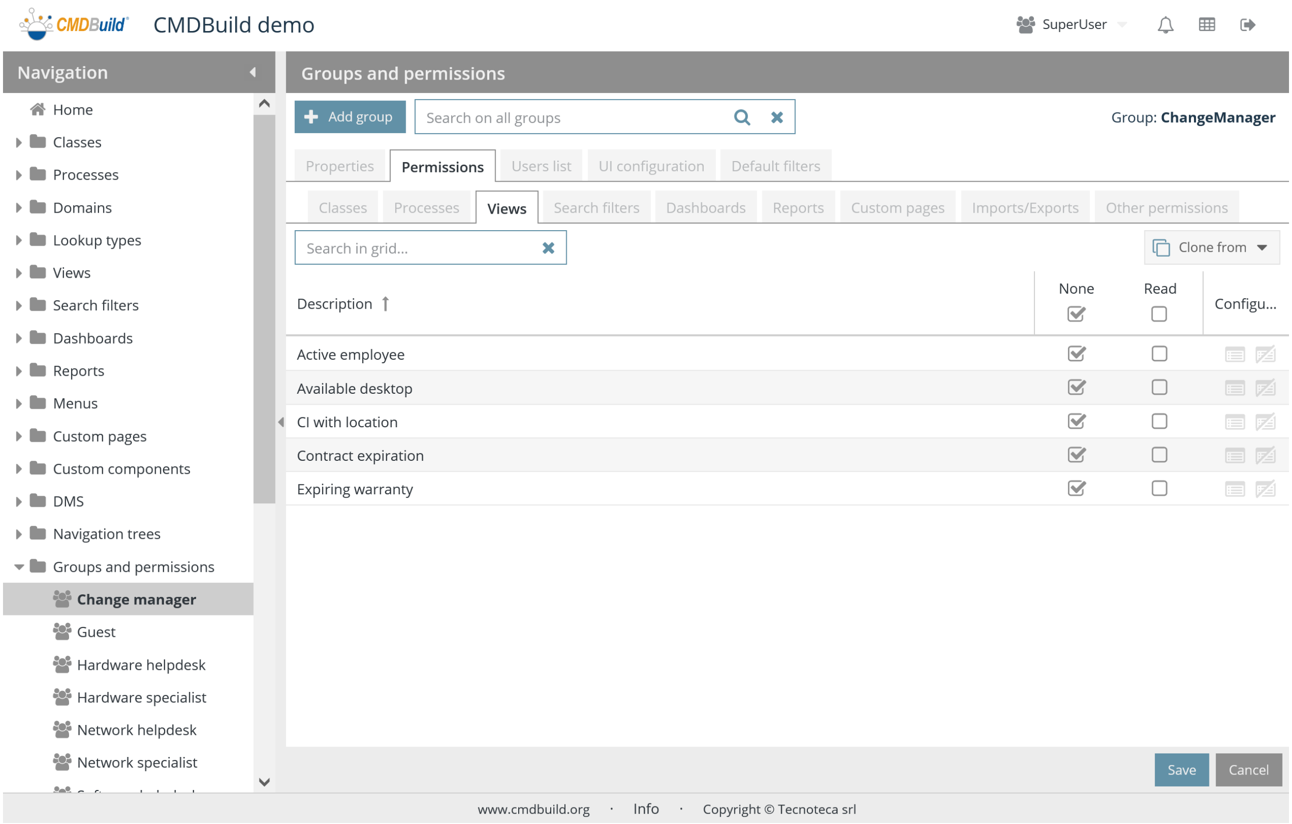
Description	None □	Basic □	Default □	Def. + R □	Filters	Configu...
Asset management	□	□	☑	□		
Change management	□	□	☑	□		
CM Workorder	□	□	□	☑		
Dynamic child	□	□	□	☑		
Incident management	□	□	☑	□		
Incoming email management	□	□	☑	□		
IP Addresses management	□	□	☑	□		
IT Process	□	□	☑	□		
IT process extra management	□	□	☑	□		
QR Code Generator	□	□	□	☑		
Request fulfilment	□	□	☑	□		
RF Workorder	□	□	□	☑		

The same features described for classes are available, with the following differences:

- permission types are not "None", "Read", "Write". They are:
  - None: the process is not visible to the current group
  - Base: the process can be managed for the current group only in case the ongoing activity belongs to it (in contrast to the following type which keeps visible also the processes previously executed)
  - Default: permissions derived from the XPDL descriptor of the process are applied (a group of users can edit / advance a process if the current activity is in the "lane" of the current group, they can view a process if they edited / advanced it in a previous step)
  - Default + Read: besides permissions derived by the XPDL descriptor, the users' group has Read permissions on the process
- the disabling option is not available for certain buttons on the user interface of the Management Module

20.2.3. TAB Views

This tab allows you to define permissions on views defined in the system.



Group: **ChangeManager**

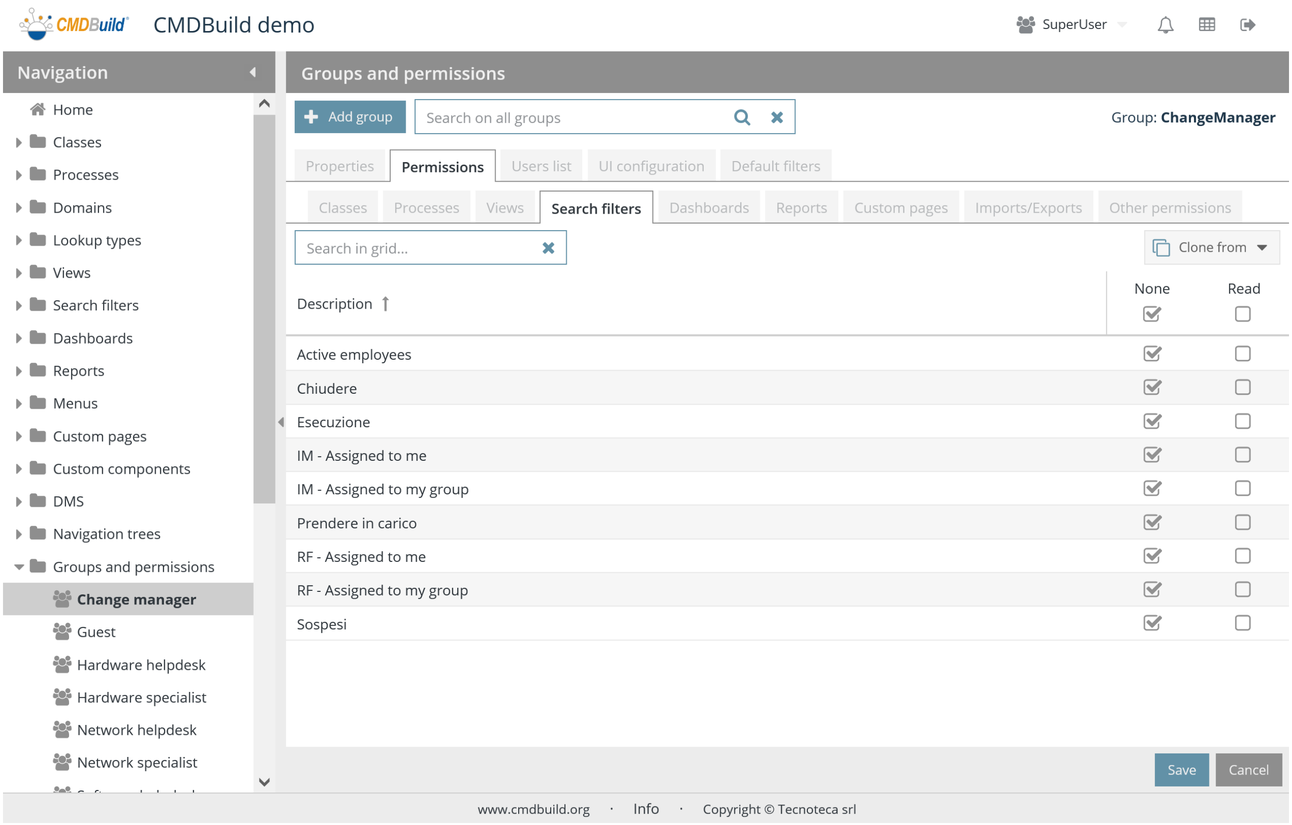
Description	None ☑	Read □	Configu...
Active employee	☑	□	
Available desktop	☑	□	
CI with location	☑	□	
Contract expiration	☑	□	
Expiring warranty	☑	□	

The types of permissions are as follows:

- None: the view is not visible to the current group
- Read: the view is visible to the current group

20.2.4. TAB Search filters

The search and filters tab allows you to define permissions on search filters defined in the system.



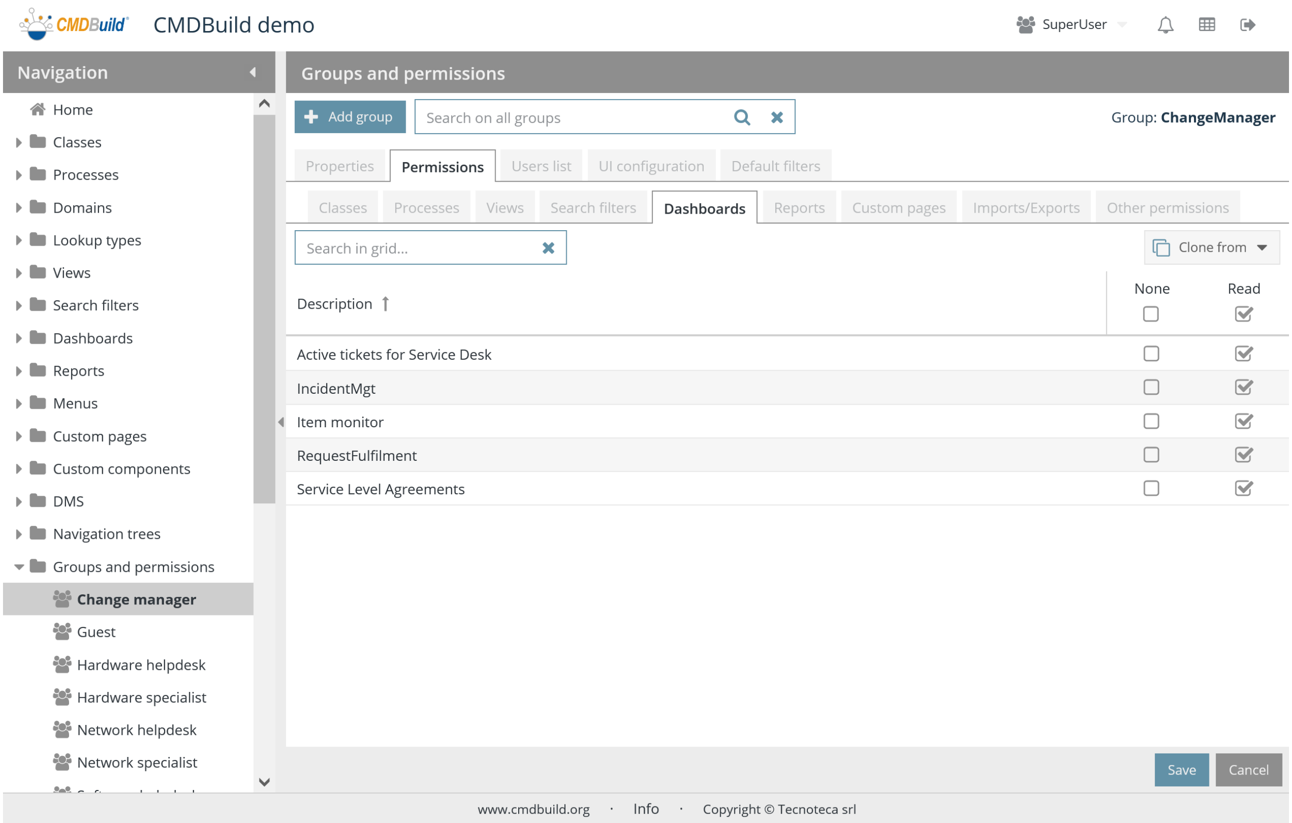
Description ↑	None ☑	Read □
Active employees	☑	□
Chiudere	☑	□
Esecuzione	☑	□
IM - Assigned to me	☑	□
IM - Assigned to my group	☑	□
Prendere in carico	☑	□
RF - Assigned to me	☑	□
RF - Assigned to my group	☑	□
Sospesi	☑	□

The types of permissions are as follows:

- None: the search filter is not visible to the current group
- Read: the search filter is visible to the current group

20.2.5. Dashboard tab

This tab allows you to define permissions on views defined in the system.



Group: **ChangeManager**

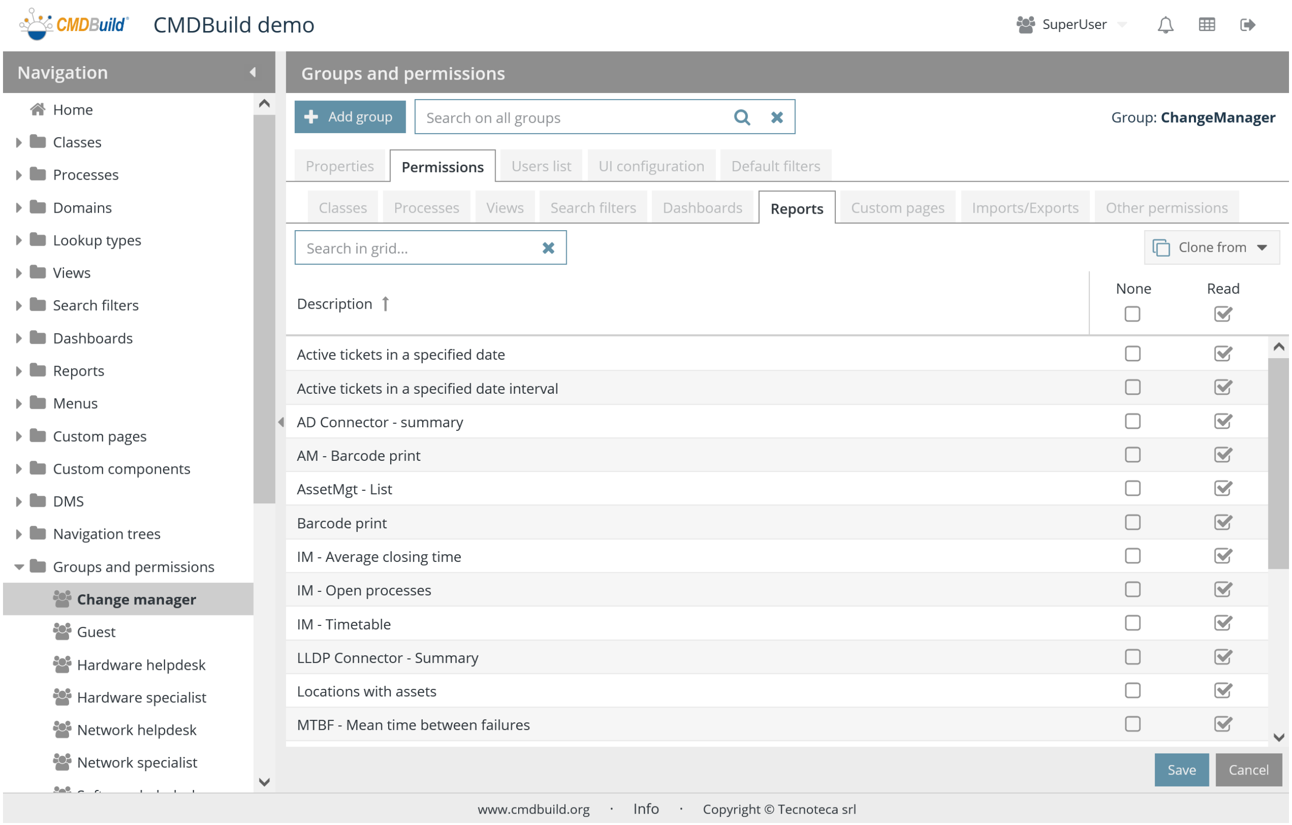
Description	None □	Read ☑
Active tickets for Service Desk	□	☑
IncidentMgt	□	☑
Item monitor	□	☑
RequestFulfilment	□	☑
Service Level Agreements	□	☑

The types of permissions are as follows:

- None: the dashboard is not visible to the current group
- Read: the dashboard is visible to the current group

20.2.6. Report tab

This tab allows you to define permissions on reports defined in the system.



Group: **ChangeManager**

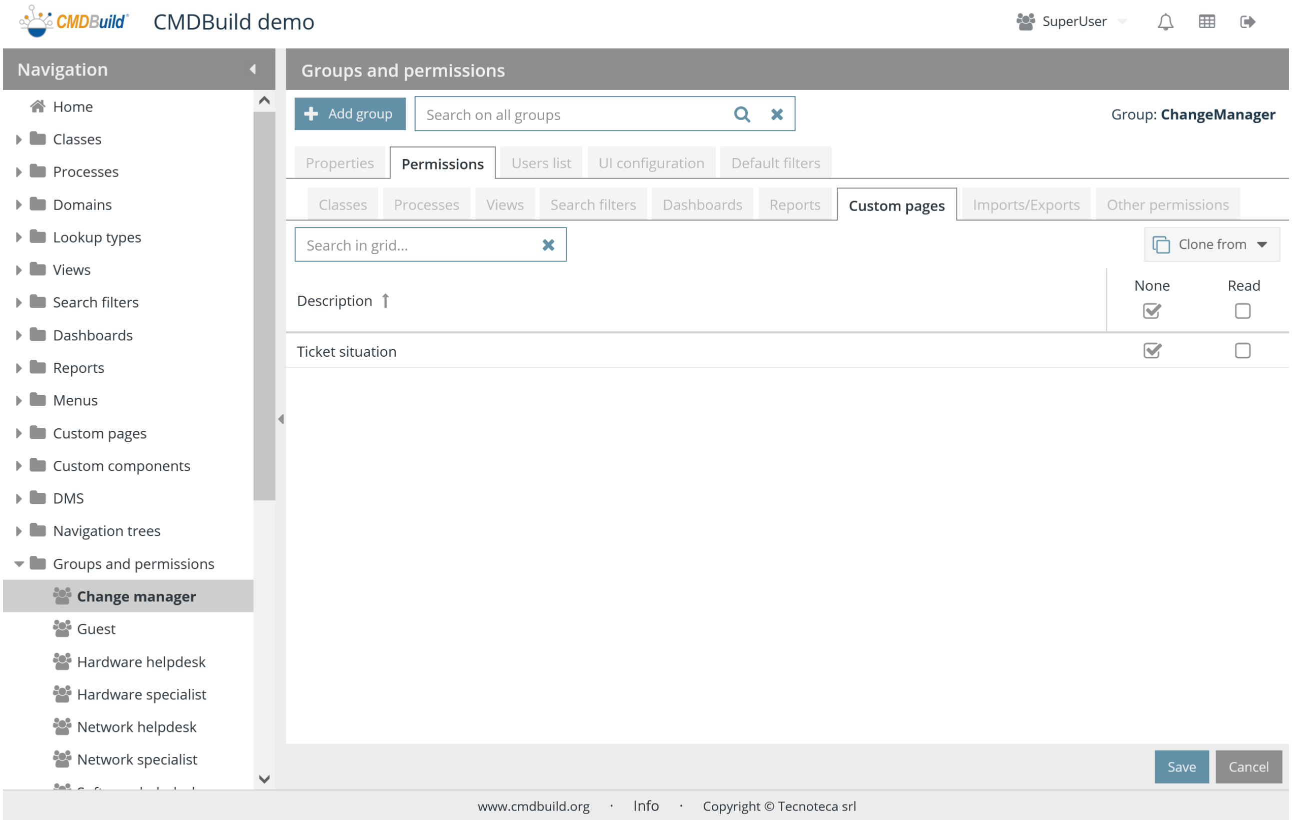
Description	None □	Read ☑
Active tickets in a specified date	□	☑
Active tickets in a specified date interval	□	☑
AD Connector - summary	□	☑
AM - Barcode print	□	☑
AssetMgt - List	□	☑
Barcode print	□	☑
IM - Average closing time	□	☑
IM - Open processes	□	☑
IM - Timetable	□	☑
LLDP Connector - Summary	□	☑
Locations with assets	□	☑
MTBF - Mean time between failures	□	☑

The types of permissions are as follows:

- None: the report is not visible to the current group
- Read: the report is visible to the current group

20.2.7. TAB Custom pages

This tab allows you to define permissions on custom pages defined in the system.



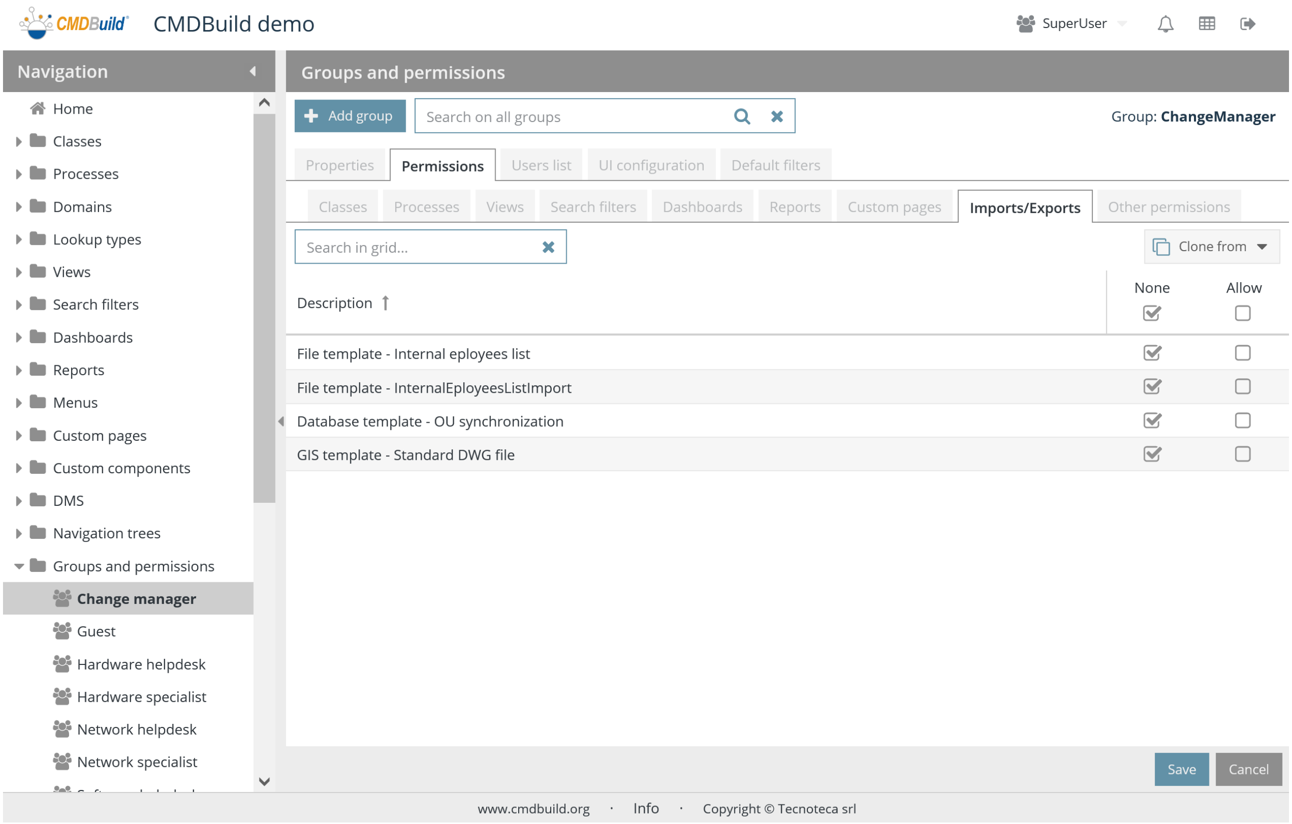
Description ↑	None ☑	Read □
Ticket situation	☑	□

The types of permissions are as follows:

- None: the custom page is not visible to the current group
- Read: the custom page is visible to the current group

20.2.8. Import / Export tab

The Import / Export tab allows you to define permissions on the Import / Export templates configured in the system.



Group: **ChangeManager**

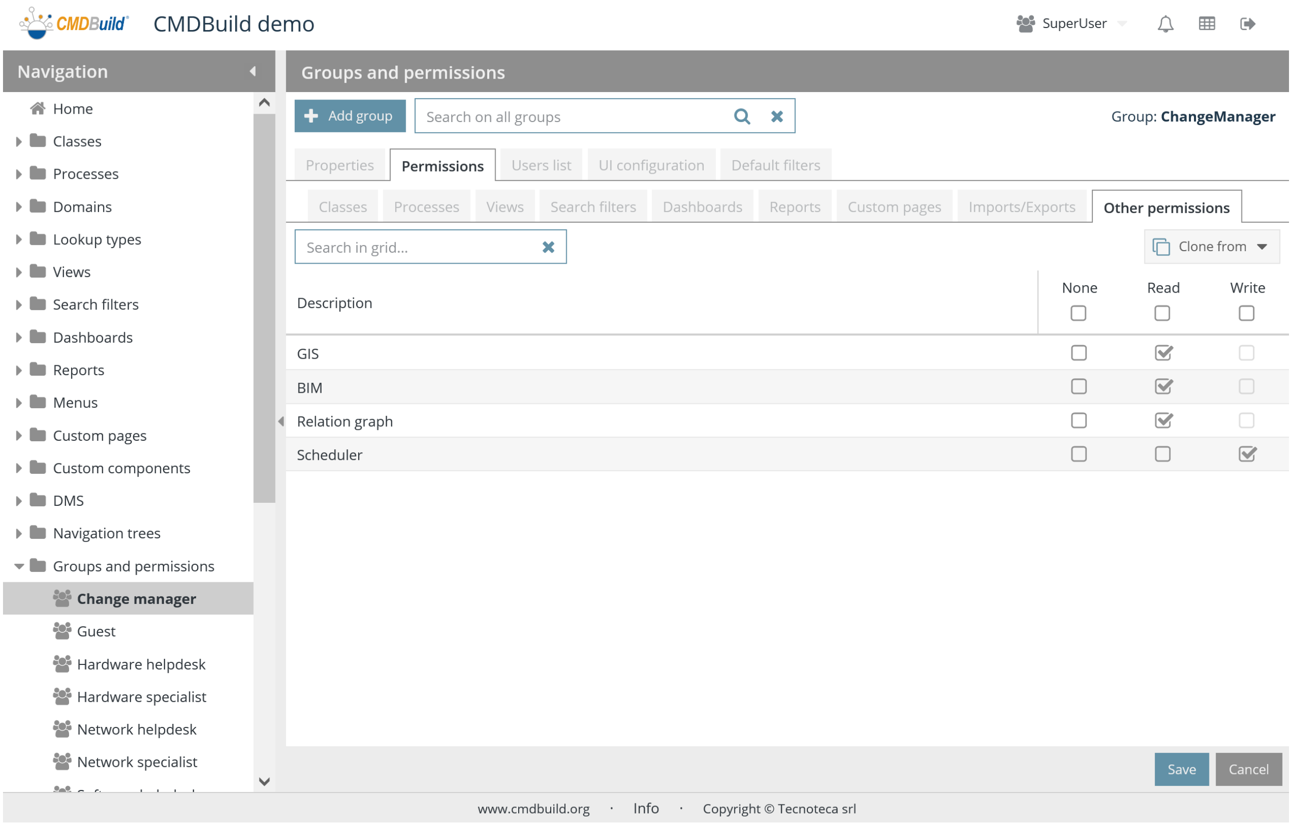
Description	None ☑	Allow □
File template - Internal eployees list	☑	□
File template - InternalEployeesListImport	☑	□
Database template - OU synchronization	☑	□
GIS template - Standard DWG file	☑	□

The types of permissions are as follows:

- None: the template is not visible to the current group
- Enabled: the template is usable by the current group

20.2.9. Other permissions TAB

This tab allows you to define the permissions on further features: GIS, BIM, Relation graph, Scheduler.



Group: **ChangeManager**

Description	None □	Read □	Write □
GIS	□	☑	□
BIM	□	☑	□
Relation graph	□	☑	□
Scheduler	□	□	☑

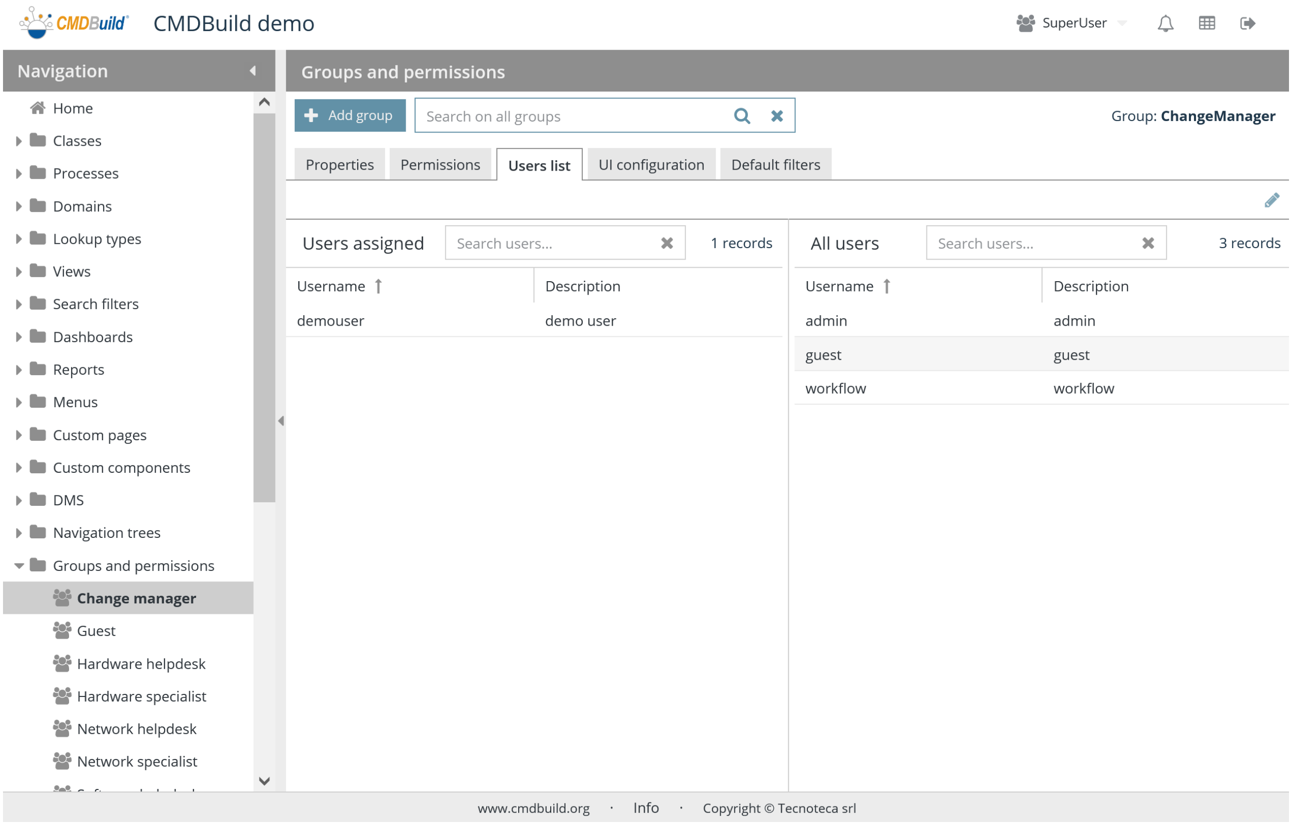
The types of permissions are as follows:

- None: the feature is not visible to the current group
- Read: the feature is visible to the current group
- Write: the feature allows writing operations to the current group

20.3. Users' list tab

This function allows you to associate one or more users to the current group.

You can drag and drop users from the list of available user (on the right) to the list of users in the group (on the left).



Users assigned — Search users... — 1 records

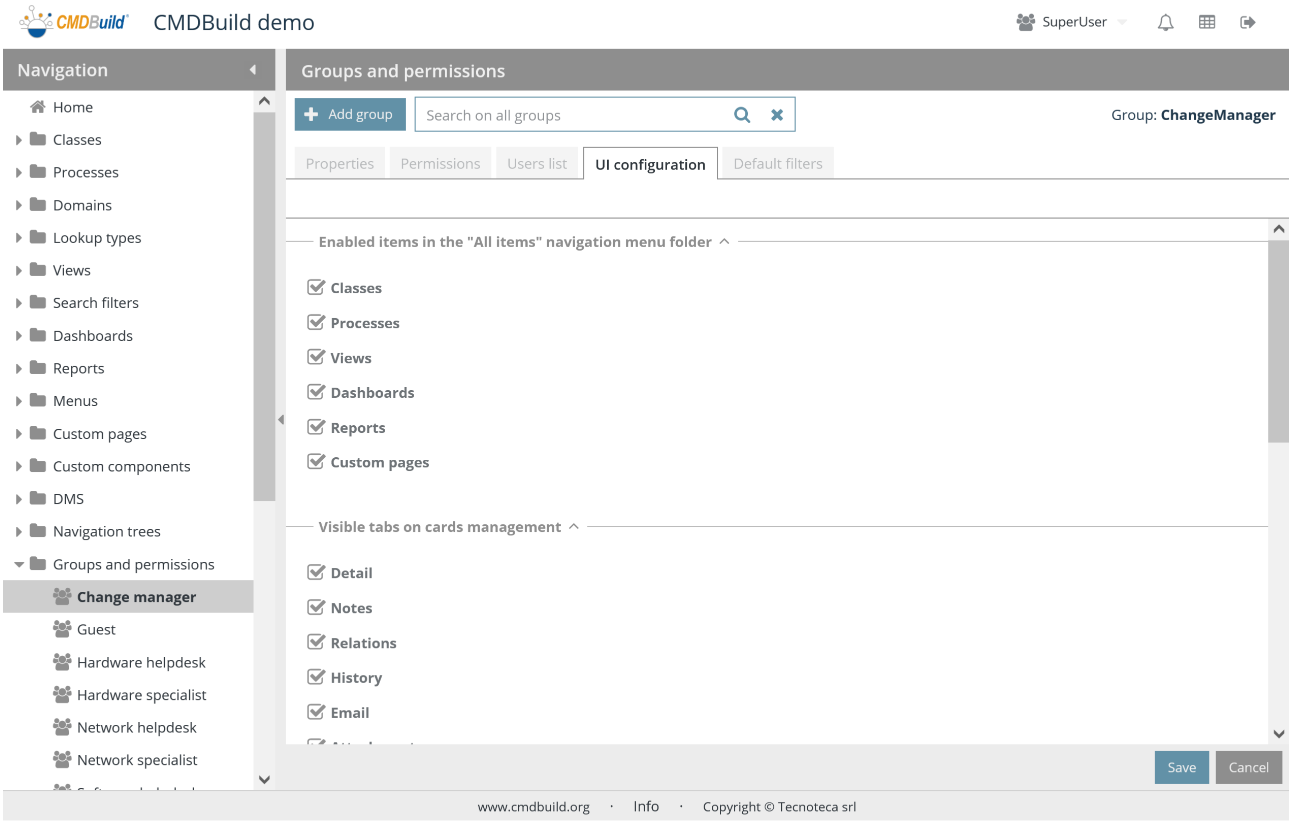
Username	Description
demouser	demo user

All users — Search users... — 3 records

Username	Description
admin	admin
guest	guest
workflow	workflow

20.4. TAB UI setup

This function allows you to associate some permissions to the current group on UI elements of the Management Module.



Group: **ChangeManager**

Enabled items in the "All items" navigation menu folder

☑ Classes

☑ Processes

☑ Views

☑ Dashboards

☑ Reports

☑ Custom pages

Visible tabs on cards management

☑ Detail

☑ Notes

☑ Relations

☑ History

☑ Email

It is possible to enable or disable, as described below:

- the entries of the submenu "All elements" of the navigation menu
- the available TABs for the card management
- the available TABs for the process management
- massive actions
- CMDBuild chat

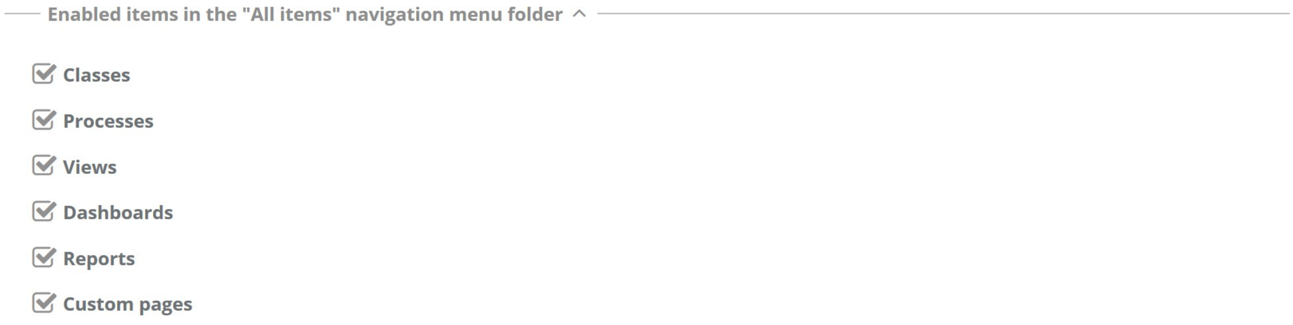
Here follow all the details for each option.

20.4.1. Menu "All elements"

You can disable the following entries:

- Cards
- Processes
- Views

- Dashboard
- Report
- Custom pages



Enabled items in the "All items" navigation menu folder

☑ Classes

☑ Processes

☑ Views

☑ Dashboards

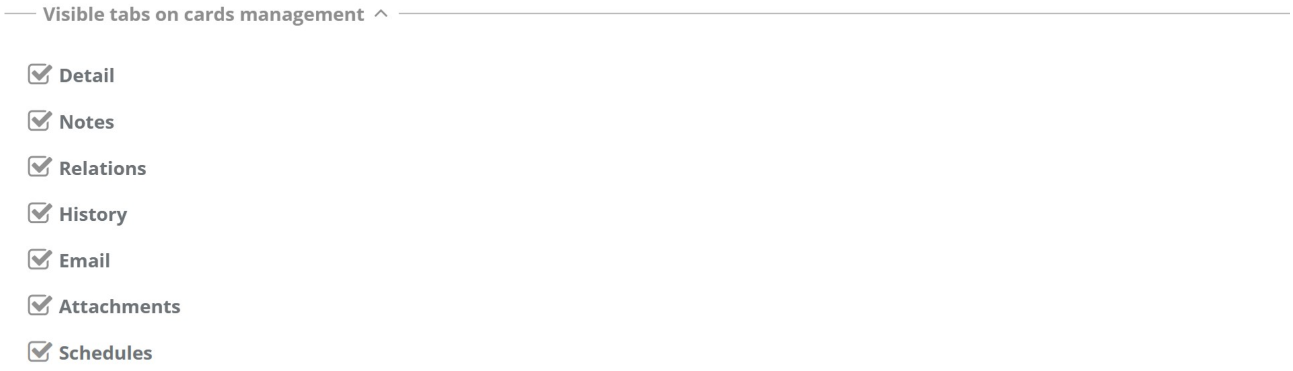
☑ Reports

☑ Custom pages

20.4.2. Classes management

You can disable the following tabs:

- Details
- Notes
- Relations
- History
- Email
- Attachments
- Schedule



Visible tabs on cards management

☑ Detail

☑ Notes

☑ Relations

☑ History

☑ Email

☑ Attachments

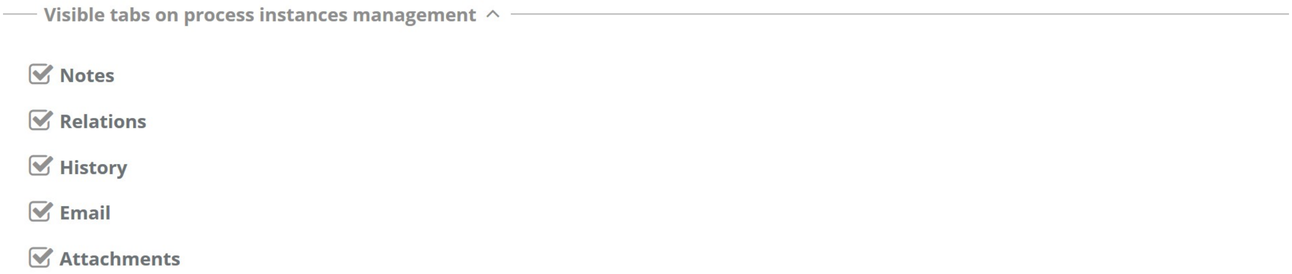
☑ Schedules

20.4.3. Process management

You can disable the following tabs:

- Notes
- Relations
- History
- Email

- Attachments



Visible tabs on process instances management

☑ Notes

☑ Relations

☑ History

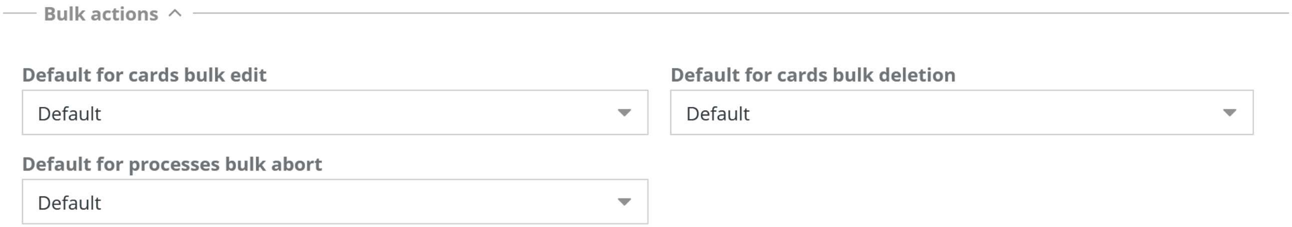
☑ Email

☑ Attachments

20.4.4. Massive actions

For the current group you can set the following configurations for a massive modification and deletion on classes and for a massive abort on processes:

- Default: according to the settings of CMDBuild instance
- Enabled
- Disabled



Bulk actions

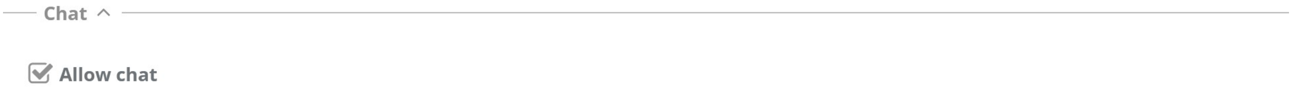
Default for cards bulk edit: Default

Default for cards bulk deletion: Default

Default for processes bulk abort: Default

20.4.5. Chat

It is possible for the current group to enable or disable the CMDBuild chat.

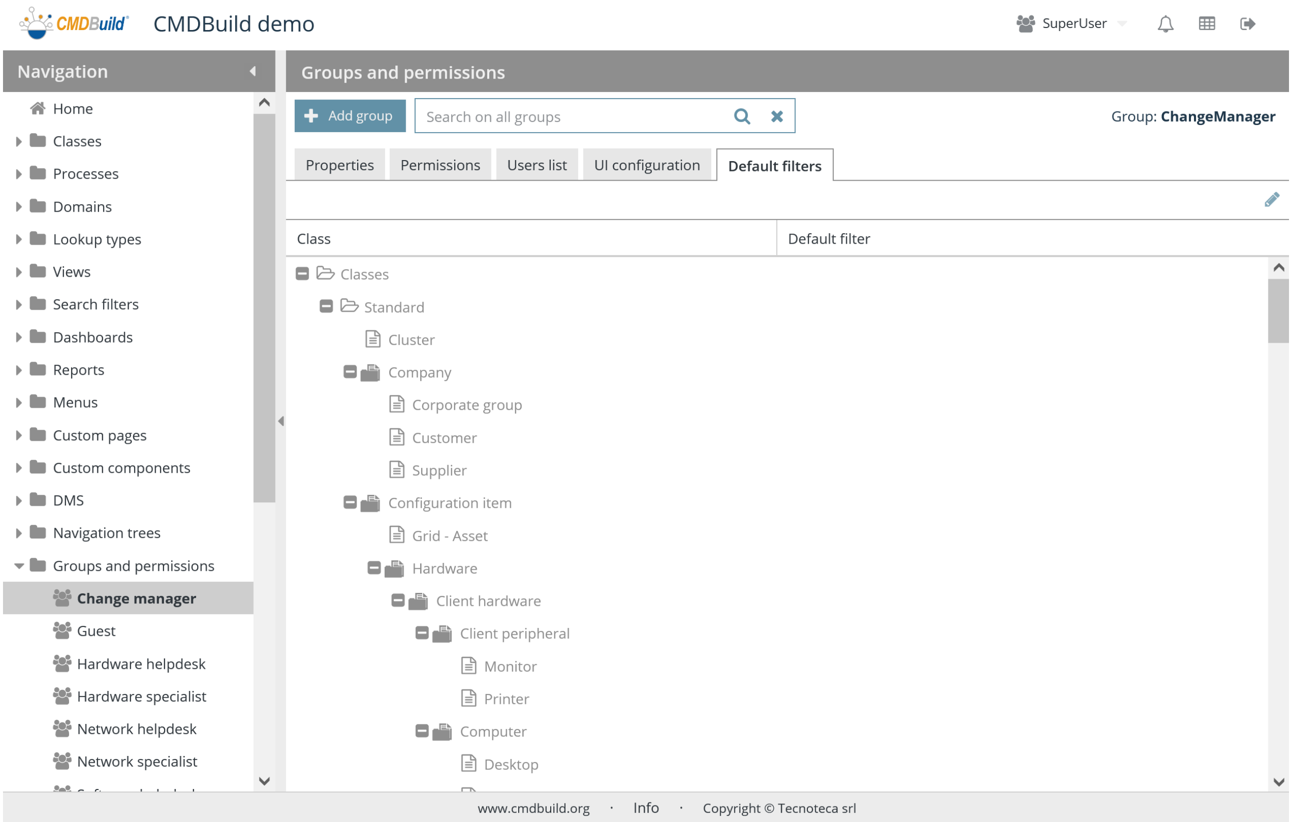


Chat

☑ Allow chat

20.5. TAB default filters

This function allows you to specify a possible filter applied into the Module for the data Management, when a user of the current group opens a class or process.

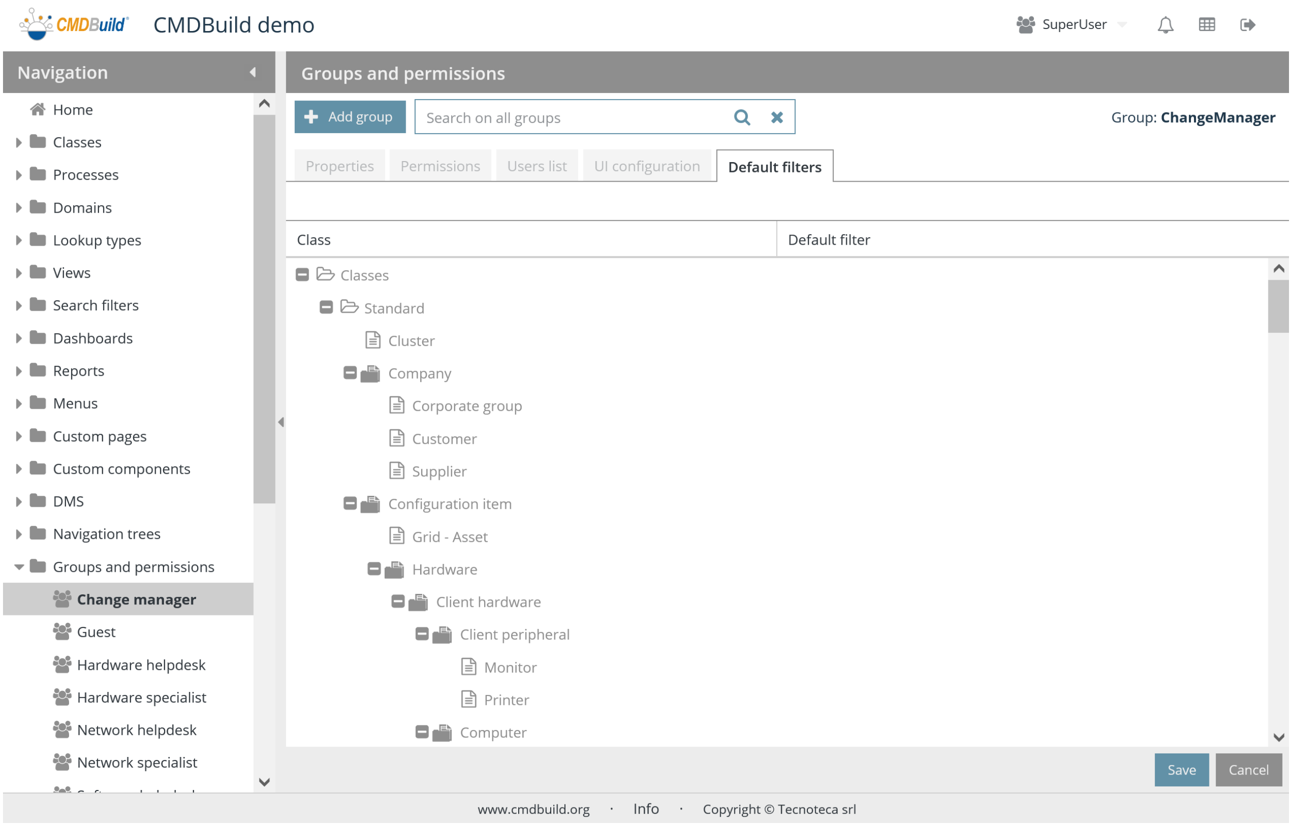


There is the possibility of performing the following operations:

• edit default filters for the current group	✎

The function that edit default filters allows you to:

- display the tree with the class and process hierarchy
- select, for each class / process, the default filter (among the ones configured with the specific function) required when a user from the current group access to that class / process in the data Management Module



Class	Default filter
Classes	
Standard	
Cluster	
Company	
Corporate group	
Customer	
Supplier	
Configuration item	
Grid - Asset	
Hardware	
Client hardware	
Client peripheral	
Monitor	
Printer	
Computer	

21. Users

As previously described, in CMDBuild the security management is based on the users' groups management (roles). Diverse permissions are associated to these groups on several item types and granular until the restriction on single rows and columns of a class.

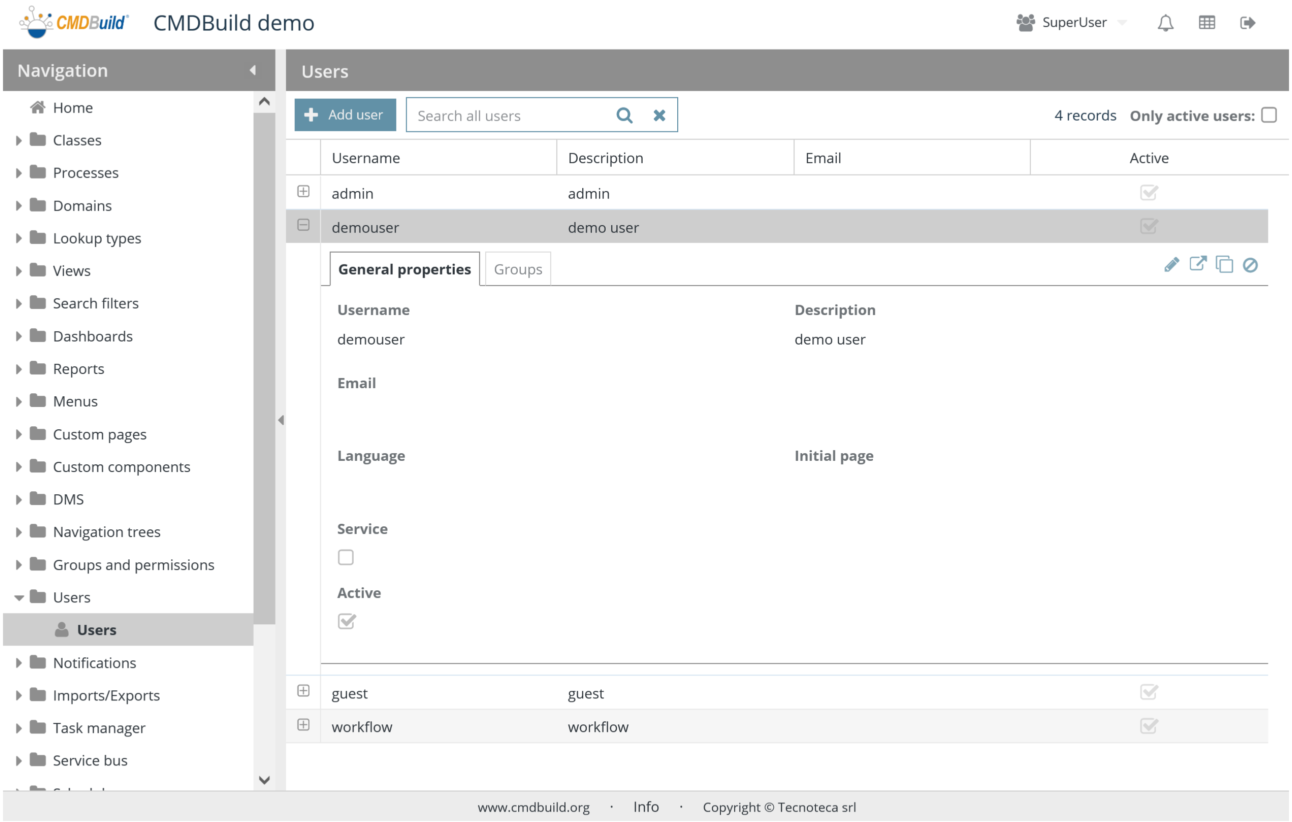
In case of multiple companies' use, or use among Departments or independent seats within a company, it is possible to configure CMDBuild in Multitenant mode, allowing each Tenant to work on an isolated CMDB subdivision.

CMDBuild allows you to manage single users that access the system associating them to one or more groups and one or more tenants.

Each user can access with the sum of permissions of groups and belonging tenants, or by selecting the desired group and tenant.

21.1. Properties

This feature allows you to create and edit system accounts.



Username	Description	Email	Active
admin	admin		✓
demouser	demo user		✓
guest	guest		✓
workflow	workflow		✓

Username
demouser

Description
demo user

Email

Language

Initial page

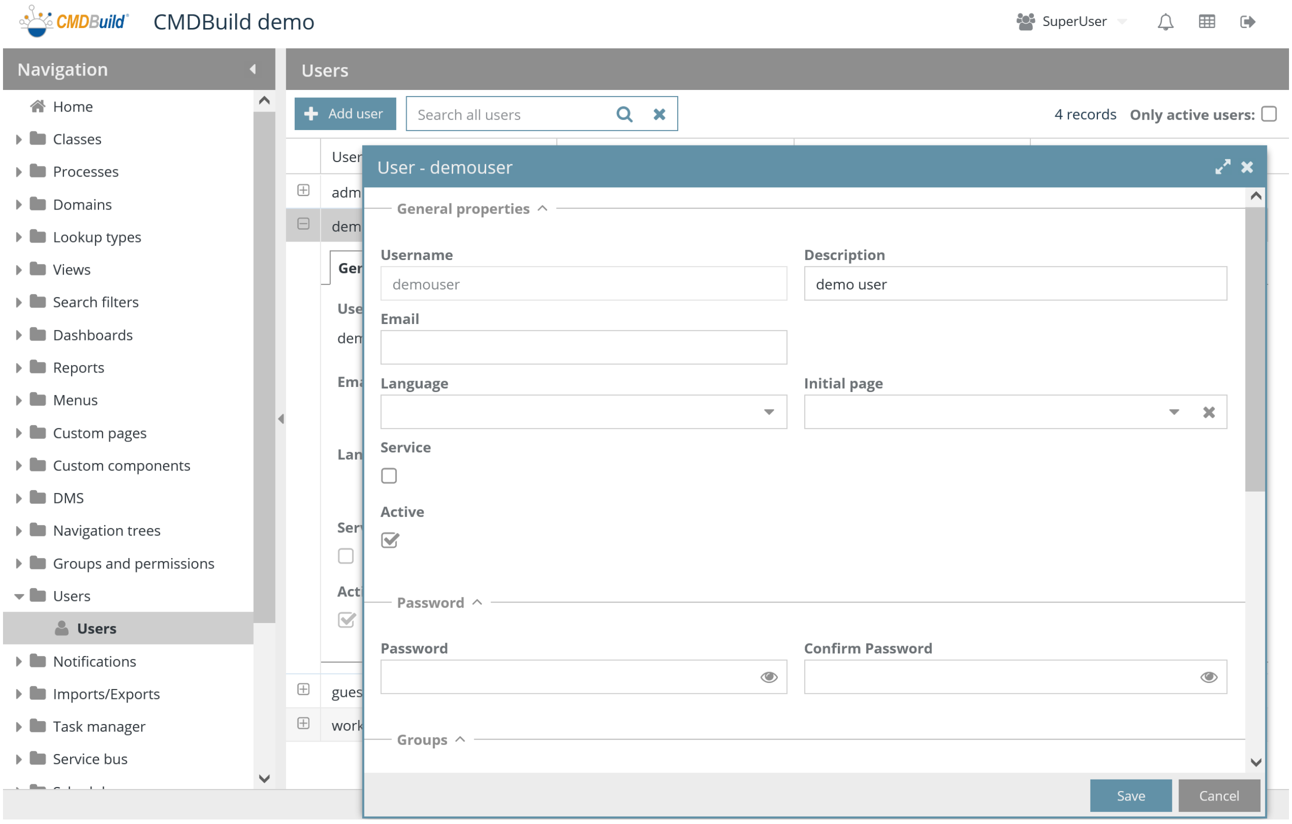
Service
□

Active
☑

There is the possibility of performing the following operations:

- add a new user	✚ Add user
- do searches on the configured users	Search all users 🔍 ✖
- display only active users	Only active users: □
- edit the selected user	✎
- display the complete information related to the selected user	↗
- clone the selected user	⧉
- disable the selected user (i.e. set the status as non-active)	⊘

For each user you are required to compile various information divided into the three following sections.



User - demouser

General properties

Username: demouser

Description: demo user

Email:

Language:

Initial page:

Service: □

Active: ☑

Password

Password:

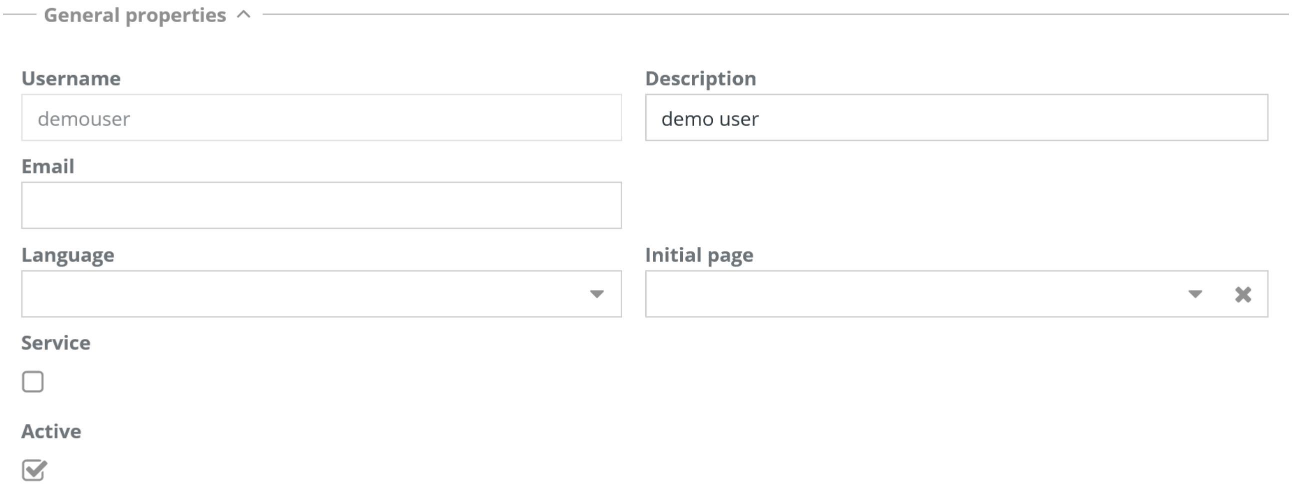
Confirm Password:

Groups

21.1.1. General properties

The following information is required:

- Username: login username
- Description: information used in the application header to recall the current user's name
- Email: user's email address, used to file notifications
- Language: default language to access the application
- Starting page: opening page when the current user access the application
- Service: it shows that it is a technical user used only by an automatic service
- Active: the user is active and can be used



General properties

Username: demouser

Description: demo user

Email:

Language:

Initial page:

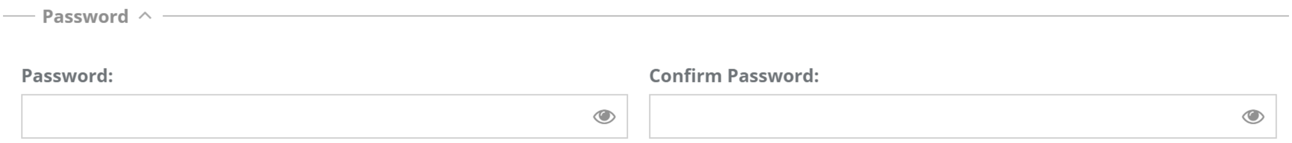
Service: □

Active: ☑

21.1.2. Password

The following information is required:

- Password: password to access the system (the inserted string is hidden when typing for security and privacy purposes)
- Confirm password



Password

Password:

Confirm Password:

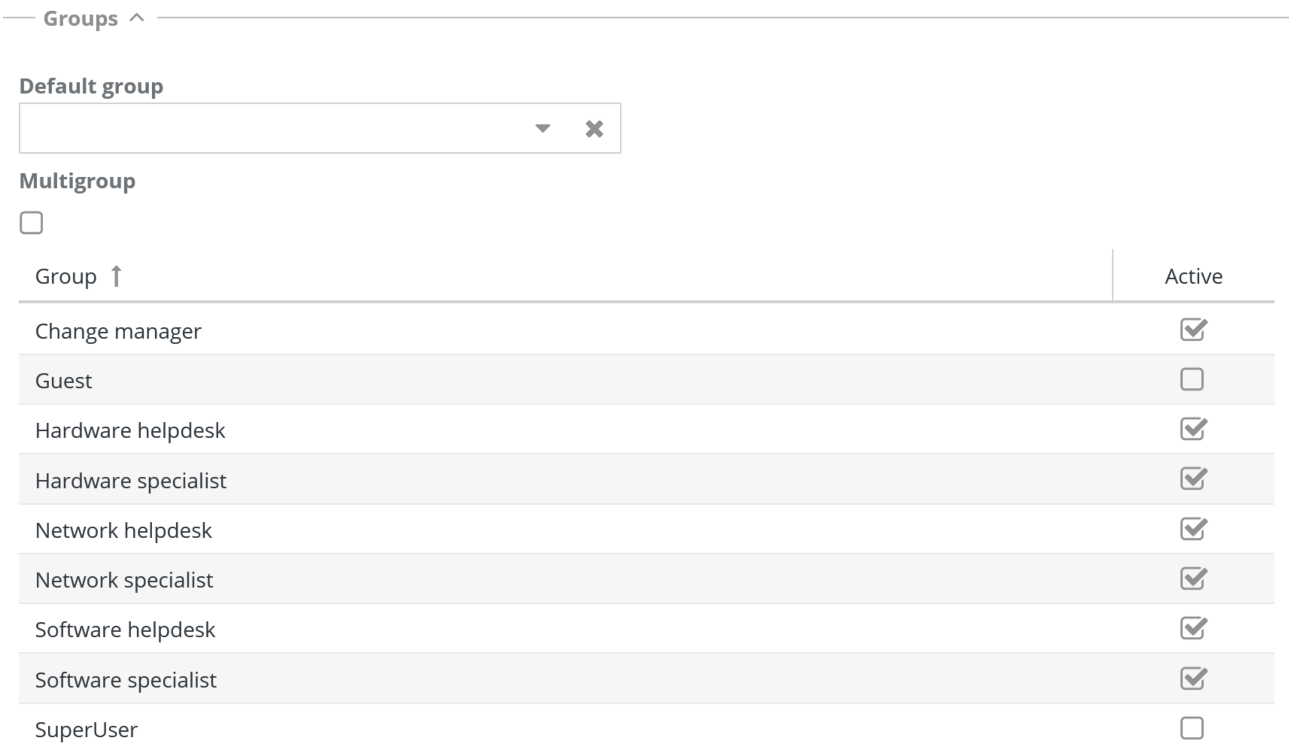
21.1.3. Belonging groups

The following information is required:

- Default group: if compiled it is the group used for the login in case the multigroup option has not been chosen. In case the multigroup option is active, it is the group whose settings are used in the expected cases (process start, etc.)
- Multigroup: the user accesses the application with the sum of permissions of all groups

he/she belongs to (see next point), even in case the user is not asked for an access group when logging-in

- List of groups: groups belonging to the user can be selected. When logging-in they are suggested in case neither the default group nor the multitenant option are set



Groups

Default group

Multigroup

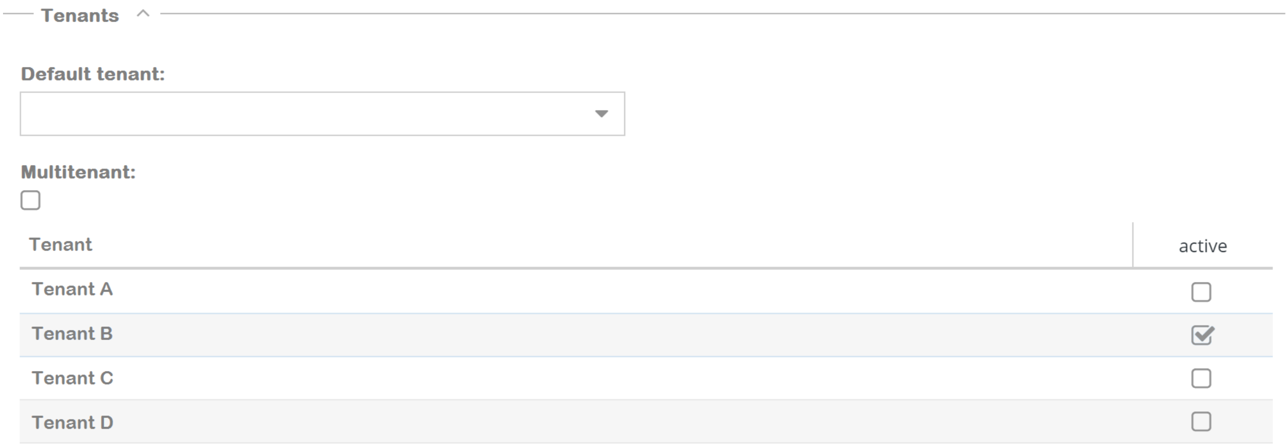
□

Group	Active
Change manager	☑
Guest	□
Hardware helpdesk	☑
Hardware specialist	☑
Network helpdesk	☑
Network specialist	☑
Software helpdesk	☑
Software specialist	☑
SuperUser	□

21.1.4. Belonging tenant

The following information is required:

- Default tenant: if compiled, it is the tenant used for the login in case the multitenant option has not been chosen
- Multitenant: the user accesses the application with the visibility rules of all tenants he/she belongs to (see next point). Also in this case when logging-in the access tenant is not required
- List of tenants: tenants belonging to the user can be selected. When logging-in they are suggested in case neither the default tenants nor the multitenant option are set



Tenants

Default tenant:

Multitenant: □

Tenant	active
Tenant A	□
Tenant B	☑
Tenant C	□
Tenant D	□

22. Notifications

In CMDBuild there are notifications based on emails and messages starting from the application (in-app notifications) displayed through popup.

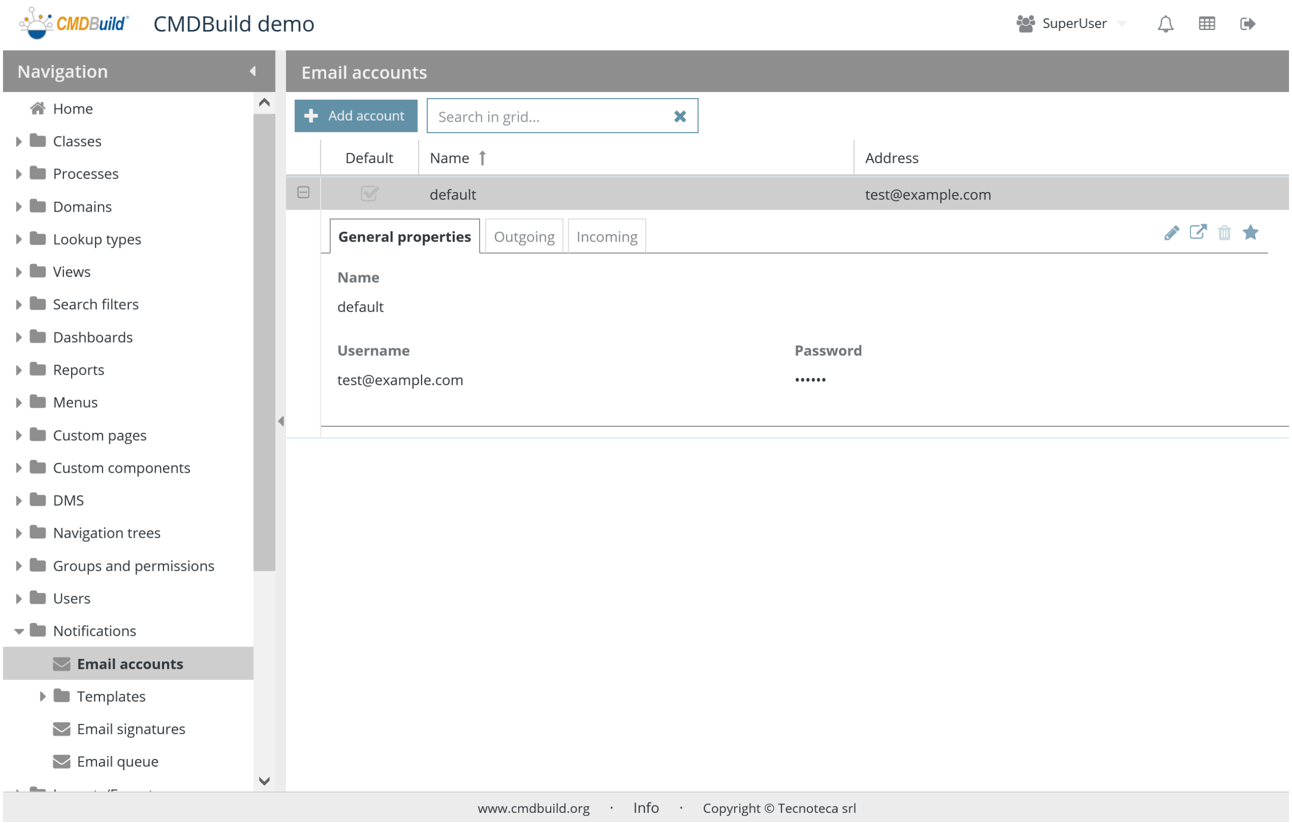
Notifications are controlled from the core for a better control and a simpler updating of sending parameters when they need to be edited.

In the Administration Module you can configure accounts for email outgoing (SNMP) and incoming (IMAP) and also templates used to send notifications to operators (workflow system, widget, Task manager, etc.), as email and in-app.

In particular the centralized definition of templates used to send notifications in the processes simplifies its maintenance, so that the XPDL descriptor has not to be edited.

22.1. Account

The function allows you to create new accounts to send and receive emails and edit the old ones.



Default	Name	Address
✓	default	test@example.com

Name
default

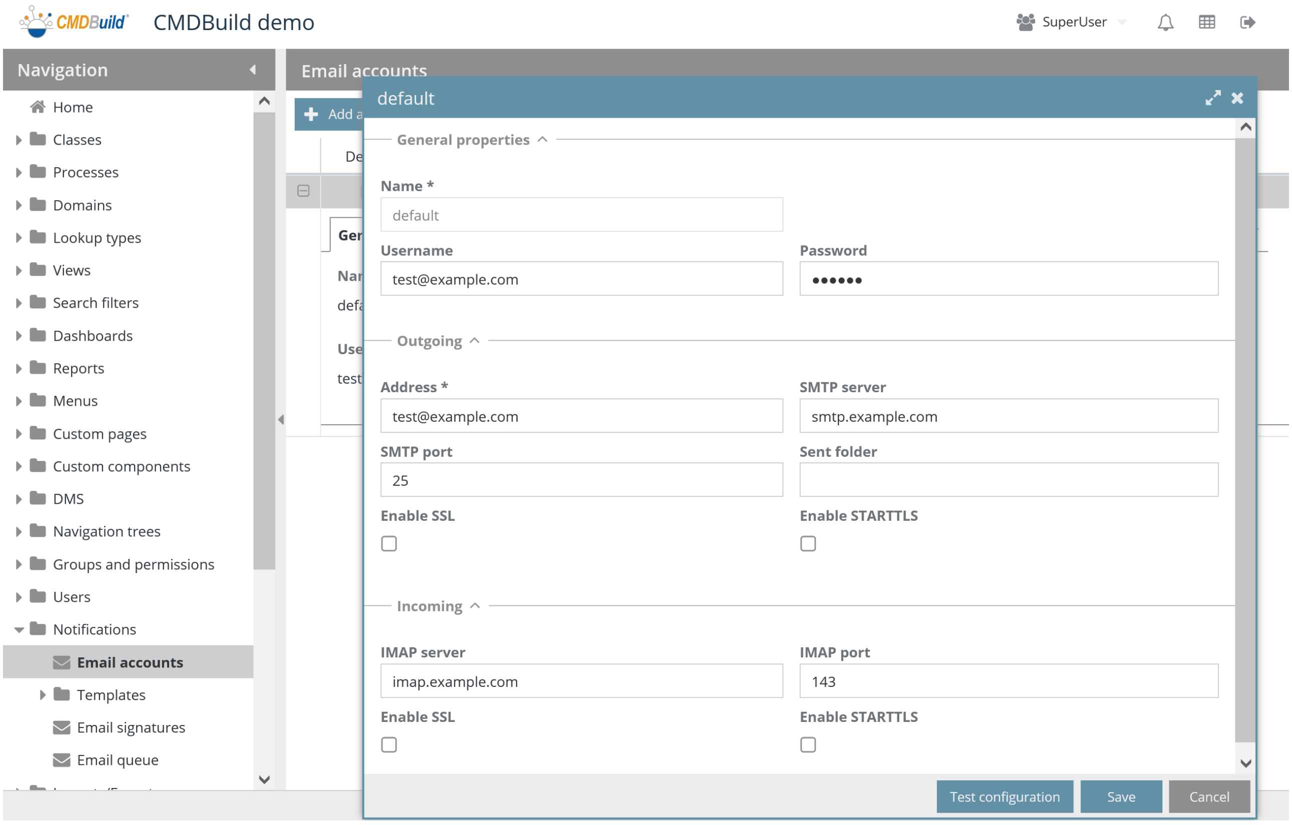
Username
test@example.com

Password
••••••

There is the possibility of performing the following operations:

• configure a new account	Add account
• do searches on the configured accounts	Search in grid... ✖

- edit the selected account	✎
- display the complete information related to the selected account	↗
- delete the selected account	🗑
- set the current account as default account	★
- execute a test for the correct configuration of the account	Test configuration
- save inserted configurations	Save
- exit without saving new configurations	Cancel



General properties

Name *: default

Username: test@example.com

Password: ••••••

Outgoing

Address *: test@example.com

SMTP server: smtp.example.com

SMTP port: 25

Sent folder:

Enable SSL: □

Enable STARTTLS: □

Incoming

IMAP server: imap.example.com

IMAP port: 143

Enable SSL: □

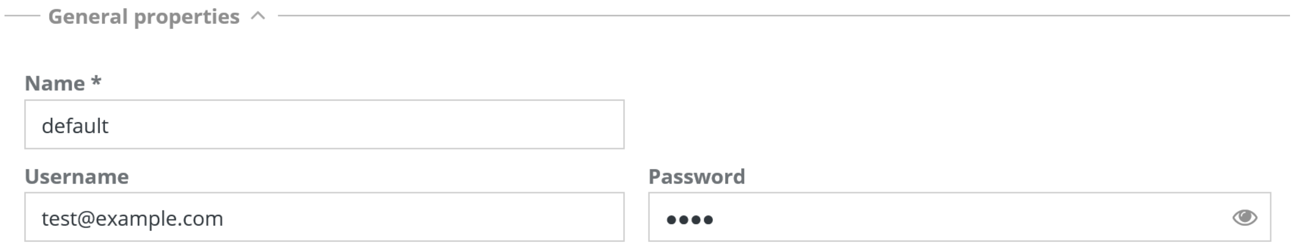
Enable STARTTLS: □

Each account requires several parameters to be compiled, which are divided into the sections described below.

22.1.1. General properties

The following information is required:

- Name: account name
- Username: used to access the email server
- Password: used to access the email server



General properties

Name *: default

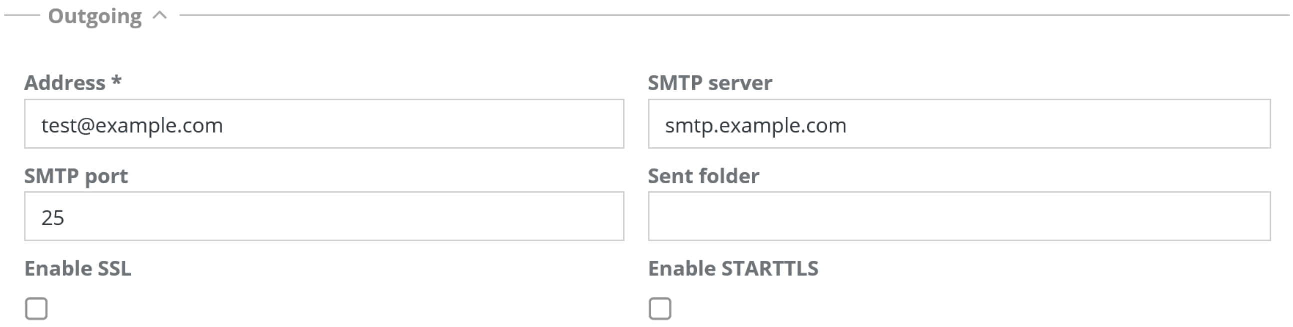
Username: test@example.com

Password: ●●●●

22.1.2. Outgoing mail parameters

The following information is required:

- E-mail address: e-mail account
- SMTP server: outgoing server
- SMTP port: server port
- Sent folder: folder where sent e-mails are filed
- Enable SSL
- Enable STARTTLS



Outgoing

Address *: test@example.com

SMTP server: smtp.example.com

SMTP port: 25

Sent folder:

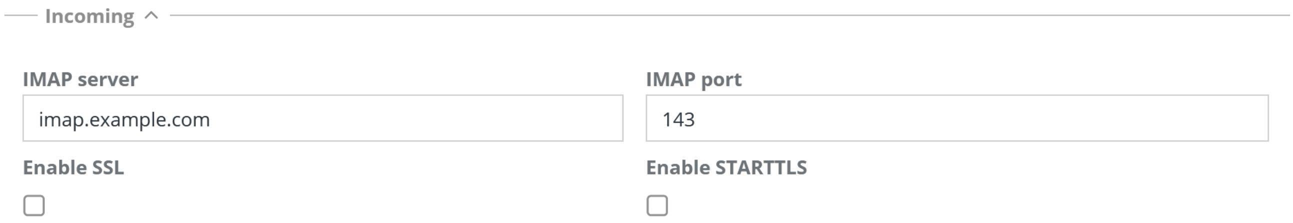
Enable SSL: □

Enable STARTTLS: □

22.1.3. Incoming mail parameters

The following information is required:

- IMAP server: incoming server
- IMAP port: server port
- Enable SSL
- Enable STARTTLS



Incoming

IMAP server: imap.example.com

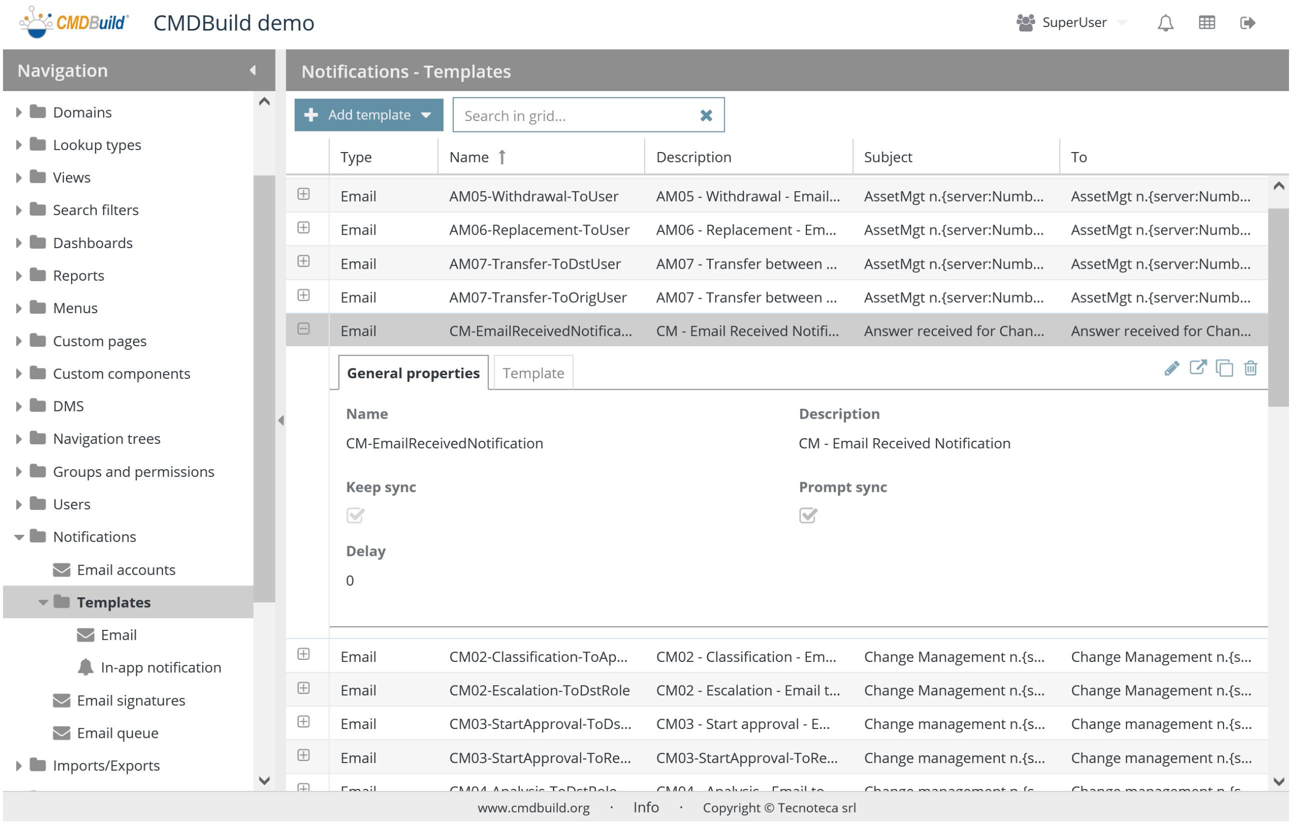
IMAP port: 143

Enable SSL: □

Enable STARTTLS: □

22.2. E-mail template

This function allows you to create or edit email notification templates that will be used for enabled features (workflow system, widget, task manager, etc.).



Type	Name	Description	Subject	To
Email	AM05-Withdrawal-ToUser	AM05 - Withdrawal - Email...	AssetMgt n.{server:Numb...	AssetMgt n.{server:Numb...
Email	AM06-Replacement-ToUser	AM06 - Replacement - Em...	AssetMgt n.{server:Numb...	AssetMgt n.{server:Numb...
Email	AM07-Transfer-ToDstUser	AM07 - Transfer between ...	AssetMgt n.{server:Numb...	AssetMgt n.{server:Numb...
Email	AM07-Transfer-ToOrigUser	AM07 - Transfer between ...	AssetMgt n.{server:Numb...	AssetMgt n.{server:Numb...
Email	CM-EmailReceivedNotifica...	CM - Email Received Notifi...	Answer received for Chan...	Answer received for Chan...

General properties | Template

Name: CM-EmailReceivedNotification

Description: CM - Email Received Notification

Keep sync: ☑

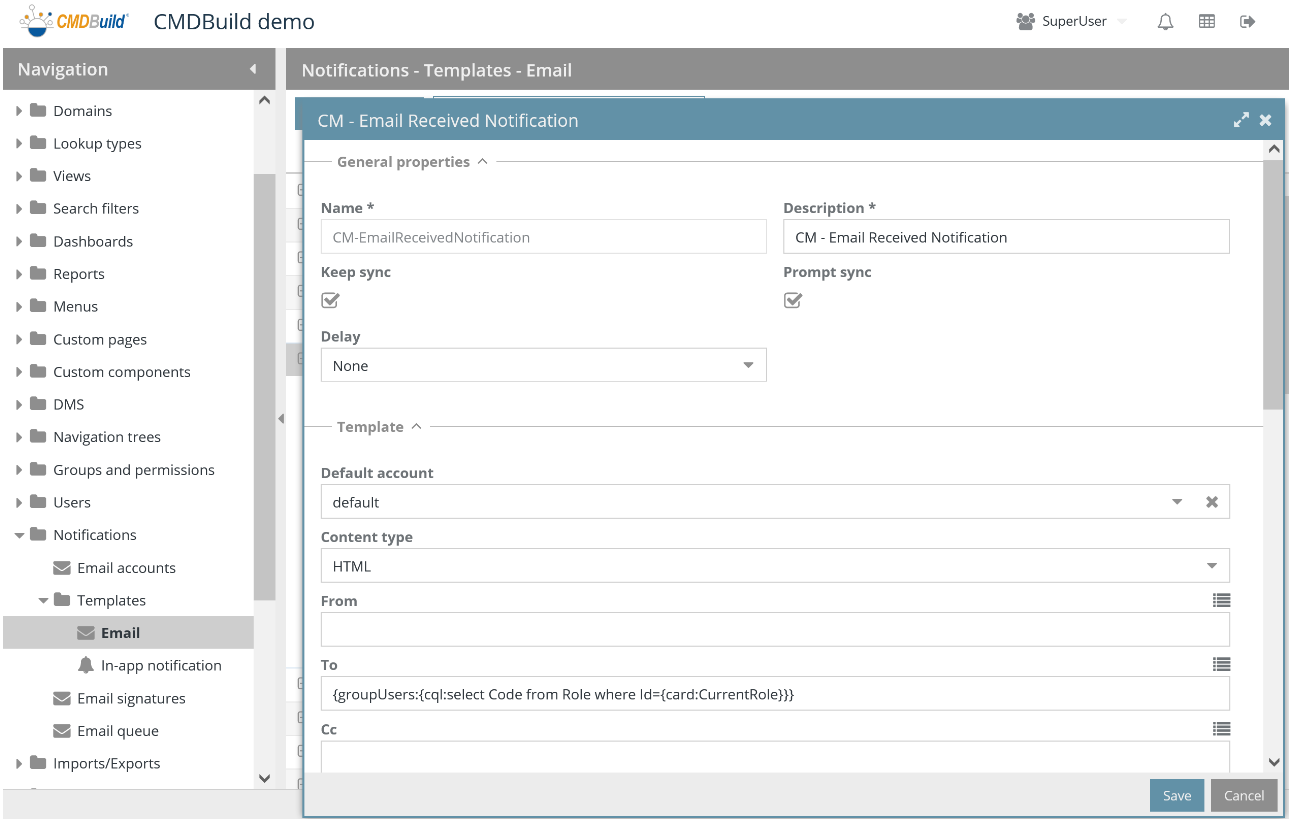
Prompt sync: ☑

Delay: 0

Type	Name	Description	Subject	To
Email	CM02-Classification-ToAp...	CM02 - Classification - Em...	Change Management n.{s...	Change Management n.{s...
Email	CM02-Escalation-ToDstRole	CM02 - Escalation - Email t...	Change Management n.{s...	Change Management n.{s...
Email	CM03-StartApproval-ToDs...	CM03 - Start approval - E...	Change management n.{s...	Change management n.{s...
Email	CM03-StartApproval-ToRe...	CM03-StartApproval-ToRe...	Change management n.{s...	Change management n.{s...
Email	CM04-Analysis-ToDstRole	CM04 - Analysis - Email to...	Change management n.{s...	Change management n.{s...

There is the possibility of performing the following operations:

• create a new template	✚ Add template
• do searches on configured templates	Search in grid... ✖
• edit the selected template	✎
• display the complete information related to the selected template	↗
• clone the selected template	⧉
• delete the selected template	🗑



CM - Email Received Notification

General properties

Name *: CM-EmailReceivedNotification

Description *: CM - Email Received Notification

Keep sync: ☑

Prompt sync: ☑

Delay: None

Template

Default account: default

Content type: HTML

From:

To: {groupUsers:{cql:select Code from Role where Id={card:CurrentRole}}}

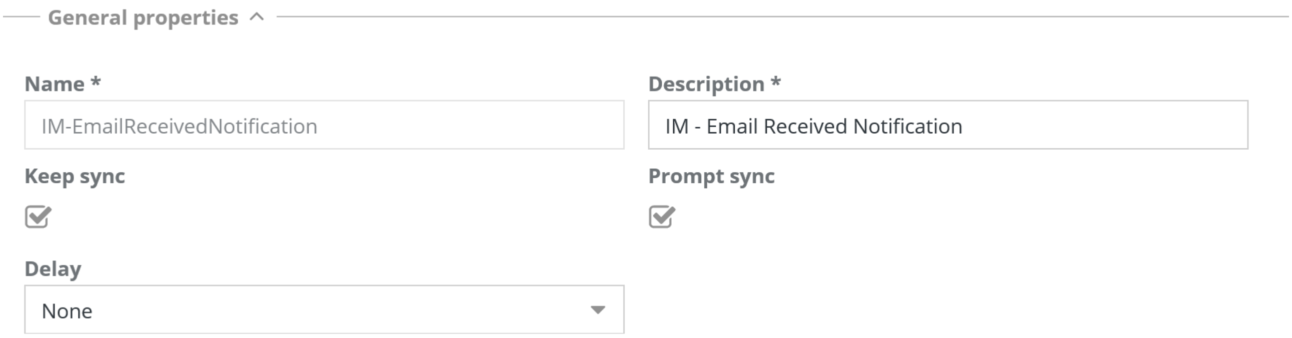
Cc:

Each template requires several parameters to be compiled, which are divided into the sections described below.

22.2.1. General properties

The following information is required:

- Name: template name
- Description: template description
- Keep sync: it shows you that the text of the e-mail is automatically updated when attributes used like email parameters vary (default = yes)
- Confirm sync: it shows you that you have to confirm every time the e-mail sync edits the text when attributes used like parameters vary
- Delay: it requires that the e-mail is sent later (after 1-2-4 hours, 1-2-4 days, 1-2 weeks, 1 month)



General properties

Name *: IM-EmailReceivedNotification

Description *: IM - Email Received Notification

Keep sync: ☑

Prompt sync: ☑

Delay: None

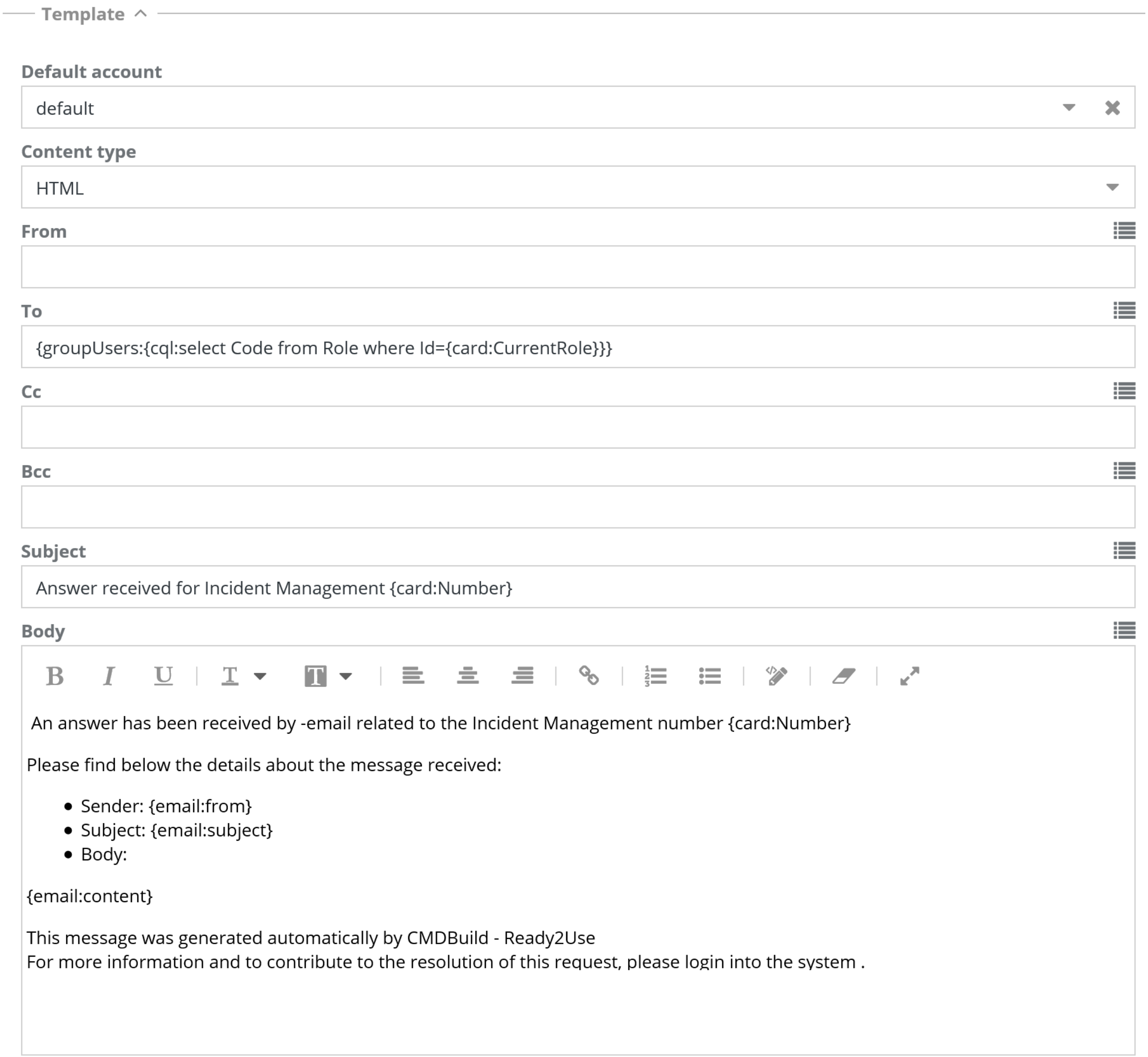
22.2.2. Template

The following information is required:

- Default account
- Type of content: HTML or TEXT
- From: sender's address
- To: addressee's address (or list of several addresses separated by comma)
- Cc: copy carbon address (or list of several addresses separated by comma)
- Bcc: blind copy carbon address (or list of several addresses separated by comma)
- Subject: notification subject
- Body: notification body

Through CQL expressions you can customize your notification in each template field with information available in the current context (i.e. workflow variables used to send emails).

CQL expressions can include metadata you can define and consult through the icon "Show metadata" on the right above the field.



Template

Default account

default

Content type

HTML

From

To

{groupUsers:{cql:select Code from Role where Id={card:CurrentRole}}}

Cc

Bcc

Subject

Answer received for Incident Management {card:Number}

Body

An answer has been received by -email related to the Incident Management number {card:Number}

Please find below the details about the message received:

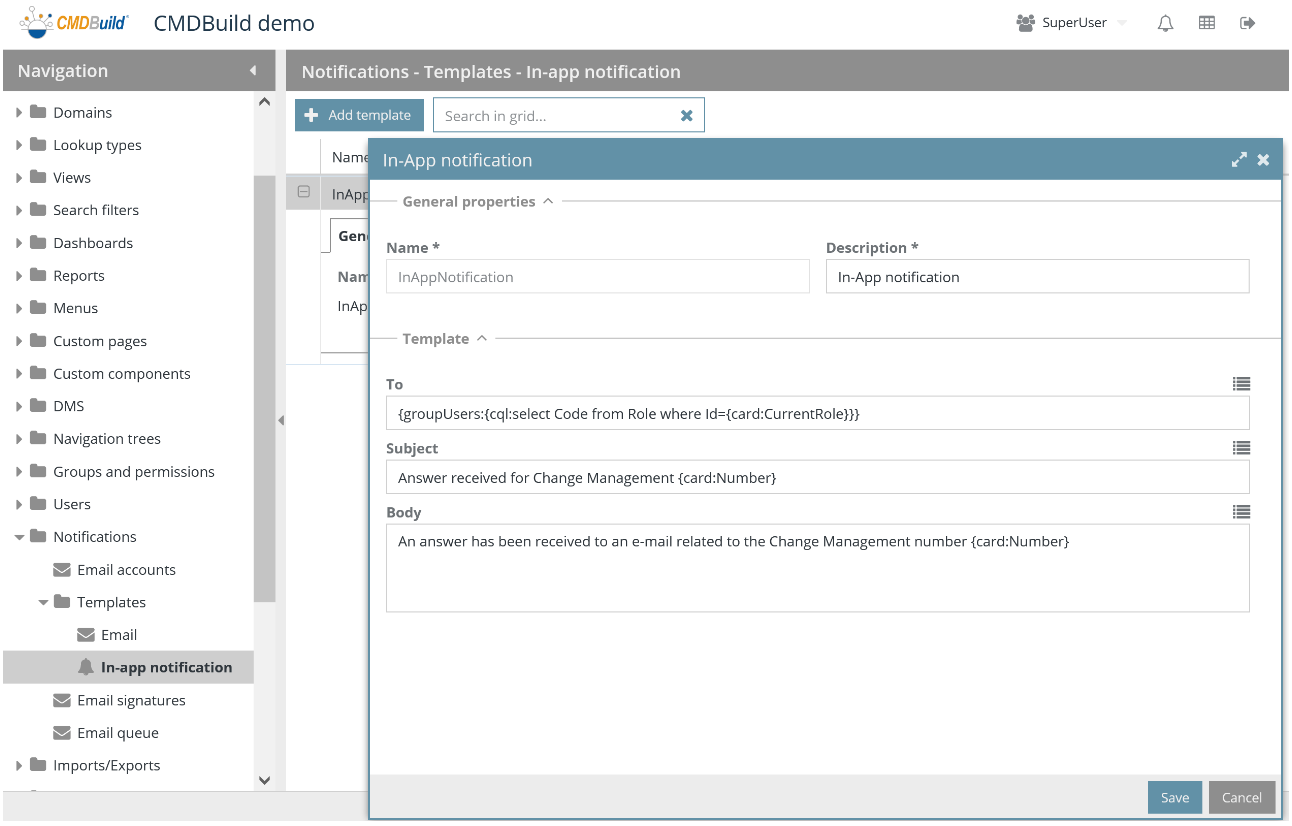
- Sender: {email:from}
- Subject: {email:subject}
- Body:

{email:content}

This message was generated automatically by CMDBuild - Ready2Use
For more information and to contribute to the resolution of this request, please login into the system .

22.3. “In-app” notification template

This function allows you to create or edit in-app notification templates that will be used for enabled features (workflow system, widget, task manager, etc.).



In-App notification

General properties

Name * InAppNotification

Description * In-App notification

Template

To: {groupUsers:{cql:select Code from Role where Id={card:CurrentRole}}}

Subject: Answer received for Change Management {card:Number}

Body: An answer has been received to an e-mail related to the Change Management number {card:Number}

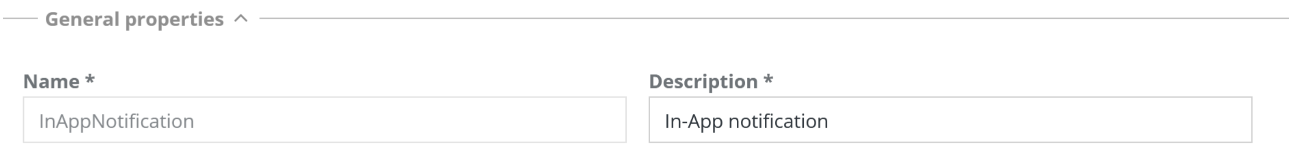
Management features are the same of email notification ones.

Each template requires several parameters to be compiled, which are divided into the sections described below.

22.3.1. General properties

The following information is required:

- Name: template name
- Description: template description



General properties

Name * InAppNotification

Description * In-App notification

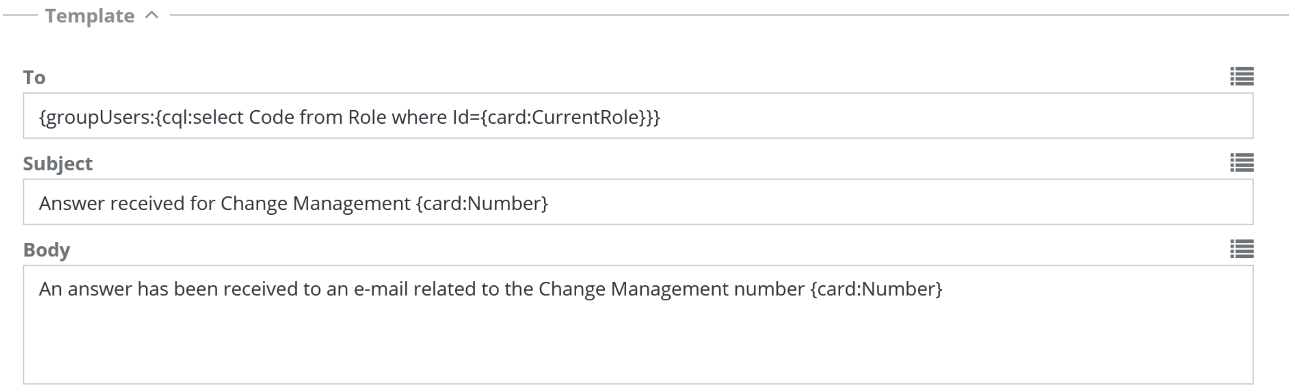
22.3.2. Template

The following information is required:

- To: addressee's address (or list of several addresses separated by comma)
- Subject: notification subject
- Body: notification body

Through CQL expressions you can customize your notification in each template field with information available in the current context (i.e. workflow variables used to send emails).

CQL expressions can include metadata you can define and consult through the icon "Show metadata" on the right above the field.



Template ^

To

{groupUsers:{cql:select Code from Role where Id={card:CurrentRole}}}

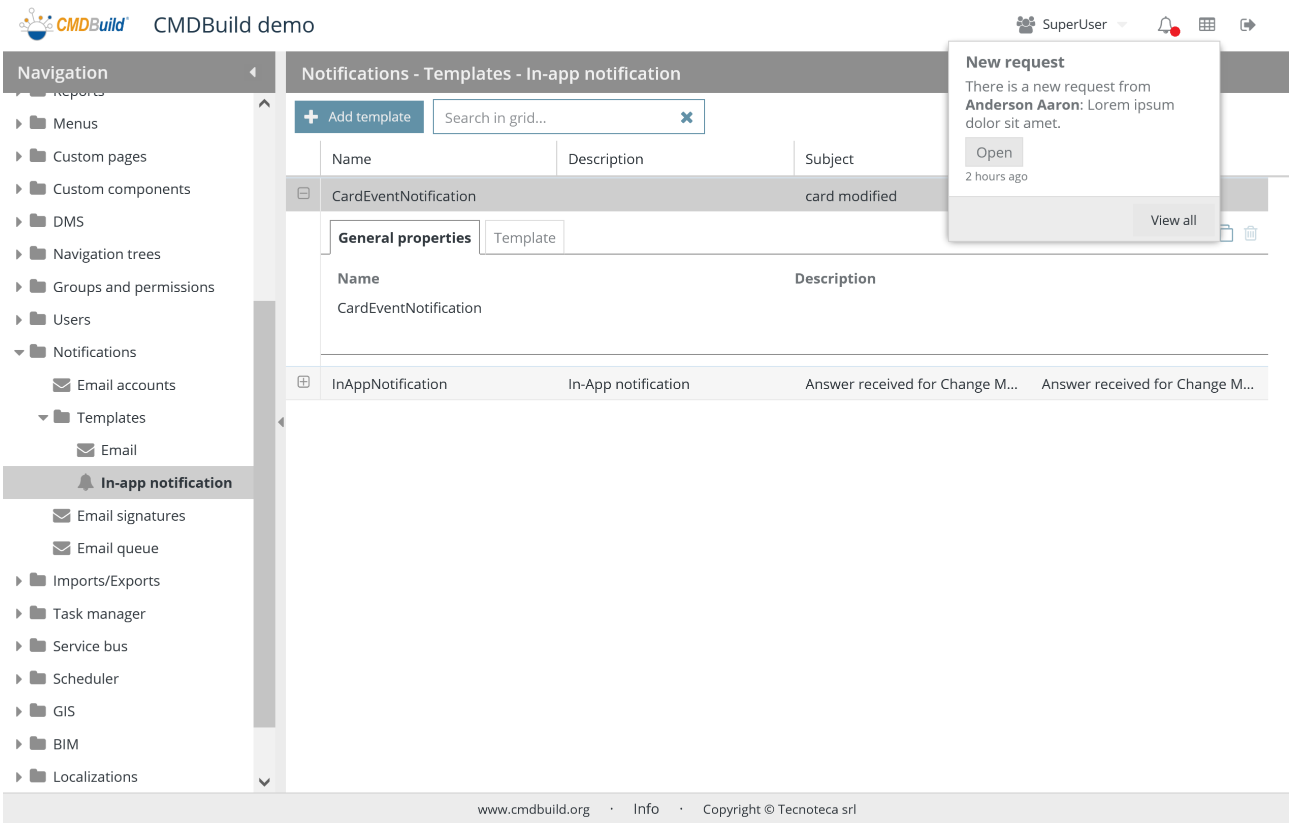
Subject

Answer received for Change Management {card:Number}

Body

An answer has been received to an e-mail related to the Change Management number {card:Number}

This is an example of in-app notification received in the Administration Module.



Notifications - Templates - In-app notification

Name	Description	Subject	
CardEventNotification		card modified	
InAppNotification	In-App notification	Answer received for Change M...	Answer received for Change M...

Name: CardEventNotification

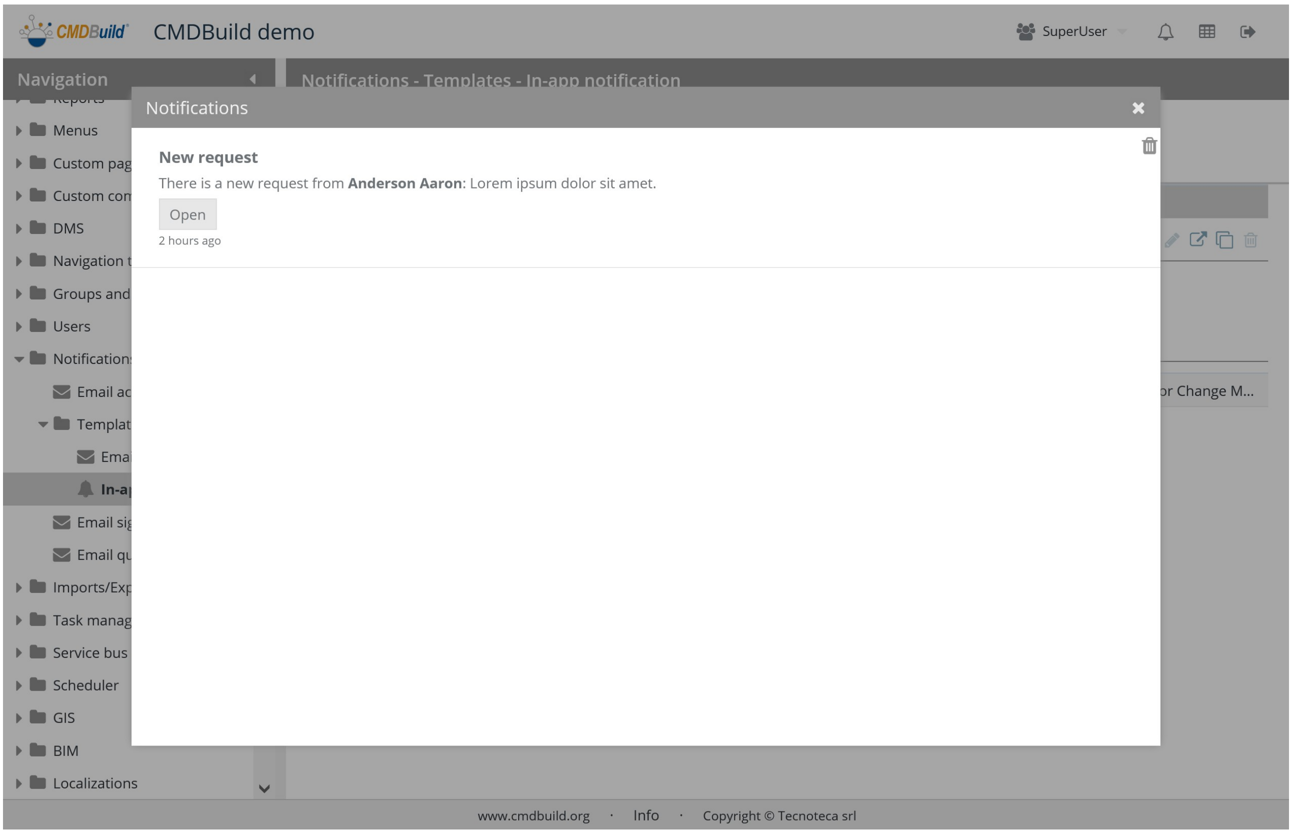
Description:

New request

There is a new request from **Anderson Aaron**: Lorem ipsum dolor sit amet.

Open

2 hours ago



Notifications

New request

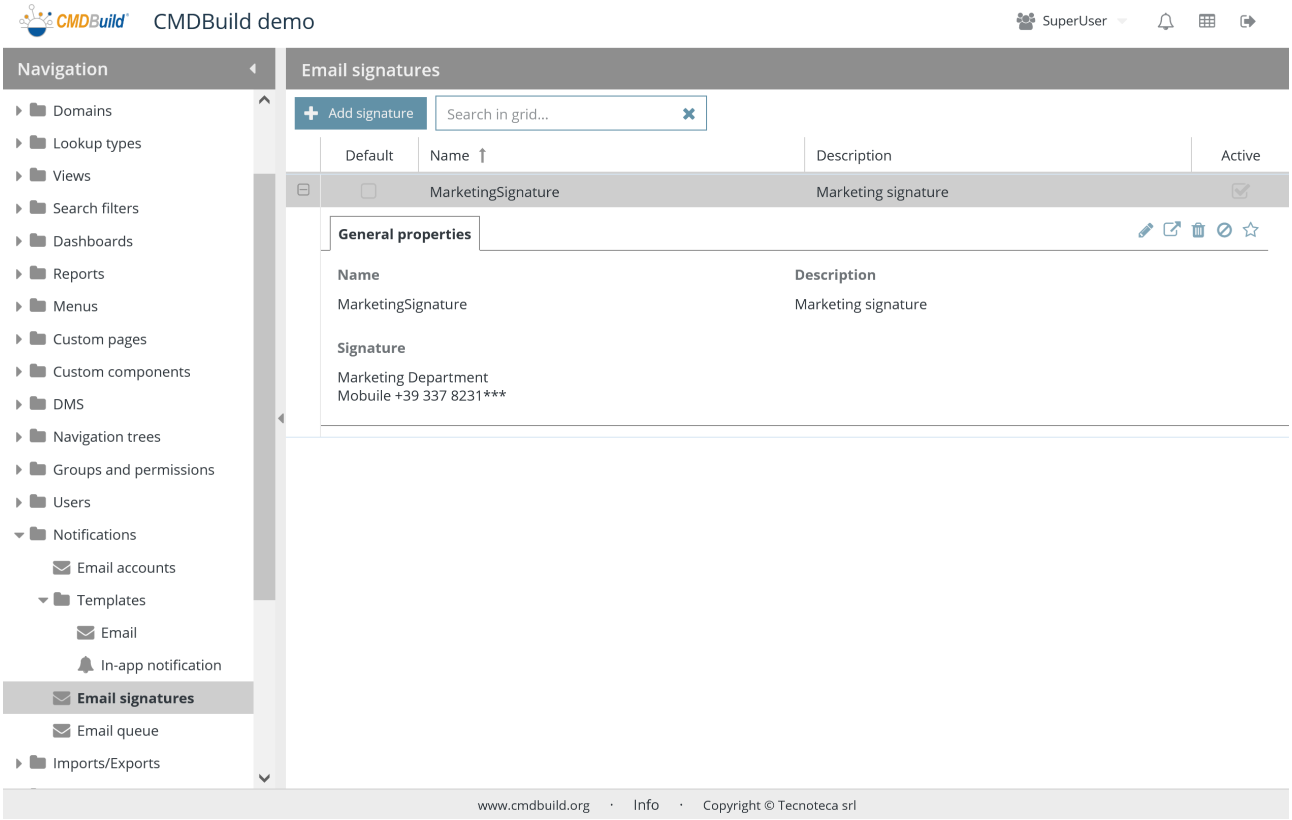
There is a new request from **Anderson Aaron**: Lorem ipsum dolor sit amet.

Open

2 hours ago

22.4. Email signatures

You can configure your signature and insert it at the bottom of your email text.



Default	Name	Description	Active
□	MarketingSignature	Marketing signature	☑

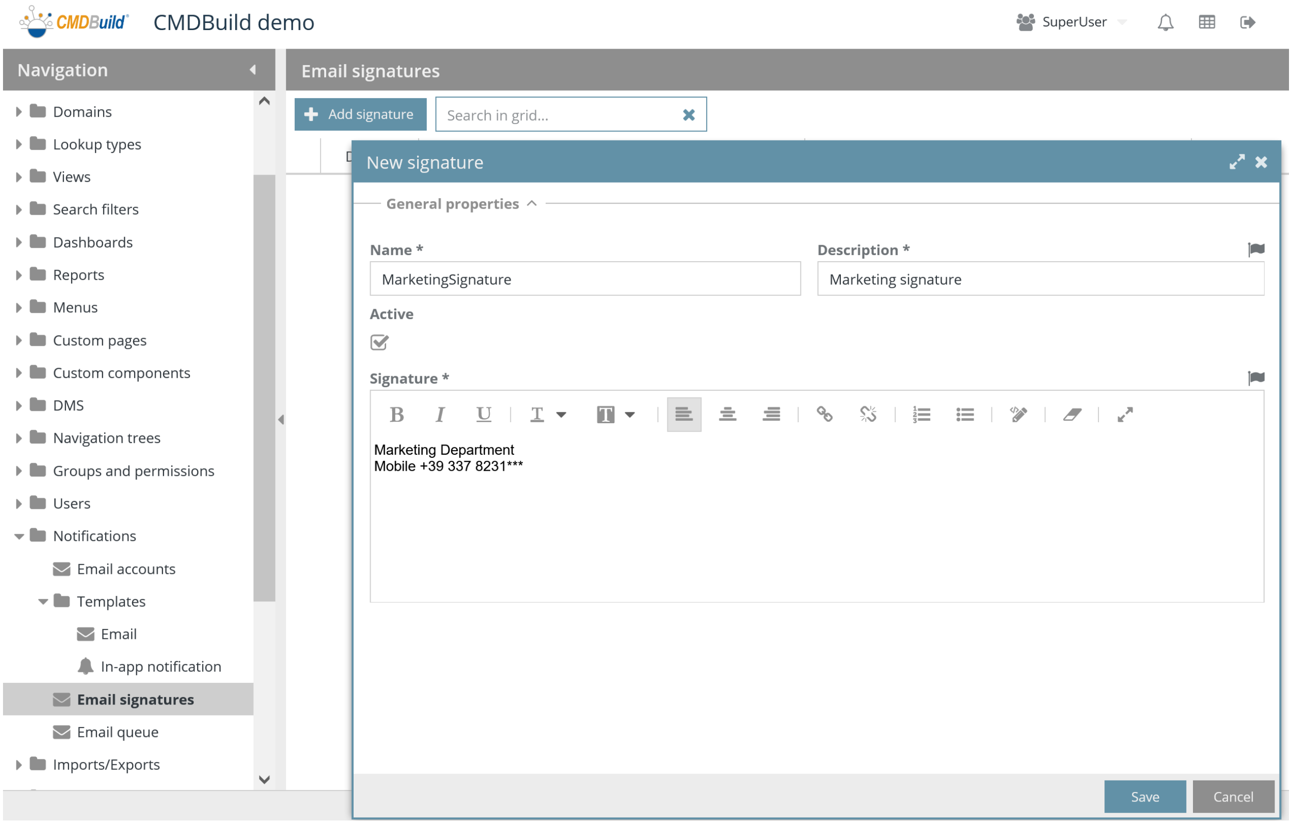
Name
MarketingSignature

Description
Marketing signature

Signature
Marketing Department
Mobuile +39 337 8231***

There is the possibility of performing the following operations:

- create a new signature	✚ Add signature
- do searched on the configured signatures	Search in grid... ✖
- edit the selected signature	✎
- display the complete information related to the selected signature	↗
- delete the selected signature	🗑
- disable the selected signature	⊘
- set the selected signature as default	☆



Name * MarketingSignature

Description * Marketing signature

Active ☑

Signature *

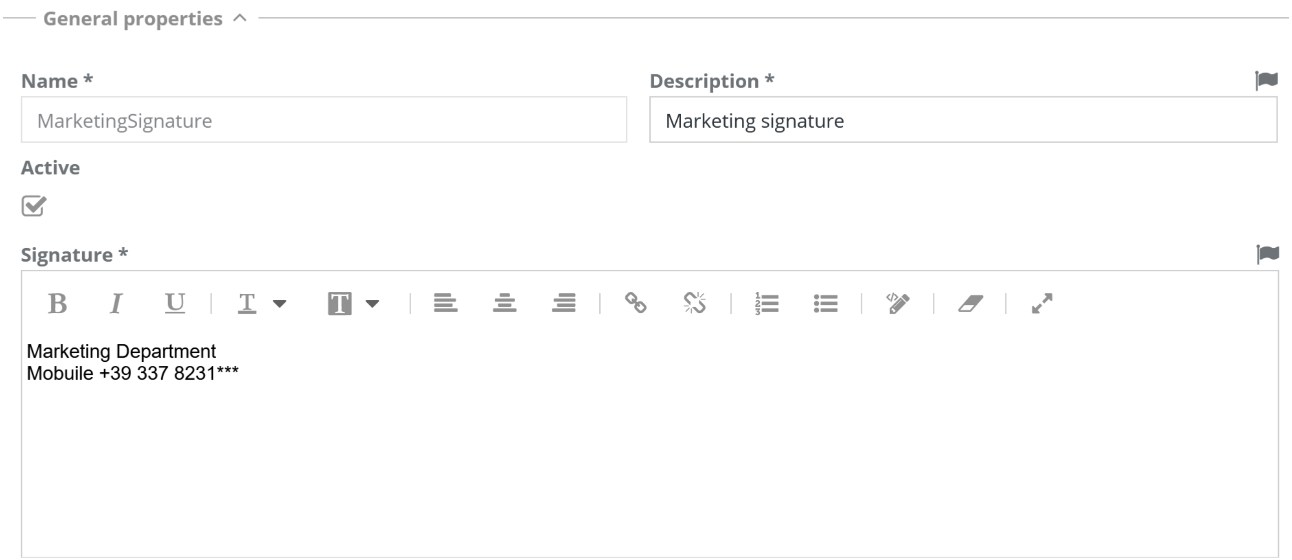
Marketing Department
Mobile +39 337 8231***

Each signature needs some parameters to be compiled.

22.4.1. General properties

The following information is required:

- Name: signature name
- Description: signature description
- Signature: signature text (HTML)



General properties

Name * MarketingSignature

Description * Marketing signature

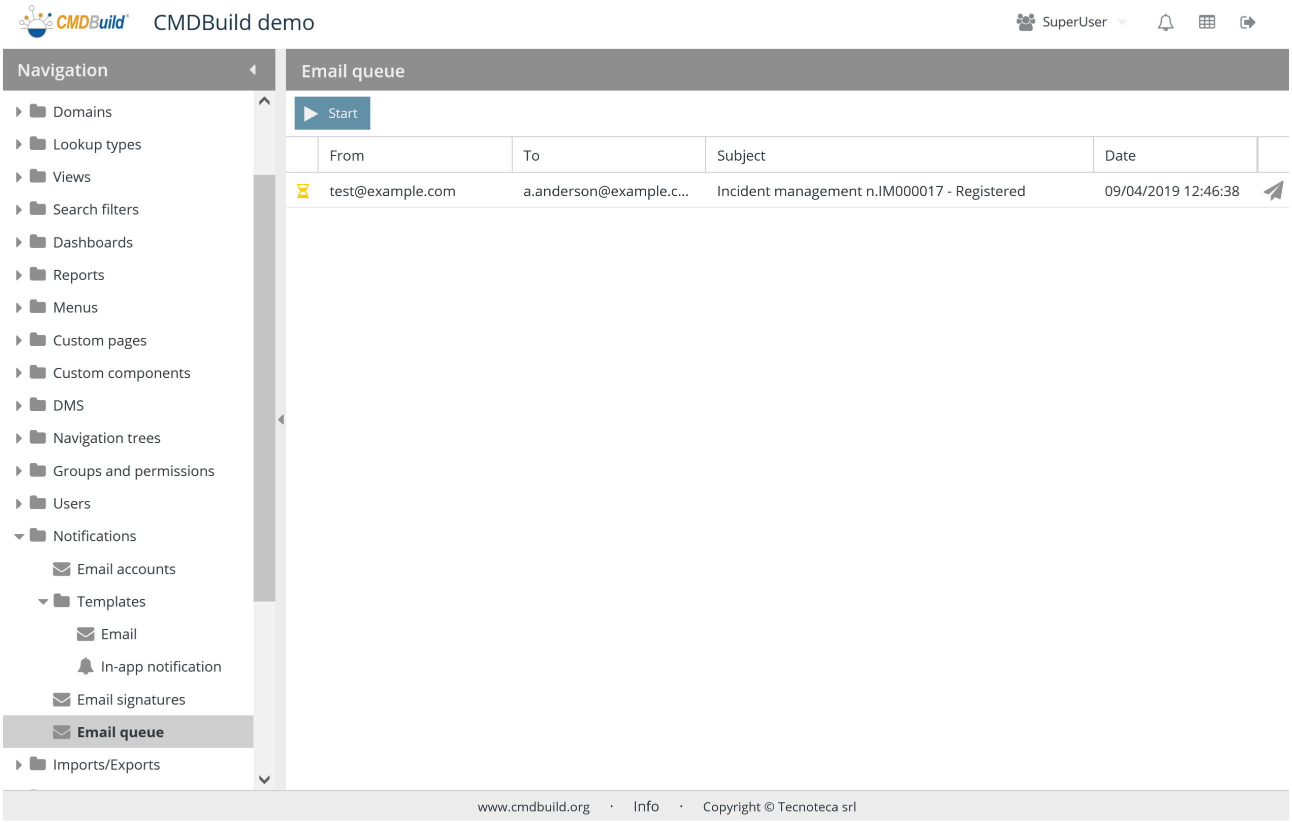
Active ☑

Signature *

Marketing Department
Mobuile +39 337 8231***

22.5. Email queue

The e-mail created by CMDBuild pass through a "queue", which can be checked through the features described in this paragraph.



From	To	Subject	Date
test@example.com	a.anderson@example.c...	Incident management n.IM000017 - Registered	09/04/2019 12:46:38

There is the possibility of performing the following operations:

- Start the email queue (please note whether this is a test and no emails have to be sent)	▶ Start
- Stop the email queue	❚❚ Stop
- Force an email in queue to be sent	➤

23. Import / Export

CMDBuild allows to configure templates including default parameters in order to execute massive data Import / Export operations, with many different formats, especially:

- template files for Import / Export through files in CSV, XLS or XLSX format
- database template, to import data from relational database charts
- GIS template, to import data from planimetries included in DWG files
- IFC template, to import IFC files produced by BIM 3D modelisation softwares

Such operations can be carried in an interactive mode from the Management Module or batch through the Task manager.

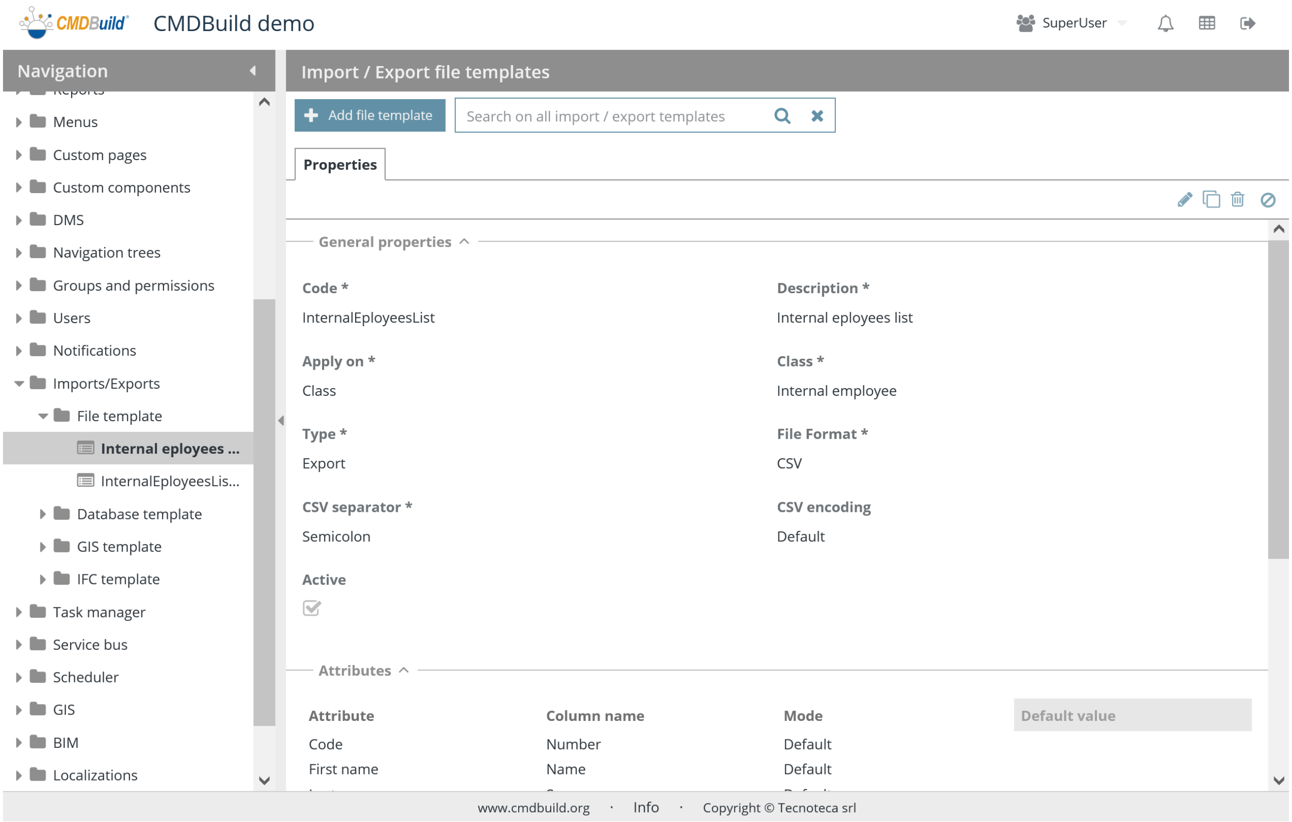
Templates gather all information necessary to automate the operation in case of batch execution and not to specify each time so many parameters in case of interactive execution.

23.1. Template file

It is possible to define both import and export templates through CSV, XLS, XLSX files.

In case of import template, the function can operate both in "merge" mode (updating the already available strings recognized through a unique key, inserting the new ones and managing the missing ones in a custom way) and in "add" mode.

23.1.1. Properties tab



Code *: InternalEployeesList

Description *: Internal eployees list

Apply on *: Class

Class *: Internal employee

Type *: Export

File Format *: CSV

CSV separator *: Semicolon

CSV encoding: Default

Active: ☑

Attribute	Column name	Mode	Default value
Code	Number	Default	
First name	Name	Default	

There is the possibility of performing the following operations:

• create a new template	✚ Add file template
• search the template in the list	Search... 🔍 ✖
• edit the selected template	✎
• clone the selected template	⧉
• delete the selected attribute	🗑
• disable the selected attribute (i.e. set the status as non-active)	⊘

Each value of the list requires some parameters to be compiled, as described below.

General properties

The following information is required:

- Code: template code
- Description: short template description
- Apply on: classes or domains
- Class / Process / Domain / View from join: it requires to select the type of element for the template
- Type: Import, Export, Import and Export (according to the type, some following parameters can be visible or hidden)
- File format: CSV, XLSX, XLS
- Active: it states whether the template is active and can be used

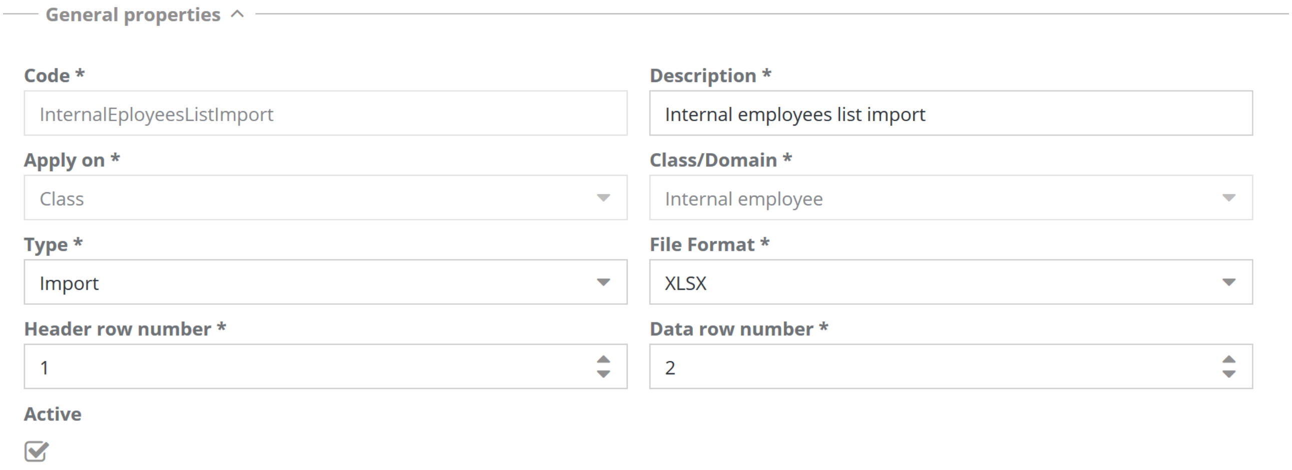
According to the format, some additional information is also required.

In case the format is CSV:

- CSV separator: "comma" or "semicolon" or "pipe (|)" or "TAB"
- CSV encoding

In case the format is XLSX or XLS:

- Header row number: position of the header row (column names)
- Data row number: position of the first row with data



General properties

Code *: InternalEployeesListImport

Description *: Internal employees list import

Apply on *: Class

Class/Domain *: Internal employee

Type *: Import

File Format *: XLSX

Header row number *: 1

Data row number *: 2

Active: ☑

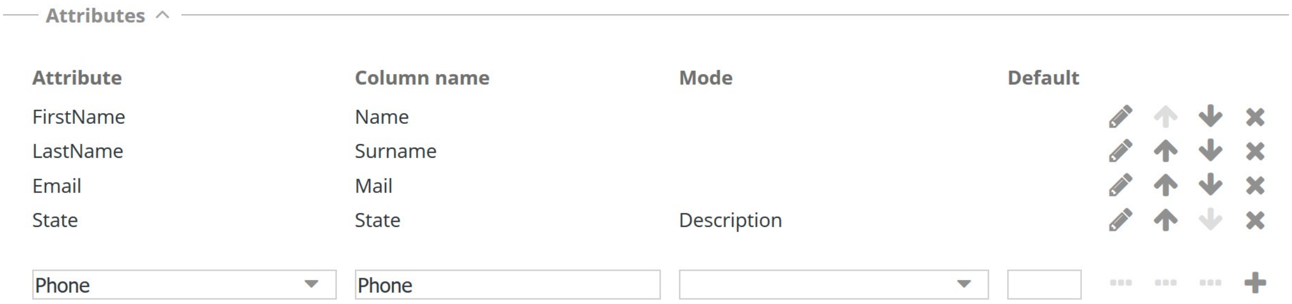
Attributes

The following information is required:

The following information is required for the setting of attribute mapping:

- Attribute: attribute name of the class in CMDBuild

- Column name: name of the column in the import or output file
- Mode: in case of Reference or Lookup CMDBuild attributes, it allows you to specify whether the date has to be understood as "Id" (unique identification in CMDBuild), as "Code" (standard attribute of all CMDBuild classes) or as "Description" (standard attribute of all CMDBuild classes)
- Default: in case of Import template, it allows you to specify a default value if missing in the file to charge



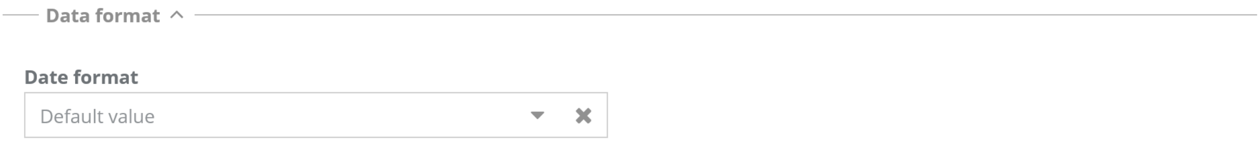
Attributes

Attribute	Column name	Mode	Default
FirstName	Name		
LastName	Surname		
Email	Mail		
State	State	Description	
Phone	Phone		

Dates format

The following information is required:

- Dates format: it is possible to choose one of the following formats (dd/mm/yyyy, dd-mm-yyyy, dd.mm.yyyy, mm/dd/yyyy, yyyy/mm/dd, yyyy-mm-dd)



Data format

Date format: Default value

Import criteria

The following information is required only for Import and Import / Export templates:

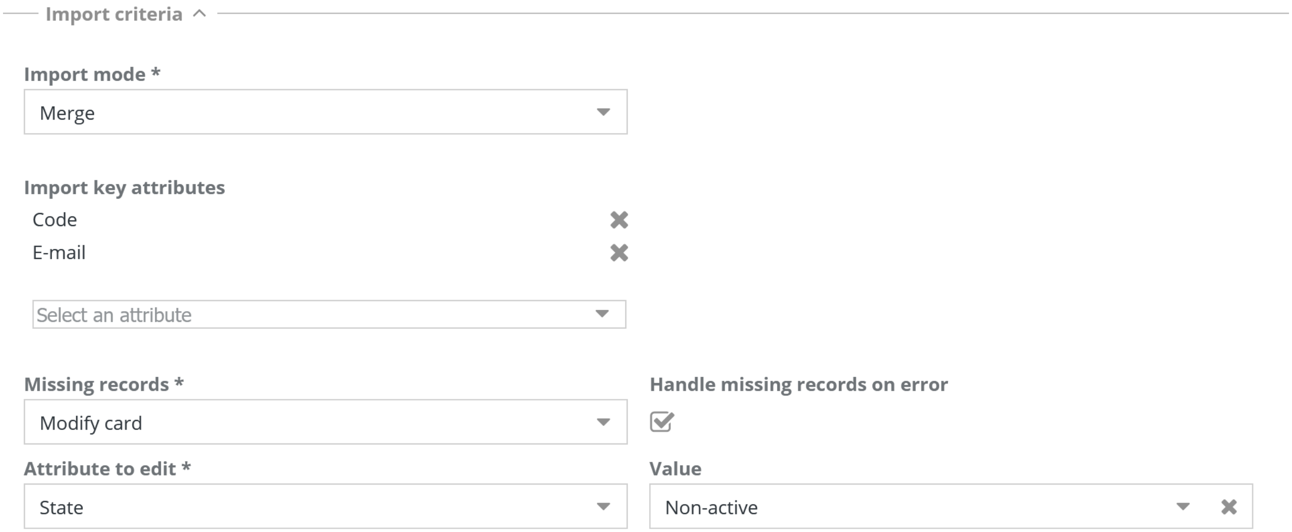
- Import modes: "Add" or "Merge"

In case of "Merge" mode, you are required to insert the following:

- Key attribute: you can specify a single or multiple attribute as univocal key to detect the CMDBuild data card to update
- Missing records: management mode of CMDBuild cards absent from the import file you want to synchronize, with the possibility to delete them, leave them untouched, edit an attribute representing the current state
- Handle missing records on error: it forces the execution of handing missing records according to the chosen criterion, also in case of an error during the import.

Only in case that for missing records you chose to edit an attribute that represents the current status, the following information is required:

- Attribute: in case the missing records are managed by editing a card attribute, it allows you to select it
- Value: in case the missing records are managed by editing a card attribute, it allows you to select the value



Import criteria

Import mode *
Merge

Import key attributes
Code
E-mail

Select an attribute

Missing records *
Modify card

Handle missing records on error
☑

Attribute to edit *
State

Value
Non-active

Export filter

The following information is required only for Export and Import / Export templates:

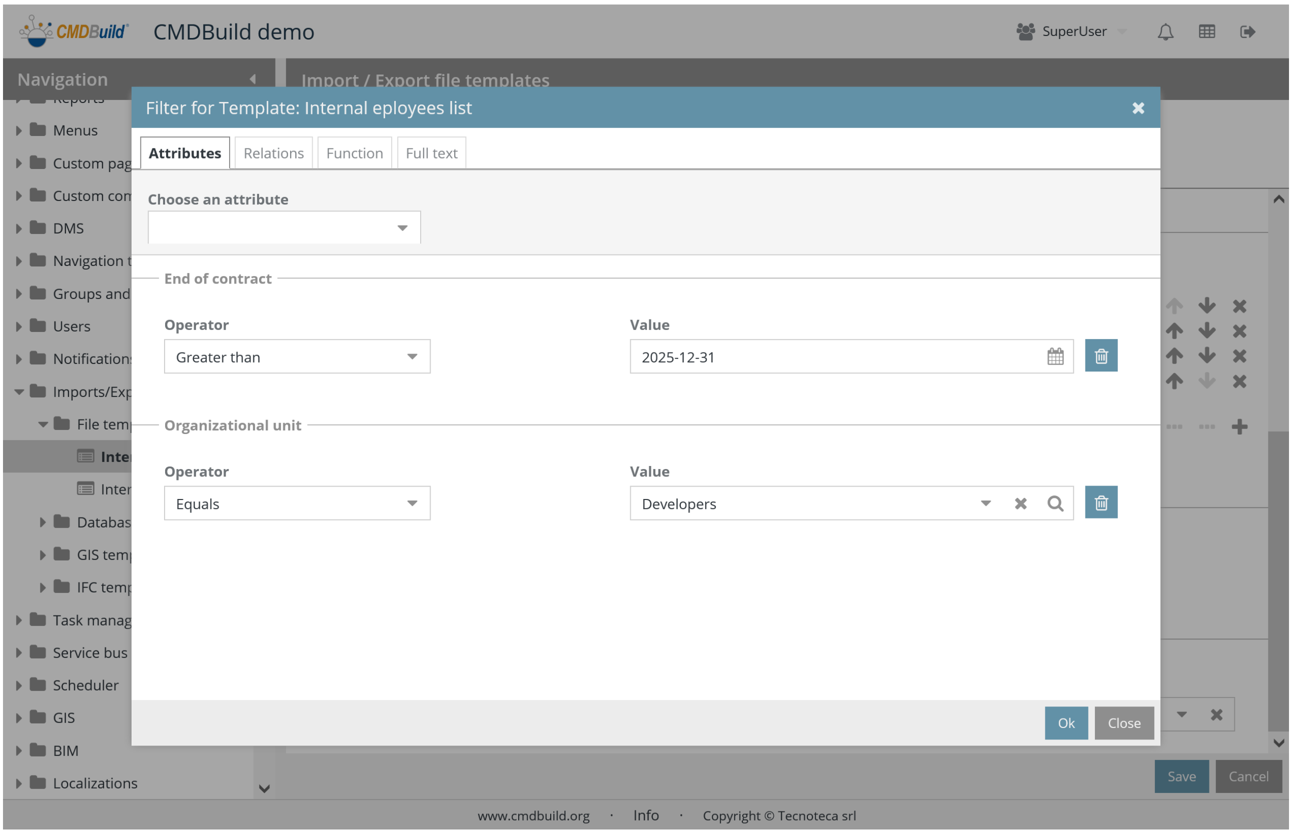
- Filer: it has to be applied to export only the information you need

Export filter

Filters

The definition of export filter is similar to the advanced filter management in the CMDBuild card management.

Here is a sample.



Filter for Template: Internal eployees list

Choose an attribute:

End of contract — Operator: Greater than — Value: 2025-12-31

Organizational unit — Operator: Equals — Value: Developers

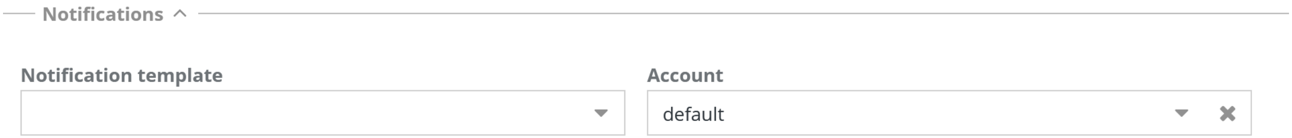
Notifications

The following information is required:

- Operations notification template: you have to choose a template that is configured in the notification management menu
- Account: it allows you to overwrite the standard account with another chosen account

In case of batch execution (through a specific task configured in the Task Manager) of the import or export function, the above-mentioned parameters are necessary in order to get news about the outcome of the operation.

In case of interactive execution, the same information is displayed in the user interface.



Notification template:

Account: default

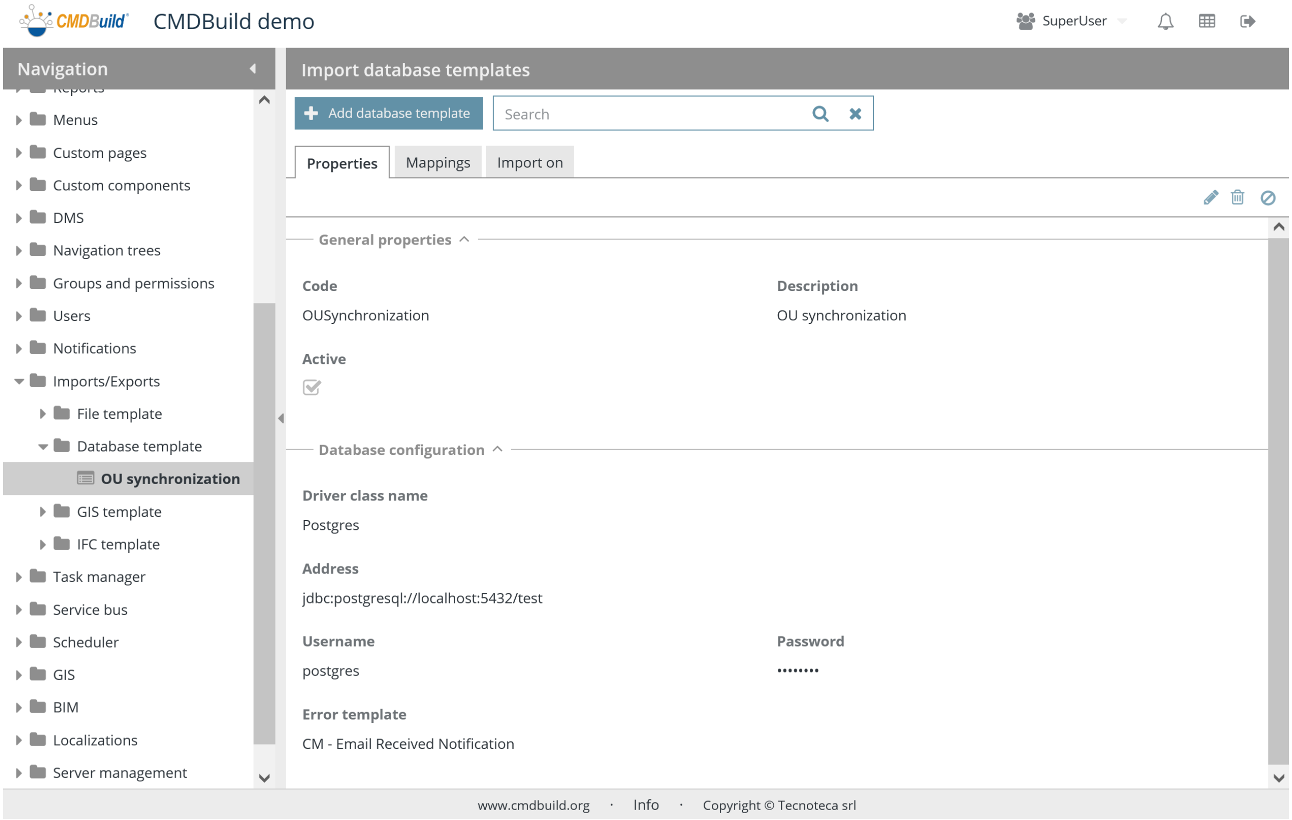
23.2. Database template

It is possible to define import templates from relational database like PostgreSQL, My SQL, Oracle or SQLServer.

The definition of the template can include the update of several classes of CMDBuild, starting with several charts of the relational database.

You can operate the data update in "merge" mode, updating the already available lines (recognized through a unique key), inserting the new ones and managing the missing ones in a custom way or in "add" mode.

23.2.1. Properties tab



Code: OUSynchronization

Description: OU synchronization

Active: ☑

Driver class name: Postgres

Address: jdbc:postgresql://localhost:5432/test

Username: postgres

Password: ••••••••

Error template: CM - Email Received Notification

There is the possibility of performing the following operations:

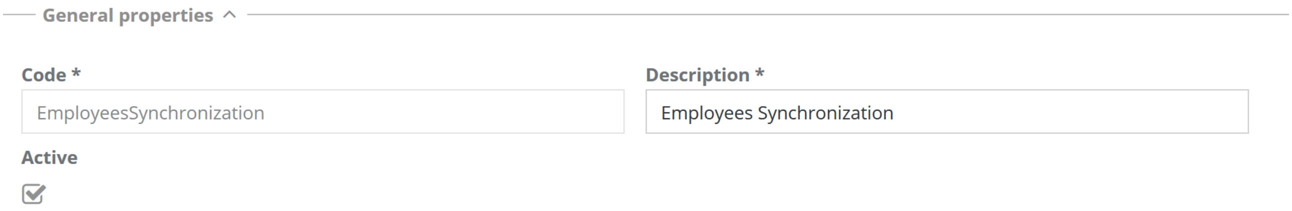
• create a new template	+ Add database template
• search the template in the list	Search... 🔍 ✖
• edit the selected template	✎
• delete the selected template	🗑
• disable the selected attribute (i.e. set the status as non-active)	⊘

Each value of the list requires some parameters to be compiled, as described below.

General properties

The following information is required:

- Code: template code
- Description: short template description
- Active: it states whether the template is active and can be used



General properties

Code *: EmployeesSynchronization

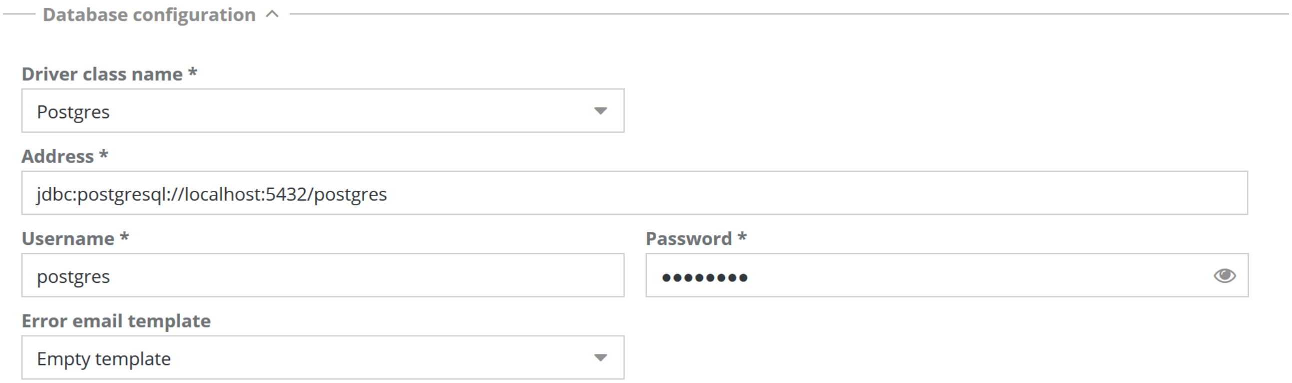
Description *: Employees Synchronization

Active: ☑

Database configuration

The following information is required:

- Driver name: PostgreSQL, MySQL, Oracle and SQLServer database are supported, having installed the proper access drivers
- Address: connections string to the database
- Username: username to access the database
- Password: password to access database
- Error email template: select one of the email templates for the notifications of some errors



Database configuration

Driver class name *: Postgres

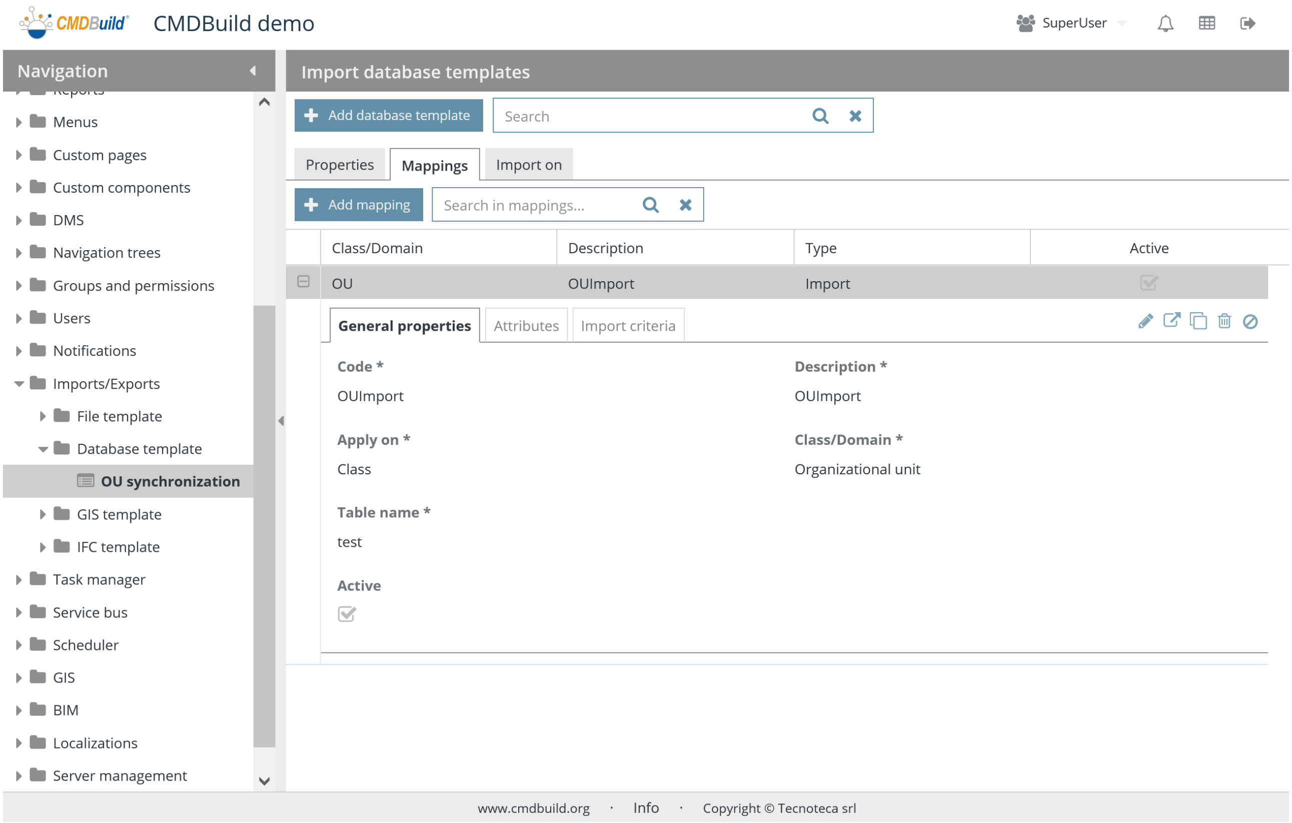
Address *: jdbc:postgresql://localhost:5432/postgres

Username *: postgres

Password *: ••••••••

Error email template: Empty template

23.2.2. TAB Mapping



Class/Domain	Description	Type	Active
OU	OUImport	Import	✓

Code *
OUImport

Description *
OUImport

Apply on *
Class

Class/Domain *
Organizational unit

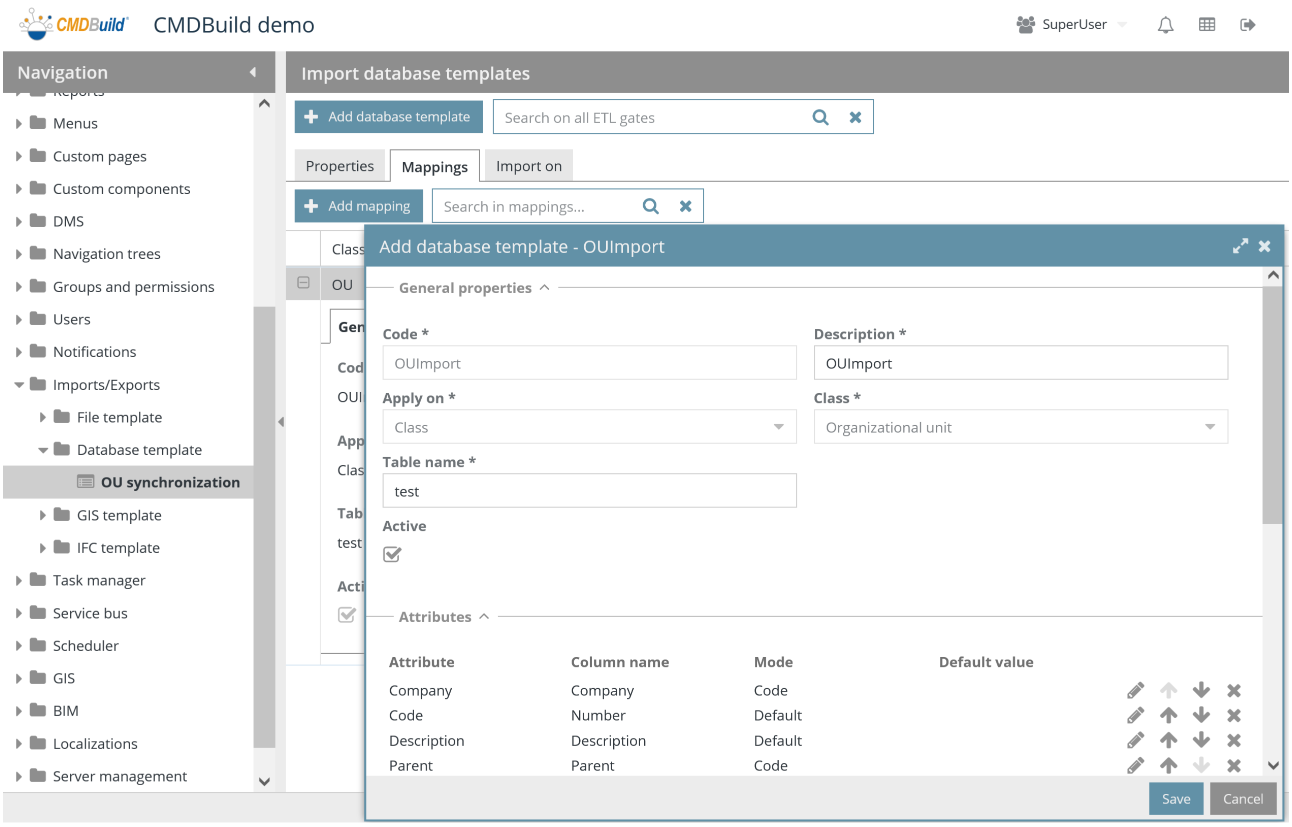
Table name *
test

Active
✓

There is the possibility of performing the following operations:

• add a new mapping	✚ Add mapping
• search the mapping in the list	Search... 🔍 ✖
• edit the selected mapping	✎
• open the selected mapping in edit	↗
• clone the selected mapping	⧉
• delete the selected mapping	🗑
• disable the selected mapping (i.e. set the status as non-active)	⊘

Each attribute needs several parameters (metadata) to be compiled, as described mentioned below.



Add database template - OUImport

General properties

Code *: OUImport

Description *: OUImport

Apply on *: Class

Class *: Organizational unit

Table name *: test

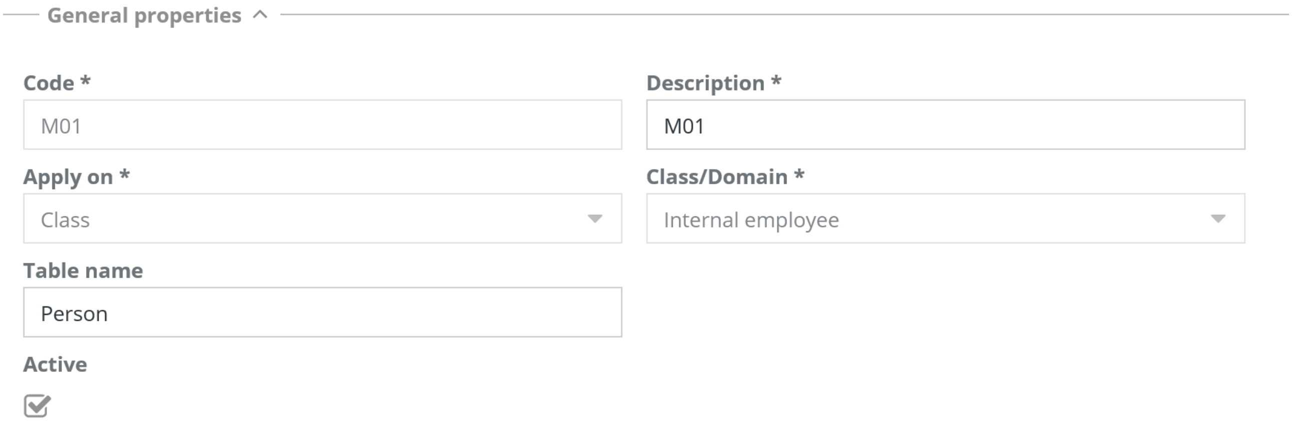
Active: ☑

Attributes

Attribute	Column name	Mode	Default value
Company	Company	Code	
Code	Number	Default	
Description	Description	Default	
Parent	Parent	Code	

General properties

- Code: mapping code
- Description: mapping description
- Apply on: classes or domains
- Class / domain: it requires to select the class or domain
- Table name: database origin table
- Active: the mapping is active and can be used


General properties

Code *: M01

Description *: M01

Apply on *: Class

Class/Domain *: Internal employee

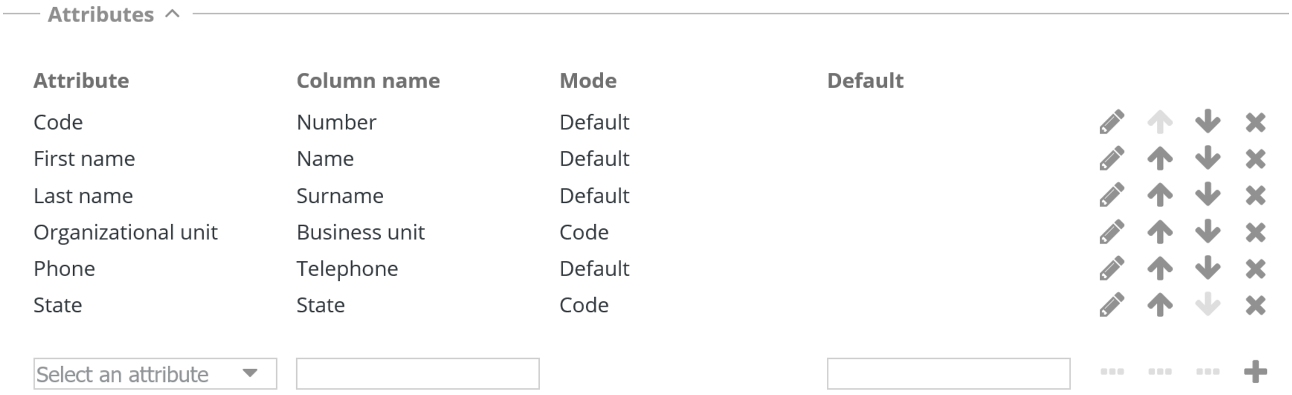
Table name: Person

Active: ☑

Attributes

The following information is required for the setting of attribute mapping:

- Attribute: attribute name of the class in CMDBuild
- Column name: name of the column in the import or output file
- Mode: in case of Reference or Lookup CMDBuild attributes, it allows you to specify whether the date has to be understood as "Id" (unique identification in CMDBuild), as "Code" (standard attribute of all CMDBuild classes) or as "Description" (standard attribute of all CMDBuild classes)
- Default: in case of Import template, it allows you to specify a default value if missing in the file to charge



Attribute	Column name	Mode	Default
Code	Number	Default	
First name	Name	Default	
Last name	Surname	Default	
Organizational unit	Business unit	Code	
Phone	Telephone	Default	
State	State	Code	

Import criteria

The following information is required:

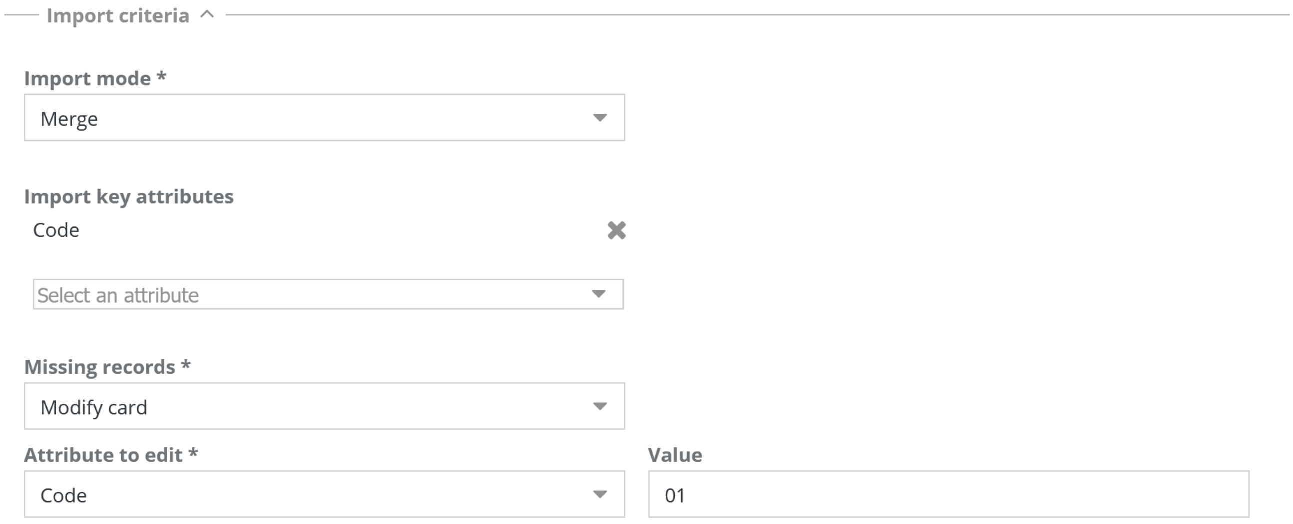
- Import modes: "Add" or "Merge"

In case of "Merge" mode, you are required to insert the following:

- Key attribute: you can specify a single or multiple attribute as univocal key to detect the CMDBuild data card to update
- Missing records: management mode of CMDBuild cards absent from the import file you want to synchronize, with the possibility to delete them, leave them untouched, edit an attribute representing the current state

Only in case you chose to edit an attribute representing the current state in the previous step, here are required the following additional information:

- Attribute: in case the missing records are managed by editing a card attribute, it allows you to select it
- Value: in case the missing records are managed by editing a card attribute, it allows you to select the value



Import criteria

Import mode *

Merge

Import key attributes

Code

Select an attribute

Missing records *

Modify card

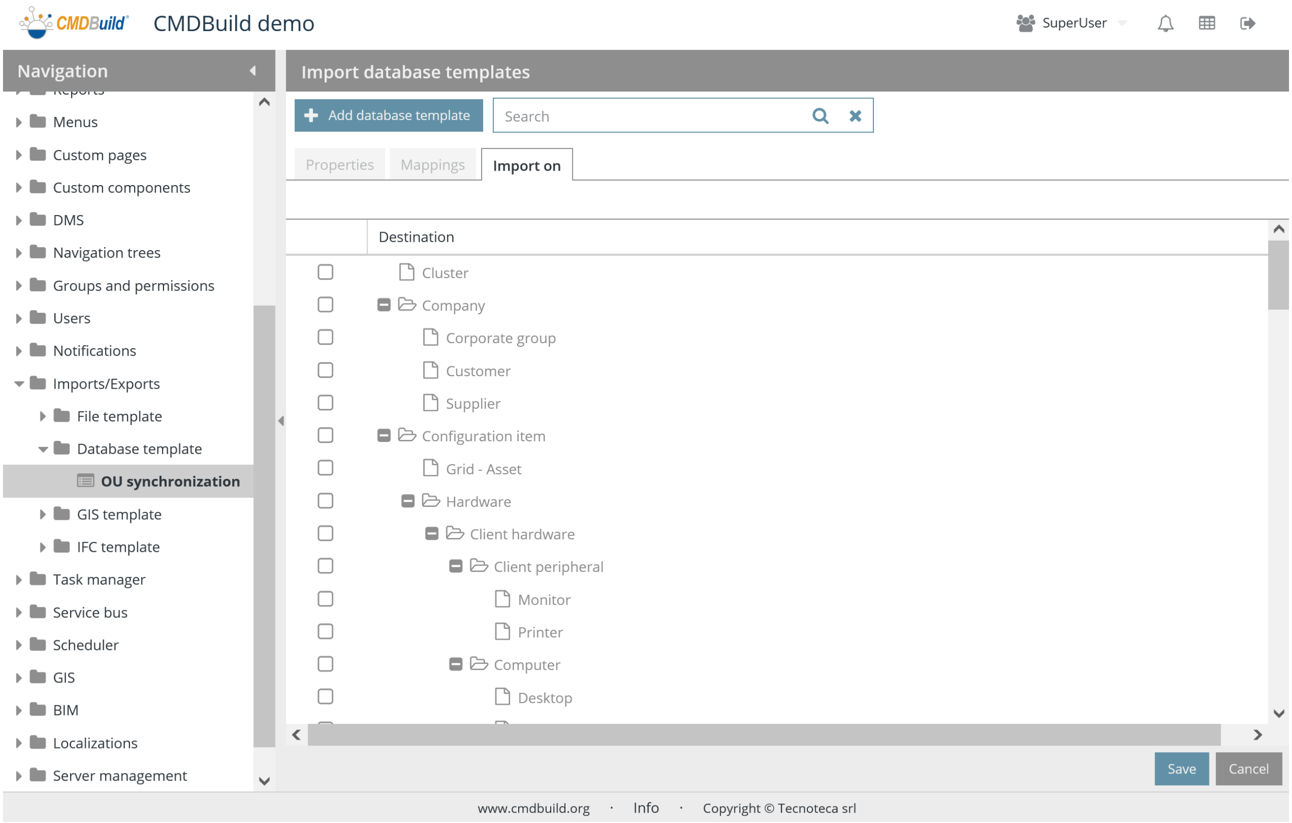
Attribute to edit *

Code

Value

01

23.2.3. TAB Import



You can select the classes for which the Import function has to appear in the contextual menu through the current template.

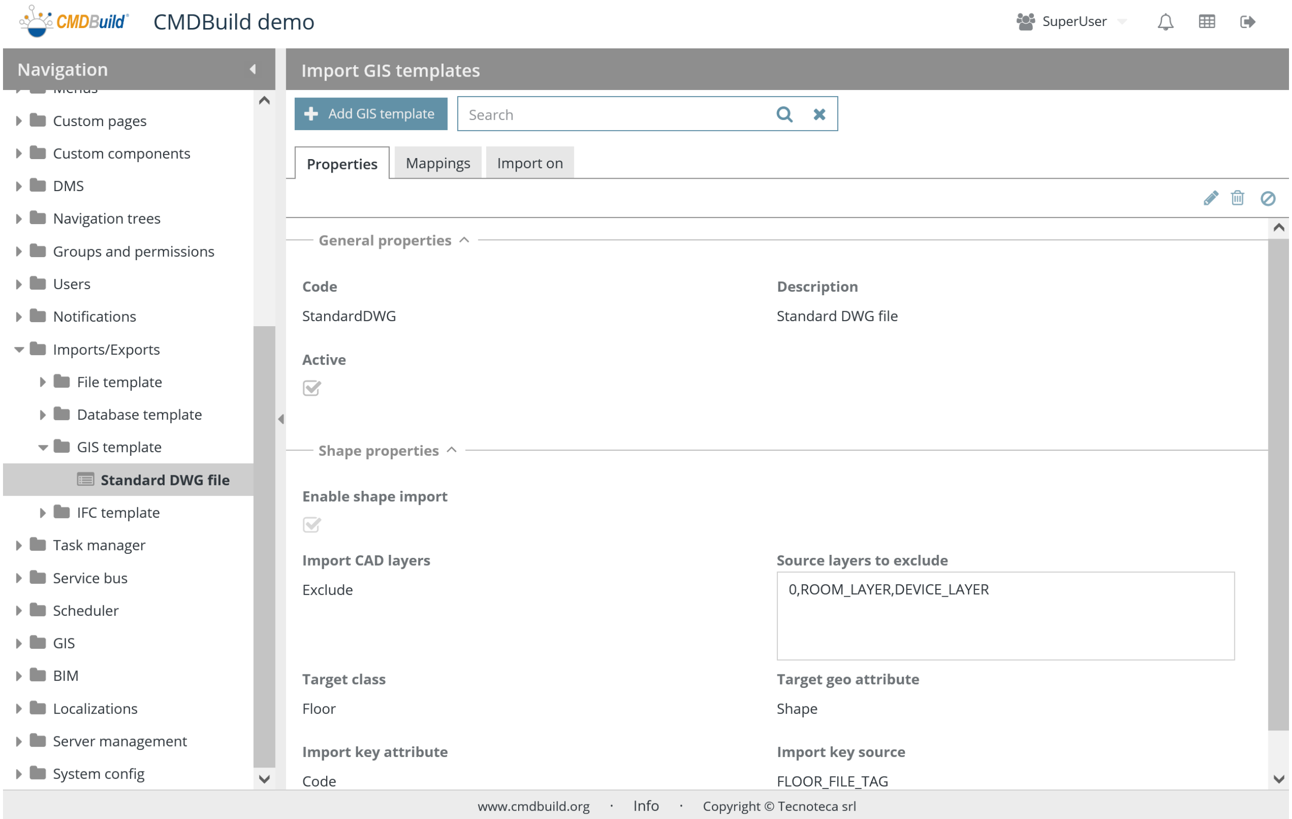
23.3. GIS template

The import template can be defined from planimetries included in DWG files.

The definition of the template can include the update of several classes of CMDBuild, starting with several planimetries layers.

Data are always updated in "merge" mode, updating the already available lines (recognized through a unique key), inserting the new ones and managing the missing ones in a custom way.

23.3.1. Properties tab



General properties

Code: StandardDWG

Description: Standard DWG file

Active: ☑

Shape properties

Enable shape import: ☑

Import CAD layers: Exclude

Source layers to exclude: 0,ROOM_LAYER,DEVICE_LAYER

Target class: Floor

Target geo attribute: Shape

Import key attribute: Code

Import key source: FLOOR_FILE_TAG

There is the possibility of performing the following operations:

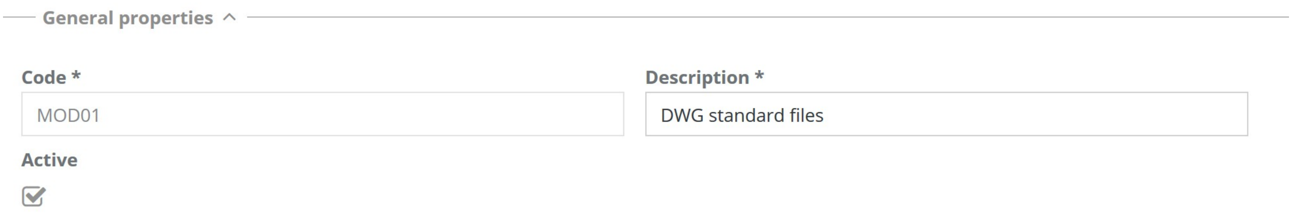
• create a new template	+ Add GIS template
• search the template in the list	Search... 🔍 ✖
• edit the selected template	✎
• delete the selected template	🗑
• disable the selected attribute (i.e. set the status as non-active)	⊘

Each value of the list requires some parameters to be compiled, as described below.

General properties

The following information is required:

- Code: template code
- Description: short template description
- Active: it states whether the template is active and can be used



General properties

Code *
MOD01

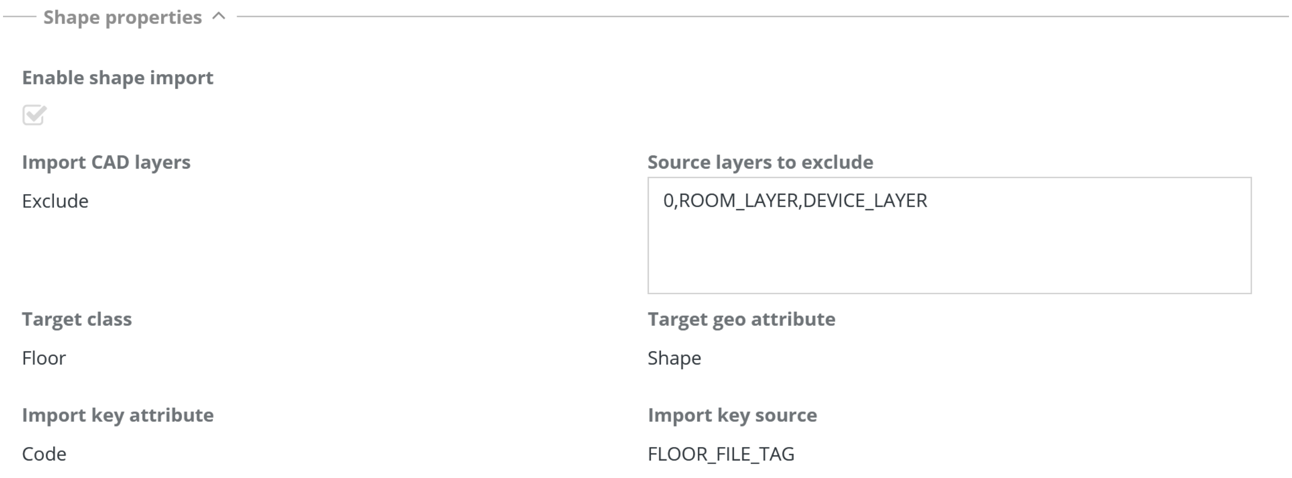
Description *
DWG standard files

Active
☑

Shape properties

The following information is required:

- Enable Shape import: planimetries components not refered to CMDBuild items (walls, windows, stairs, etc.) have to be imported into Geoserver as background files, in case the flag is selected further parameters will appear.
- Layer to import: all, elements to include or to exclude
- List of elements to include / exclude
- Target class
- Target geo attribute
- Import key attribute
- Import key source



Shape properties

Enable shape import
☑

Import CAD layers
Exclude

Source layers to exclude
0,ROOM_LAYER,DEVICE_LAYER

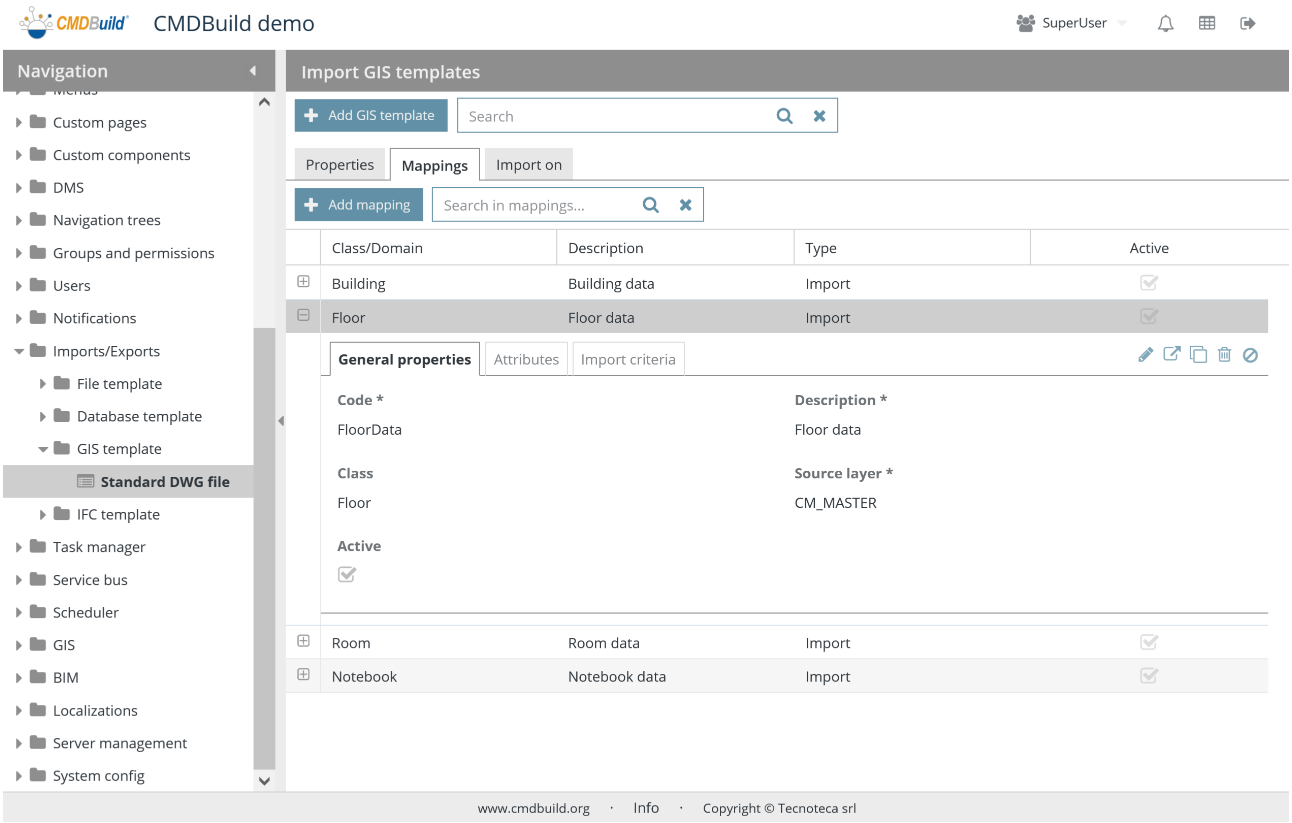
Target class
Floor

Target geo attribute
Shape

Import key attribute
Code

Import key source
FLOOR_FILE_TAG

23.3.2. TAB Mapping



Class/Domain	Description	Type	Active
Building	Building data	Import	✓
Floor	Floor data	Import	✓
Room	Room data	Import	✓
Notebook	Notebook data	Import	✓

Code *
FloorData

Description *
Floor data

Class
Floor

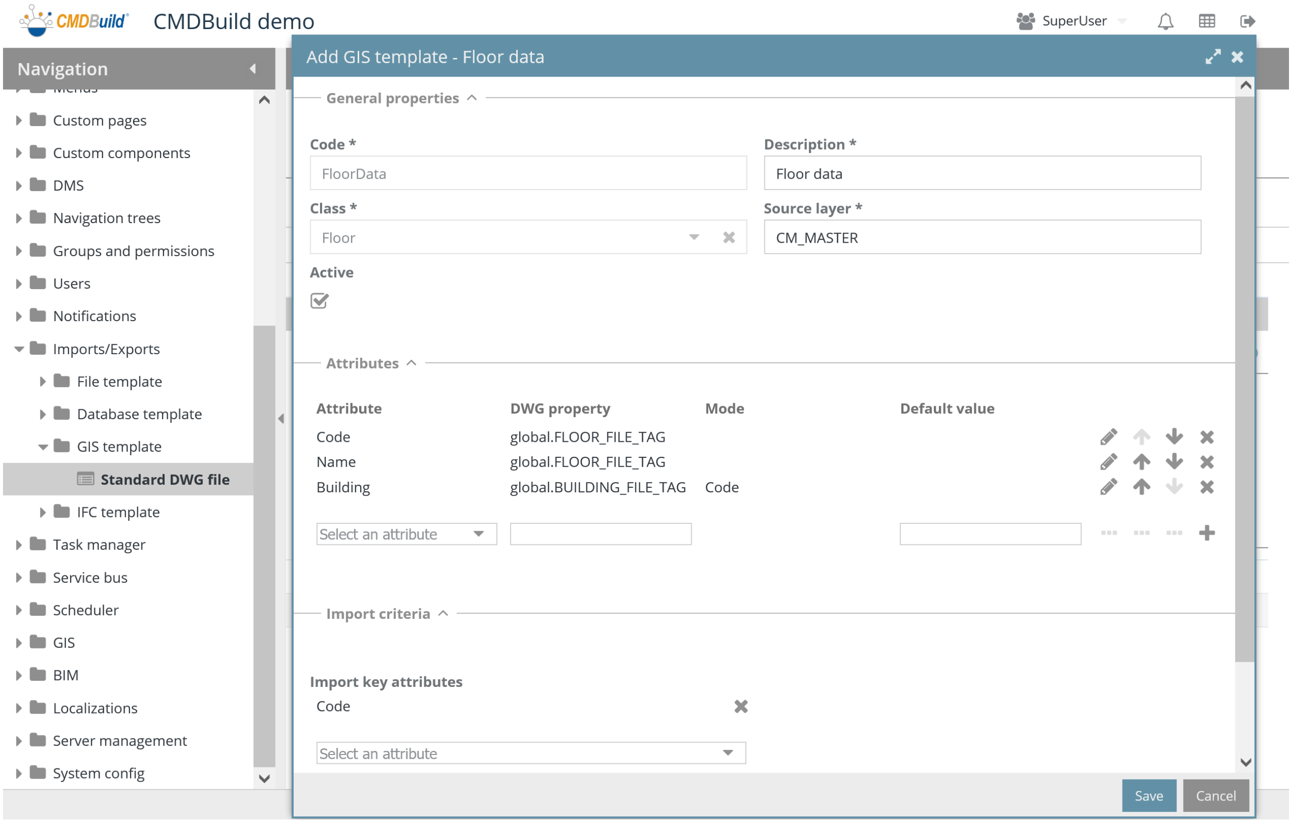
Source layer *
CM_MASTER

Active
✓

There is the possibility of performing the following operations:

• add a new mapping	✚ Add mapping
• search the mapping in the list	Search in mappings... 🔍 ✖
• edit the selected mapping	✎
• open the selected mapping in edit	↗
• clone the selected mapping	⧉
• delete the selected mapping	🗑
• disable the selected mapping (i.e. set the status as non-active)	⊘

Each attribute needs several parameters (metadata) to be compiled, as described mentioned below.



Add GIS template - Floor data

General properties

Code *: FloorData

Description *: Floor data

Class *: Floor

Source layer *: CM_MASTER

Active ☑

Attributes

Attribute	DWG property	Mode	Default value
Code	global.FLOOR_FILE_TAG		
Name	global.FLOOR_FILE_TAG		
Building	global.BUILDING_FILE_TAG	Code	

Select an attribute

Import criteria

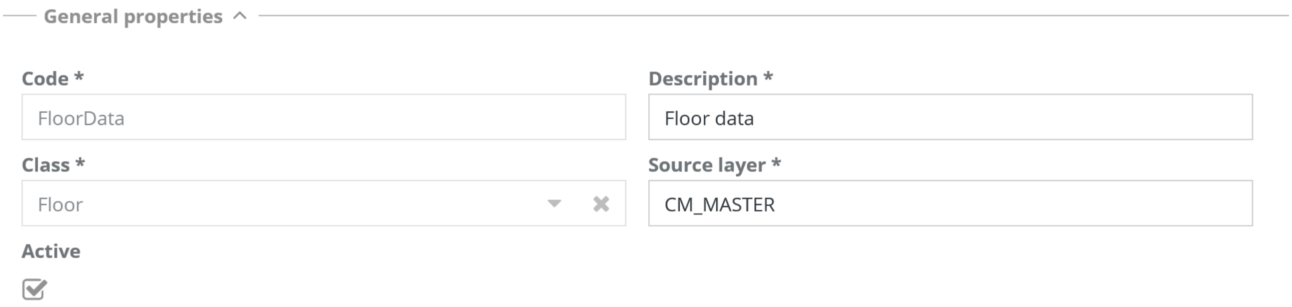
Import key attributes

Code

Select an attribute

General properties

- Code: mapping code
- Description: mapping description
- Class/Domain: for import mapping
- Source layer: source layer in the DWG file
- Active: the mapping is active and can be used



General properties

Code *: FloorData

Description *: Floor data

Class *: Floor

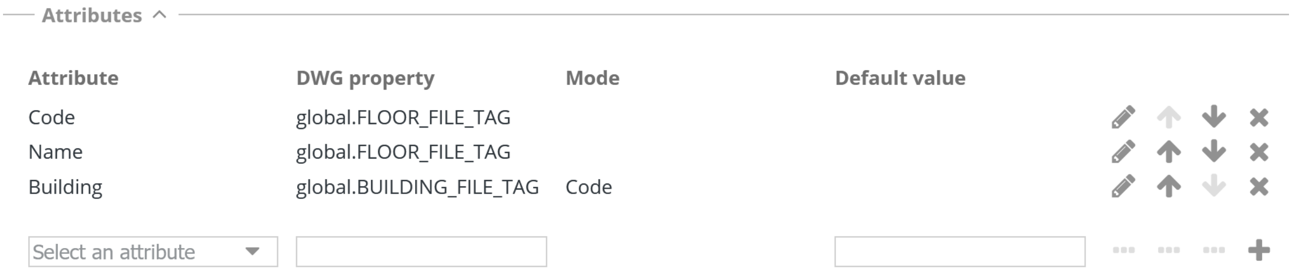
Source layer *: CM_MASTER

Active ☑

Attributes

The following information is required for the setting of attribute mapping:

- Attribute: attribute name of the class in CMDBuild
- DWG property: property name in the DWG file
- Mode: in case of Reference or Lookup CMDBuild attributes, it allows you to specify whether the date has to be understood as "Id" (unique identification in CMDBuild), as "Code" (standard attribute of all CMDBuild classes) or as "Description" (standard attribute of all CMDBuild classes)
- Default value



Attribute	DWG property	Mode	Default value
Code	global.FLOOR_FILE_TAG		
Name	global.FLOOR_FILE_TAG		
Building	global.BUILDING_FILE_TAG	Code	

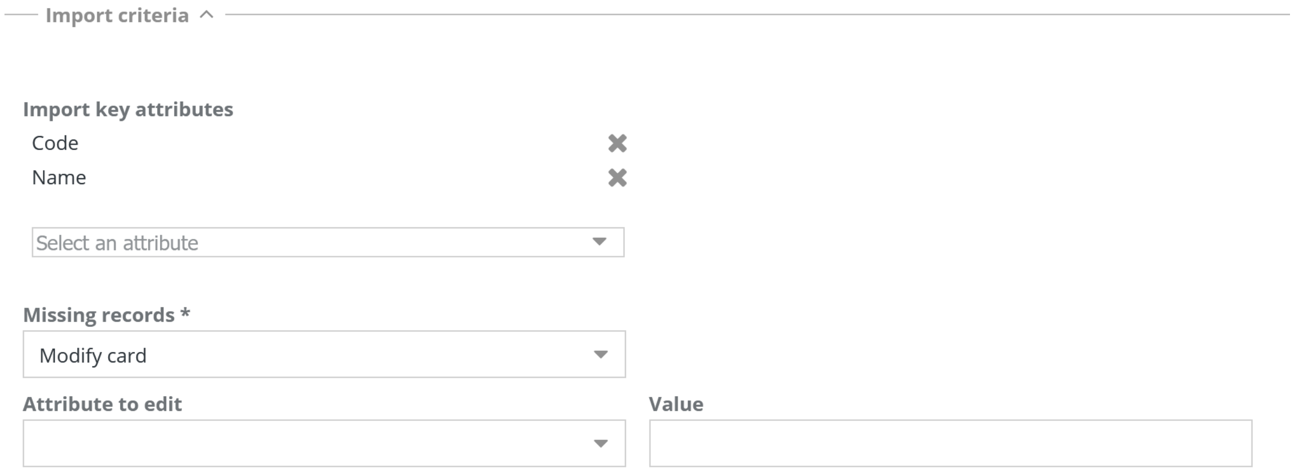
Import criteria

The following information is required:

- Key attribute: you can specify a single or multiple attribute as univocal key to detect the CMDBuild data card to update
- Missing records: management mode of CMDBuild cards absent from the import file you want to synchronize, with the possibility to delete them, leave them untouched, edit an attribute representing the current state

Only in case you chose to edit an attribute representing the current state in the previous step, here are required the following additional information:

- Attribute: in case the missing records are managed by editing a card attribute, it allows you to select it
- Value: in case the missing records are managed by editing a card attribute, it allows you to select the value



Import criteria

Import key attributes

Code ✖

Name ✖

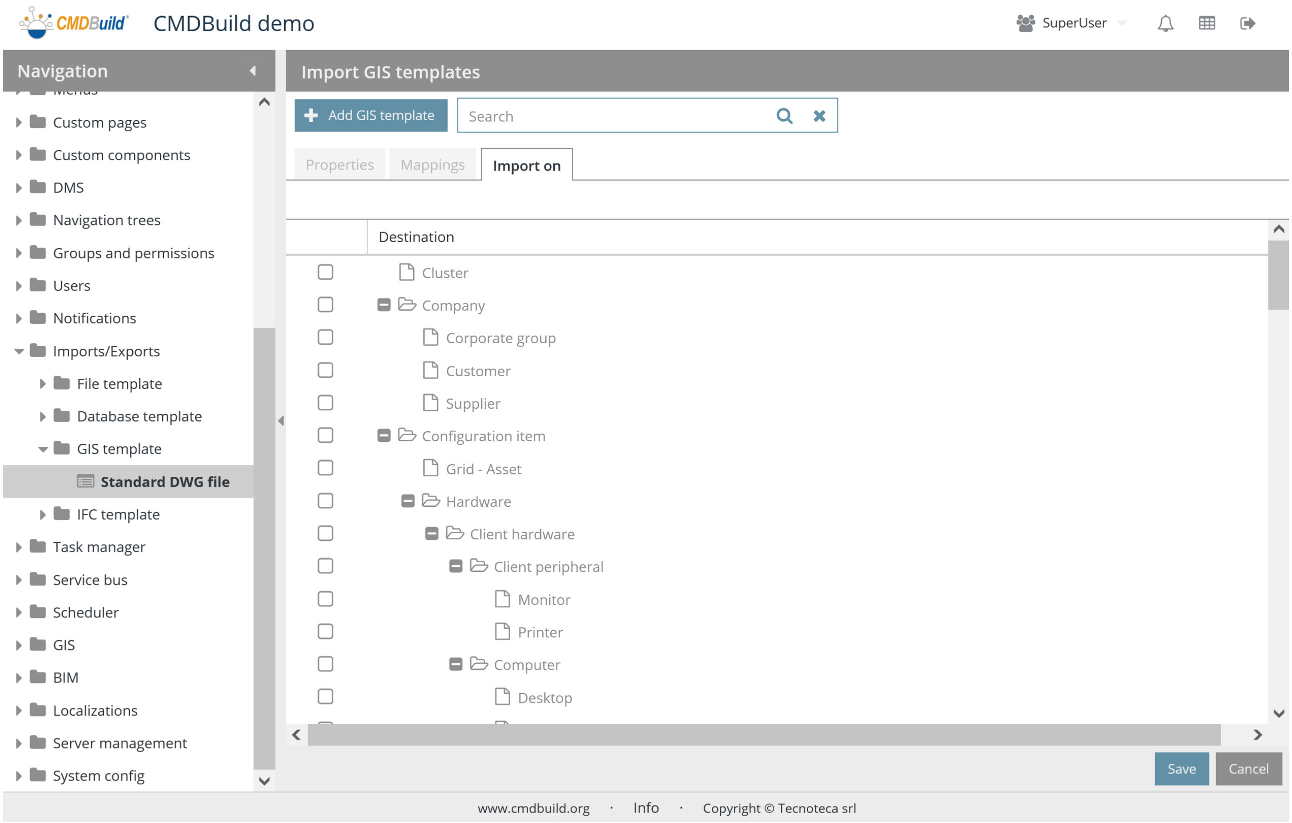
Select an attribute

Missing records *: Modify card

Attribute to edit:

Value:

23.3.3. TAB Import



You can select the classes for which the Import function has to appear in the contextual menu through the current template.

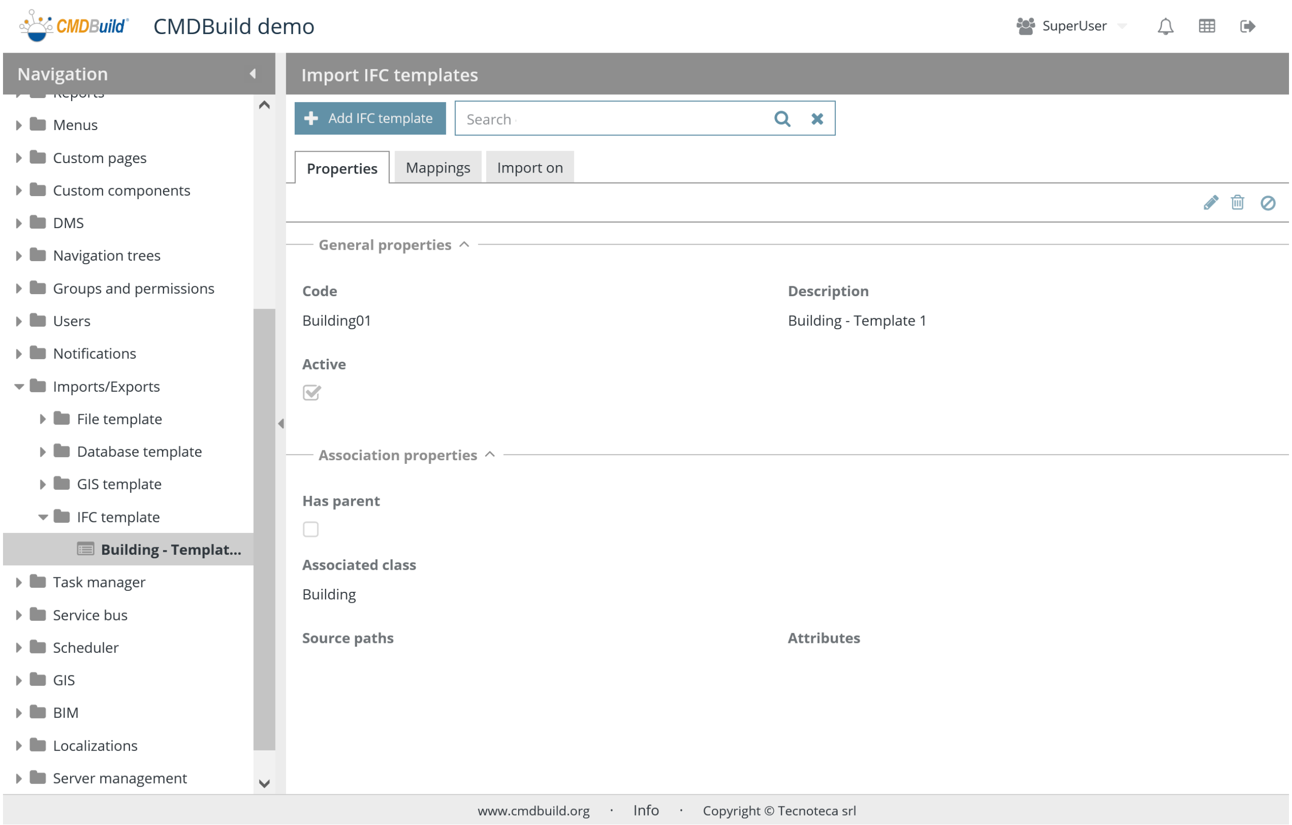
23.4. BIM template

The import template can be defined from BIM models included in IFC exchange files (Industry Foundation Classes).

The definition of the template can include the update of several classes of CMDBuild, starting with several entities described in the model.

Data are always updated in "merge" mode, updating the already available lines (recognized through a unique key), inserting the new ones and managing the missing ones in a custom way.

23.4.1. Properties tab



General properties

Code: Building01

Description: Building - Template 1

Active: ☑

Association properties

Has parent: □

Associated class: Building

Source paths

Attributes

There is the possibility of performing the following operations:

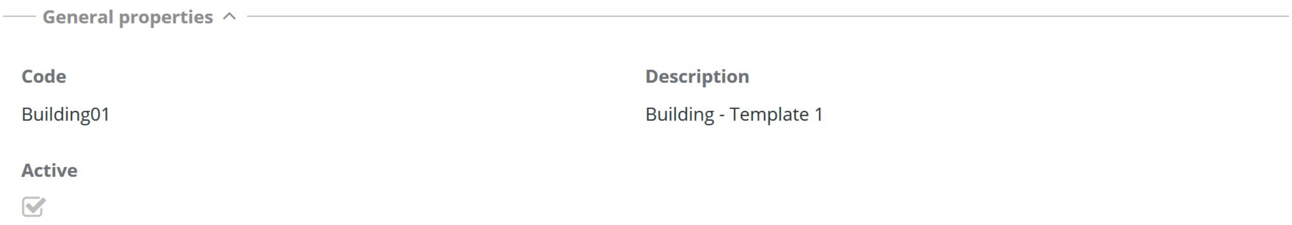
• create a new template	+ Add IFC template
• search the template in the list	Search... 🔍 ✖
• edit the selected template	✎
• delete the selected template	🗑
• disable the selected attribute (i.e. set the status as non-active)	⊘

Each value of the list requires some parameters to be compiled, as described below.

General properties

The following information is required:

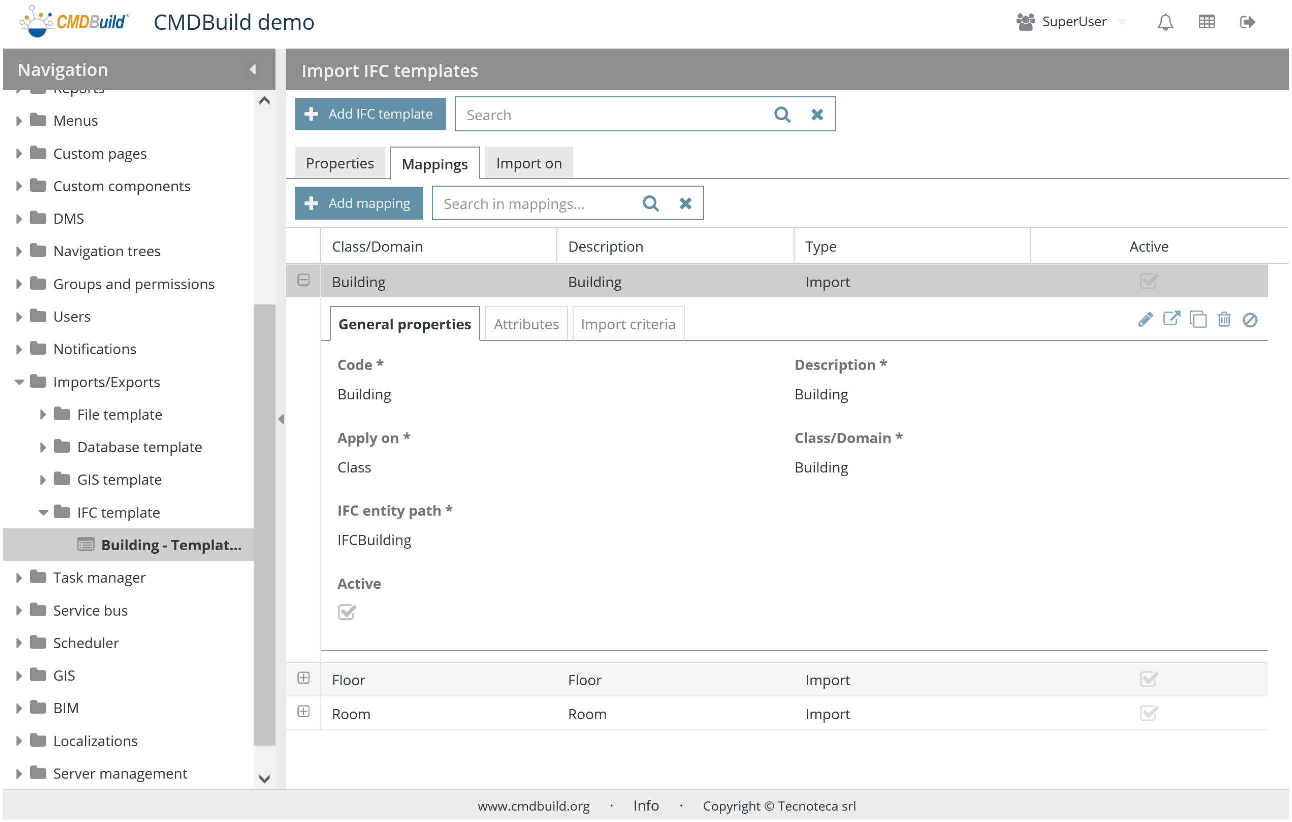
- Code: template code
- Description: short template description
- Active: it states whether the template is active and can be used



General properties

Code Building01	Description Building - Template 1
Active ☑	

23.4.2. TAB Mapping



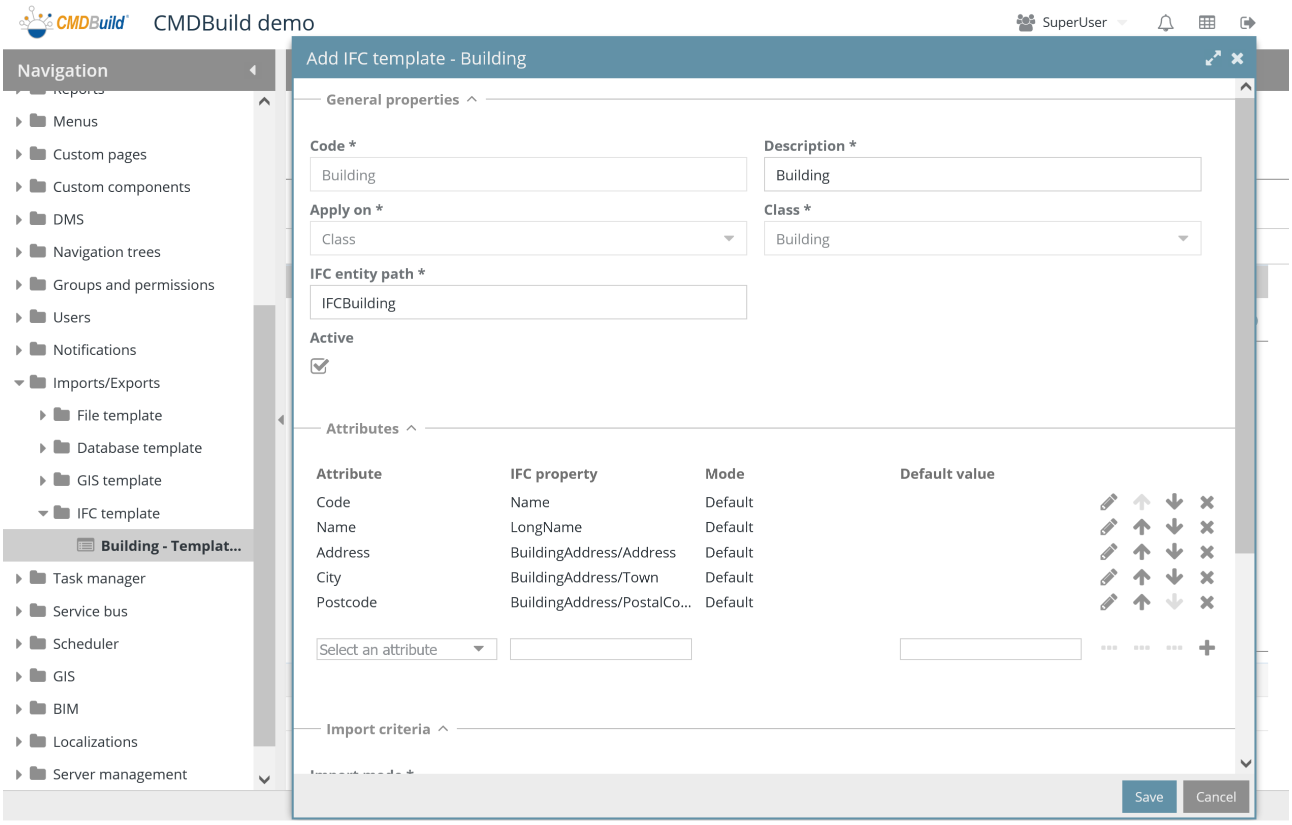
Class/Domain	Description	Type	Active
Building	Building	Import	☑
Floor	Floor	Import	☑
Room	Room	Import	☑

Code *: Building
Description *: Building
Apply on *: Class
Class/Domain *: Building
IFC entity path *: IFCBuilding
Active: ☑

There is the possibility of performing the following operations:

• add a new mapping	+ Add mapping
• search the mapping in the list	Search in mappings... 🔍 ✖
• edit the selected mapping	✎
• open the selected mapping in edit	↗
• clone the selected mapping	⧉
• delete the selected mapping	🗑
• disable the selected mapping (i.e. set the status as non-active)	⊘

Each attribute needs some parameters (metadata) to be compiled, as described mentioned below.



Add IFC template - Building

General properties

Code *: Building

Description *: Building

Apply on *: Class

Class *: Building

IFC entity path *: IFCBuilding

Active: ☑

Attributes

Attribute	IFC property	Mode	Default value
Code	Name	Default	
Name	LongName	Default	
Address	BuildingAddress/Address	Default	
City	BuildingAddress/Town	Default	
Postcode	BuildingAddress/PostalCo...	Default	

Select an attribute

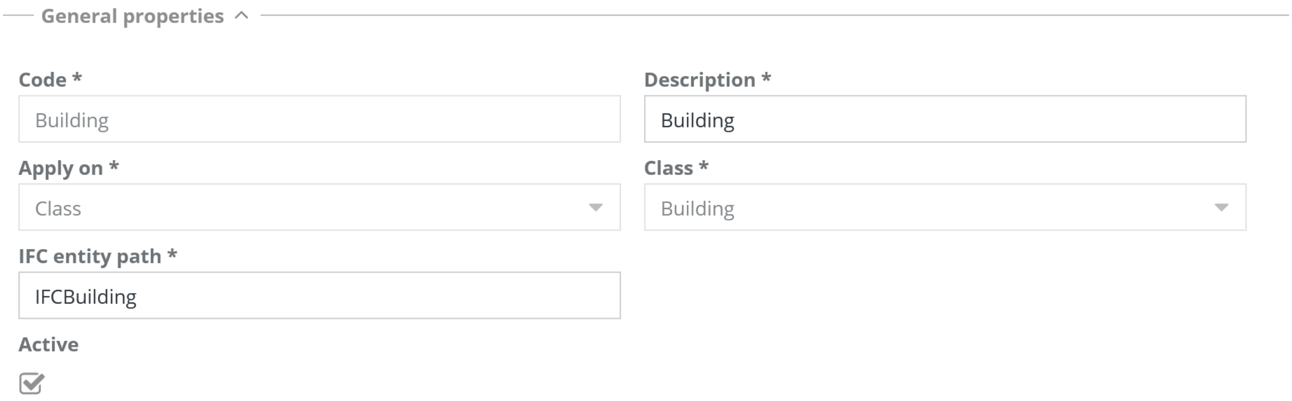
Import criteria

Import mode *

General properties

- Code: mapping code
- Description: mapping description
- Apply on: classes or domains
- Class / domain: it requires to select the class or domain

- IFC entity path: entity location in the IFC file structure
- Active: the mapping is active and can be used



General properties

Code *: Building

Description *: Building

Apply on *: Class

Class *: Building

IFC entity path *: IFCBuilding

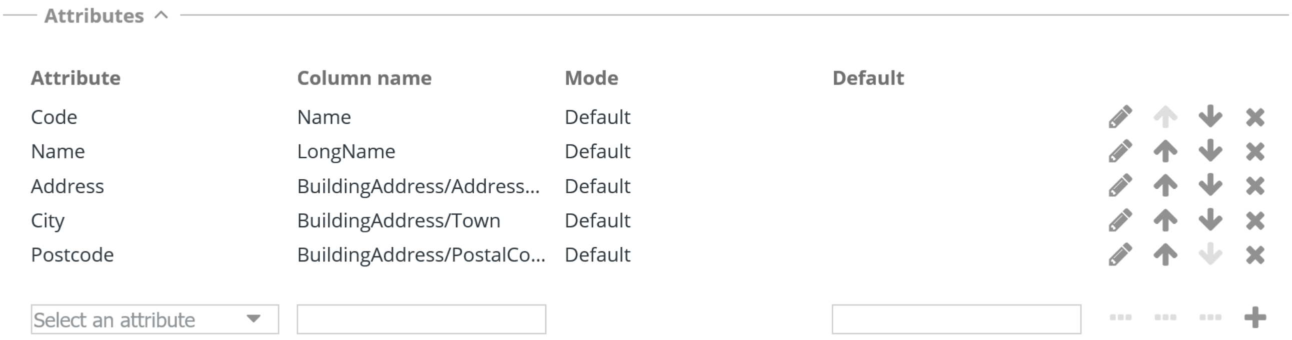
Active

☑

Attributes

The following information is required for the setting of attribute mapping:

- Attribute: attribute name of the class in CMDBuild
- Column name: property in the IFC file
- Mode: in case of Reference or Lookup CMDBuild attributes, it allows you to specify whether the date has to be understood as "Id" (unique identification in CMDBuild), as "Code" (standard attribute of all CMDBuild classes) or as "Description" (standard attribute of all CMDBuild classes)



Attributes

Attribute	Column name	Mode	Default
Code	Name	Default	
Name	LongName	Default	
Address	BuildingAddress/Address...	Default	
City	BuildingAddress/Town	Default	
Postcode	BuildingAddress/PostalCo...	Default	

Select an attribute

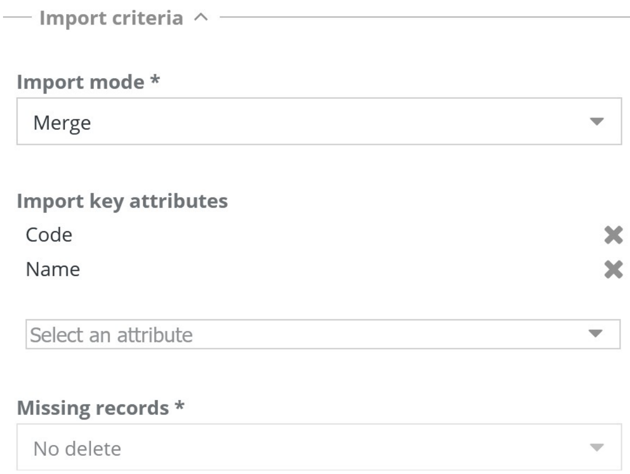
Import criteria

The following information is required:

- Key attribute: you can specify a single or multiple attribute as univocal key to detect the CMDBuild data card to update
- Missing records: management mode of CMDBuild cards absent from the import file you want to synchronize, with the possibility to delete them, leave them untouched, edit an attribute representing the current state

Only in case you chose to edit an attribute representing the current state in the previous step, here are required the following additional information:

- Attribute: in case the missing records are managed by editing a card attribute, it allows you to select it
- Value: in case the missing records are managed by editing a card attribute, it allows you to select the value



Import criteria

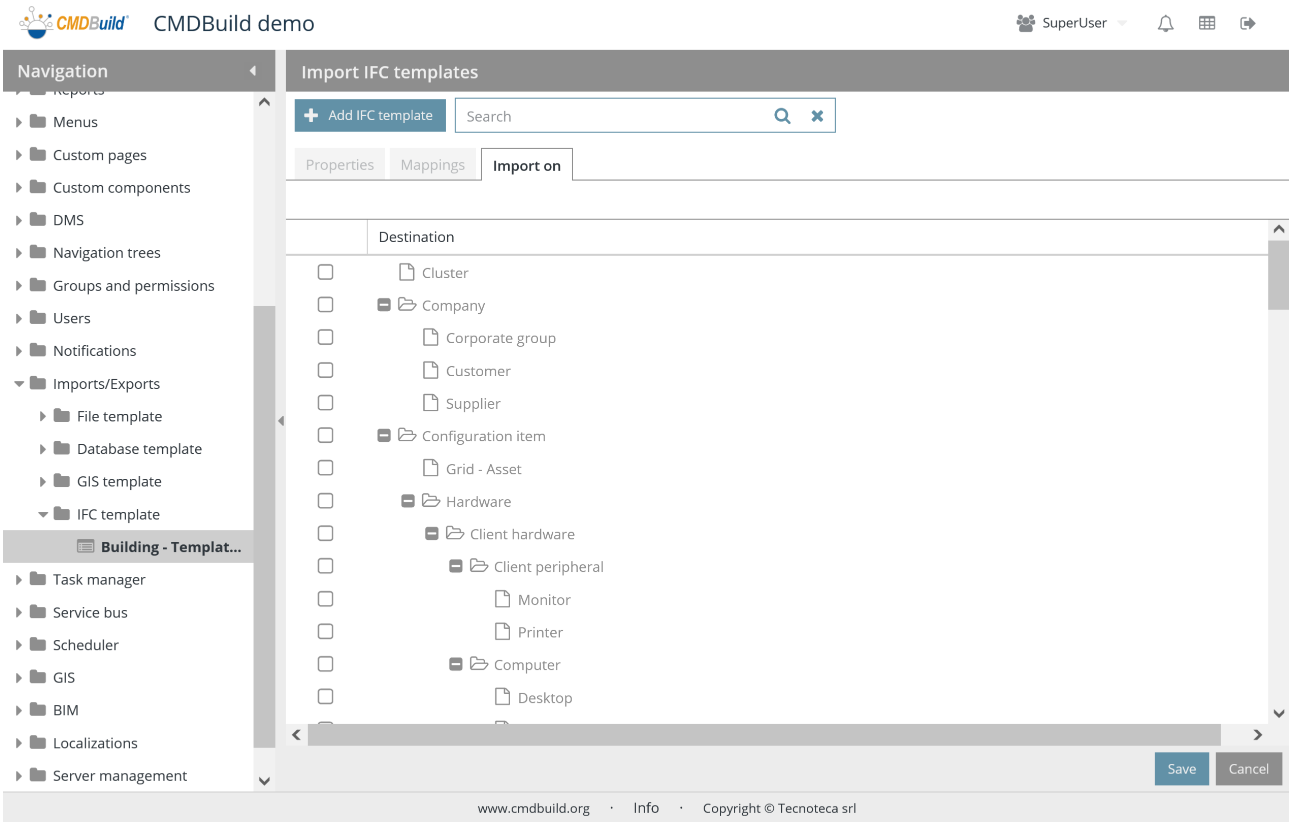
Import mode *: Merge

Import key attributes

- Code
- Name

Missing records *: No delete

23.4.3. TAB Import



You can select the classes for which the Import function has to appear in the contextual menu through the current template.

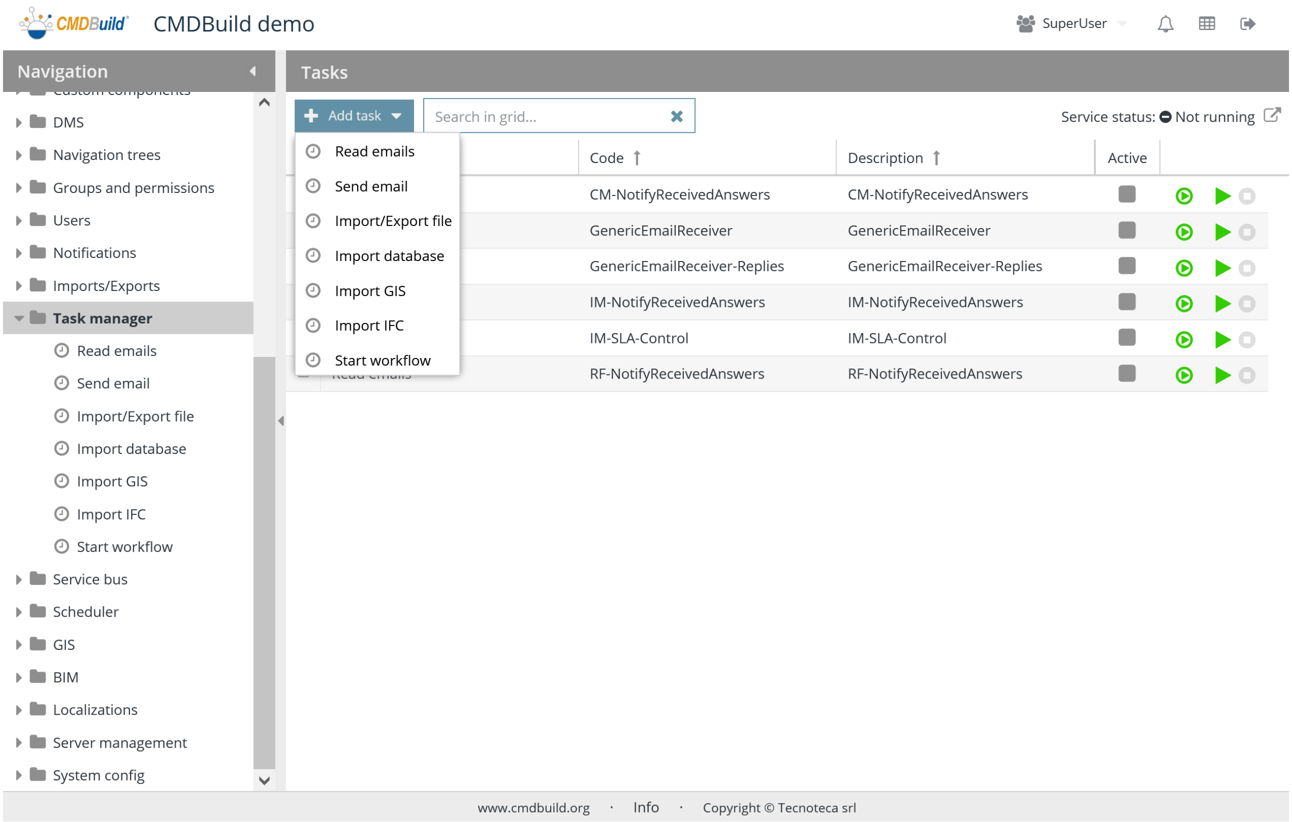
24. Task manager

CMDBuild allows you to configure automatic operations executed in background by a Task Manager integrated in the system.

The Task Manager groups the management of such operations. In this way you can have a general view on the active tasks and on the execution parameters.

The types of available tasks are:

- inbox: control the e-mail inbox, file emails into the system together with any attachments, notify their incoming and start a workflow
- send email, also with attached report
- scheduling of Import / Export files, based on file template
- scheduling of Import from database, based on database template
- scheduling of Import GIS, based on GIS template
- scheduling of IFC Import, based on IFC template
- Scheduling workflow: automatic start of a workflow



Code	Description	Active
CM-NotifyReceivedAnswers	CM-NotifyReceivedAnswers	■
GenericEmailReceiver	GenericEmailReceiver	■
GenericEmailReceiver-Replies	GenericEmailReceiver-Replies	■
IM-NotifyReceivedAnswers	IM-NotifyReceivedAnswers	■
IM-SLA-Control	IM-SLA-Control	■
RF-NotifyReceivedAnswers	RF-NotifyReceivedAnswers	■

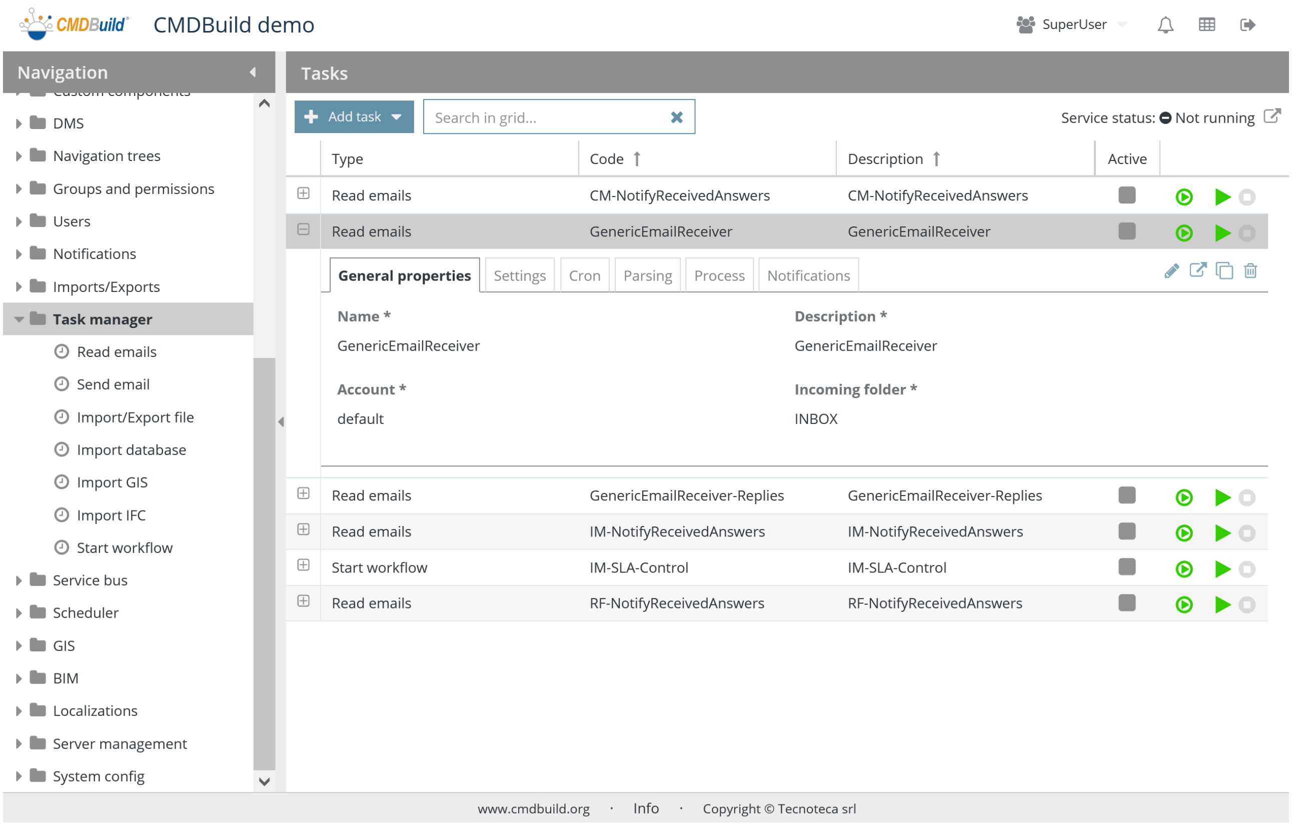
There is the possibility of performing the following operations:

• configure a new task	+ Add task

- search a task in the list	Search... 🔍 ✖
- running a single task	⏵
- start a task	▶
- stop the execution of a task	⏹
- expand the current row and display the task data	⊞
- edit the selected task	✎
- display the complete information related to the selected task	↗
- clone the selected task	⧉
- delete the selected task	🗑

The current configuration of the Task Manager is filed in the DB and every time CMDBuild restarts the active tasks are automatically restarted.

Below an example of list and task detail.



Service status: Not running

Type	Code	Description	Active
Read emails	CM-NotifyReceivedAnswers	CM-NotifyReceivedAnswers	
Read emails	GenericEmailReceiver	GenericEmailReceiver	
Read emails	GenericEmailReceiver-Replies	GenericEmailReceiver-Replies	
Read emails	IM-NotifyReceivedAnswers	IM-NotifyReceivedAnswers	
Start workflow	IM-SLA-Control	IM-SLA-Control	
Read emails	RF-NotifyReceivedAnswers	RF-NotifyReceivedAnswers	

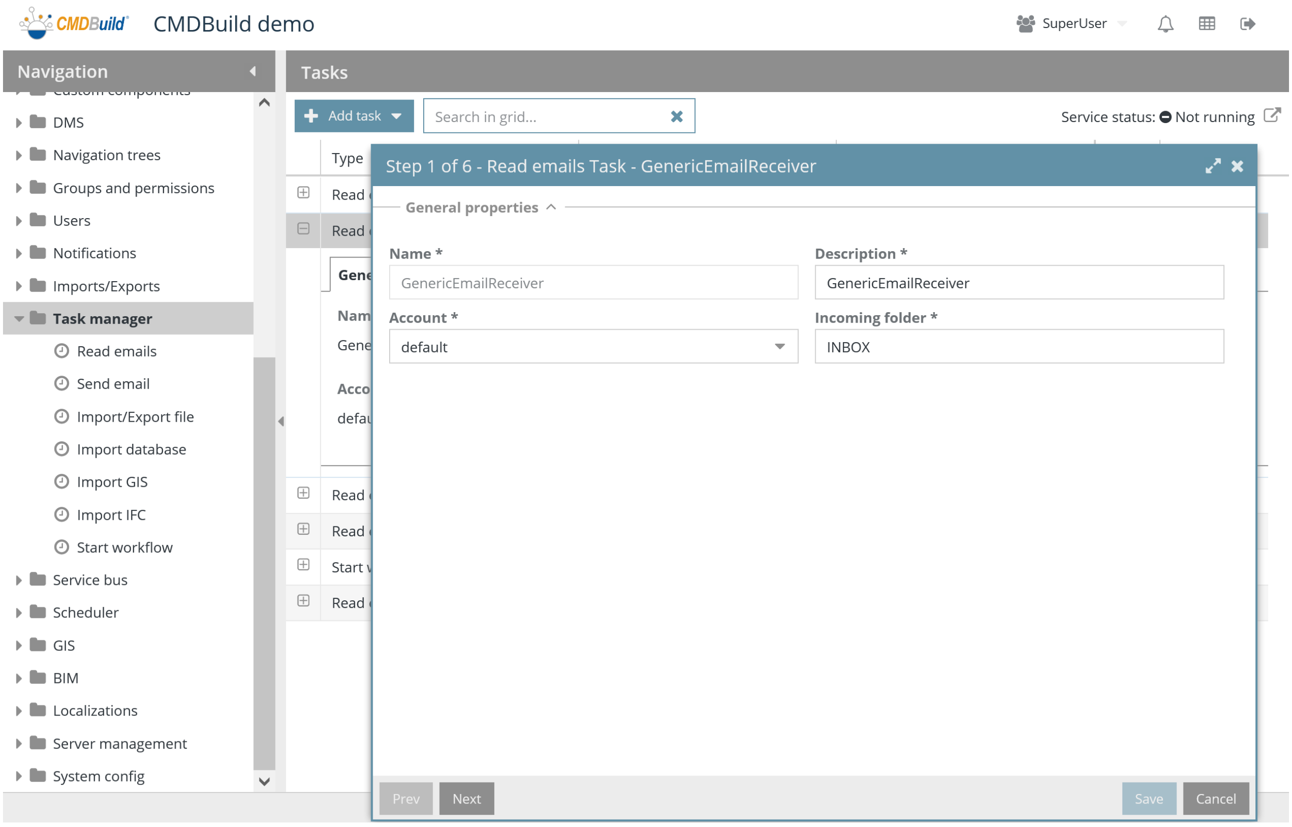
General properties | Settings | Cron | Parsing | Process | Notifications

Name *
GenericEmailReceiver

Description *
GenericEmailReceiver

Account *
default

Incoming folder *
INBOX



Step 1 of 6 - Read emails Task - GenericEmailReceiver

General properties

Name *: GenericEmailReceiver

Description *: GenericEmailReceiver

Account *: default

Incoming folder *: INBOX

For each task typology there is a specific wizard that supports its configuration.
It is described in detail in the following paragraphs.

24.1. Task inbox management

The task for the email incoming waits for the emails to be filed into a specific IMAP folder of a certain account, verifies if it coincides with any prearranged filter criteria and carries out the recommended instructions.

Such activities include also:

- sending notifications
- saving attachments in Alfresco,
- starting workflows

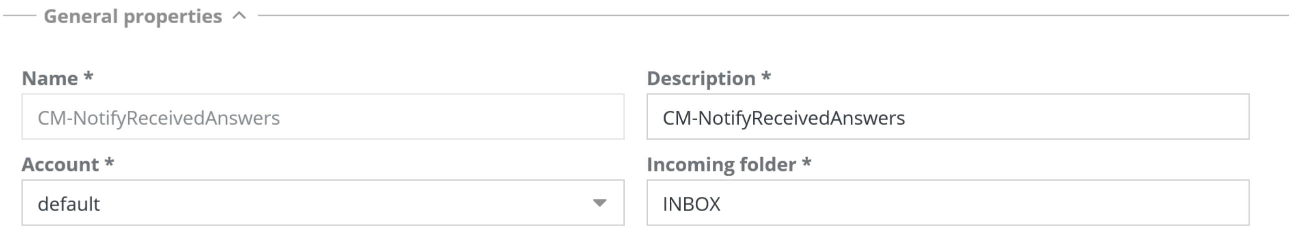
For example you can start this function in order to start Incident Management processes when you receive an e-mail from users, you can also manage any replies from users in a following step of the same workflow, etc.

The wizard includes four pages described below.

24.1.1. Page 1 of the wizard

The following information is required:

- Name: task name
- Description: task description
- Account: post account to control the incoming e-mails (see chapter related to the Email management)
- Inbox: IMAP folder to control the incoming e-mails



General properties

Name *: CM-NotifyReceivedAnswers

Description *: CM-NotifyReceivedAnswers

Account *: default

Incoming folder *: INBOX

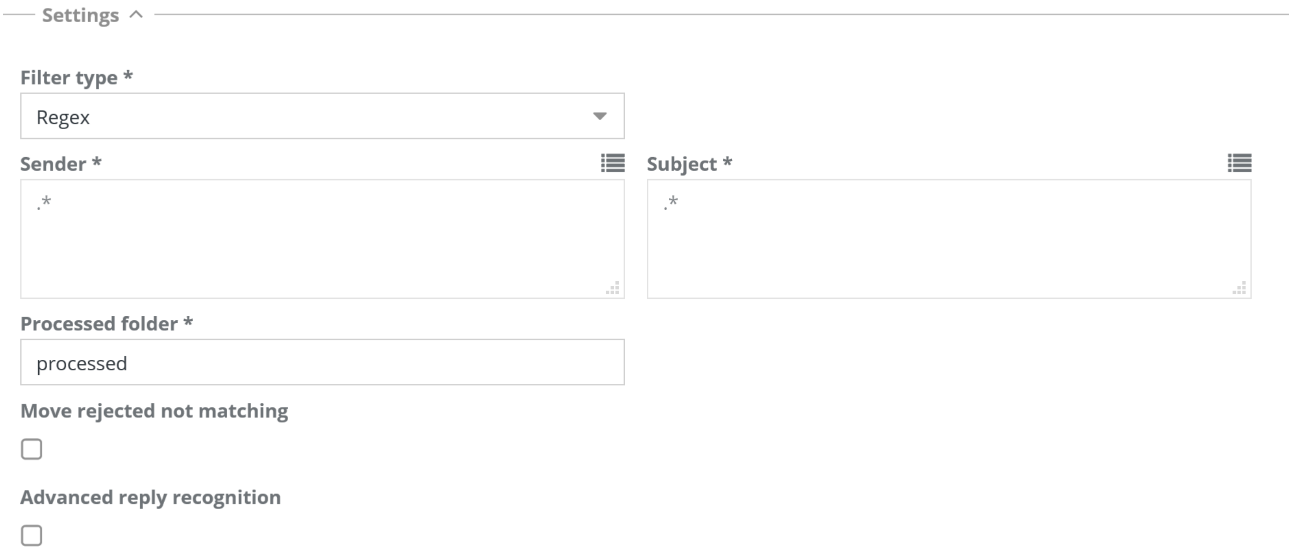
24.1.2. Page 2 of the wizard

The following information is required:

- Type of filter, which can be:
  - None
  - Regex: filter with regular expression, you can assemble it following our guide
  - PostgreSQL function: function defined in the database, with a selection list
  - This is a reply to an email sent from CMDBuild: in order to link it to the card related to the initial mail
  - This is not a reply to an email sent from CMDBuild
- Filter on sender: it includes a possible Regex filter set on the e-mail sender in order to

exclude less interested e-mails; it is expressed as regular expression and can be set through a proper pop-up window

- Filter on the subject: it includes any Regex filter set on the subject of the e-mail in order to exclude less interested e-mail; it is expressed as regular expression and can be set through a proper pop-up window
- Function: it allows to select the PostgreSQL function that implements the filter, in case this option has been chosen
- Processed e-mails: IMAP folder for processed e-mails
- Move any emails that do not satisfy the filter: flag that required the relocation of the emails discarded by the filter
- Discarded e-mails: IMAP folder for discarded e-mails
- Advanced answer recognition: uses "tags" in the header of the mail to obtain useful info and link input email to output ones and to the related data cards



Settings

Filter type *: Regex

Sender *: .*

Subject *: .*

Processed folder *: processed

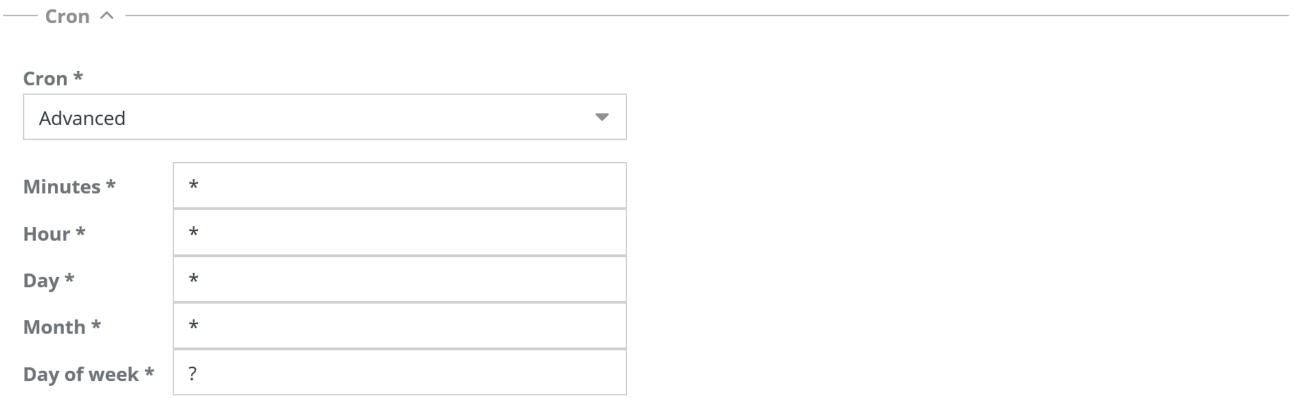
Move rejected not matching: □

Advanced reply recognition: □

24.1.3. Page 3 of the wizard

The information related to the scheduling ways of the task is required:

- Simple mode: with the options “every hour”, “every day”, “every month”, “every year”
- Advanced mode: with the same options and the same syntax as the schedule tool “cron” available in Linux



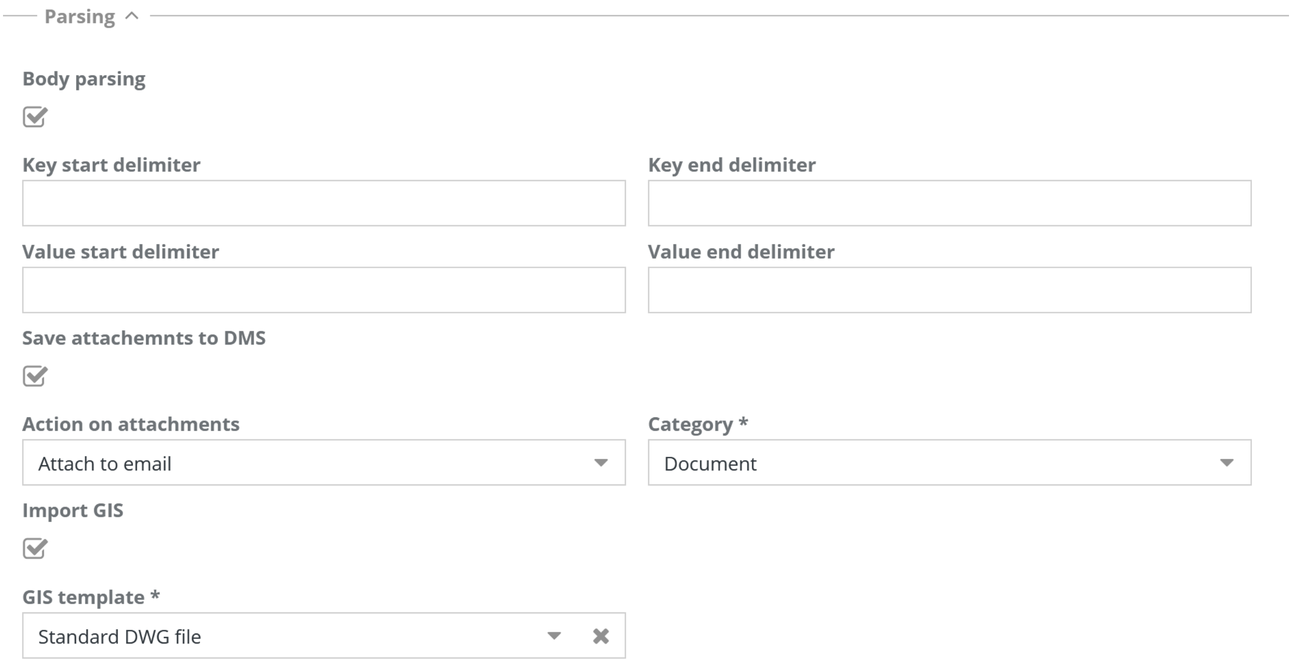
Cron *
Advanced

Minutes *	*
Hour *	*
Day *	*
Month *	*
Day of week *	?

24.1.4. Page 4 of the wizard

The following information is required:

- E-mail body parsing: it enables the email body parsing
- Key start delimiter: it allows to specify the initial delimiter of the possible key field available in the e-mail
- Key final delimiter: it allows to specify the final delimiter of the possible key field available in the e-mail
- Value start delimiter: it allows to specify the initial delimiter value field available in the e-mail
- Value end delimiter: it allows to specify the final delimiter of the value filed available in the e-mail
- Attachment filing into DMS: it enables the attachment filing into DMS
- Action on attachments: any file attached in the email can be linked to the email card in CMDBuild, or to the data card
- Category: it requires to select the category that should be associated with the attachment (all the entries of all the documental categories are available)
- Import GIS attachment: enables importation of the DWG attachment file, if present
- GIS template: allows to select the GIS Template you want to use



Parsing

Body parsing

☑

Key start delimiter

Key end delimiter

Value start delimiter

Value end delimiter

Save attachemnts to DMS

☑

Action on attachments: Attach to email

Category *: Document

Import GIS

☑

GIS template *: Standard DWG file

Note:

The “Body parsing” mechanism allows you to recognize and manage in the workflow any attributes whose value was inserted in the e-mail body, for example:

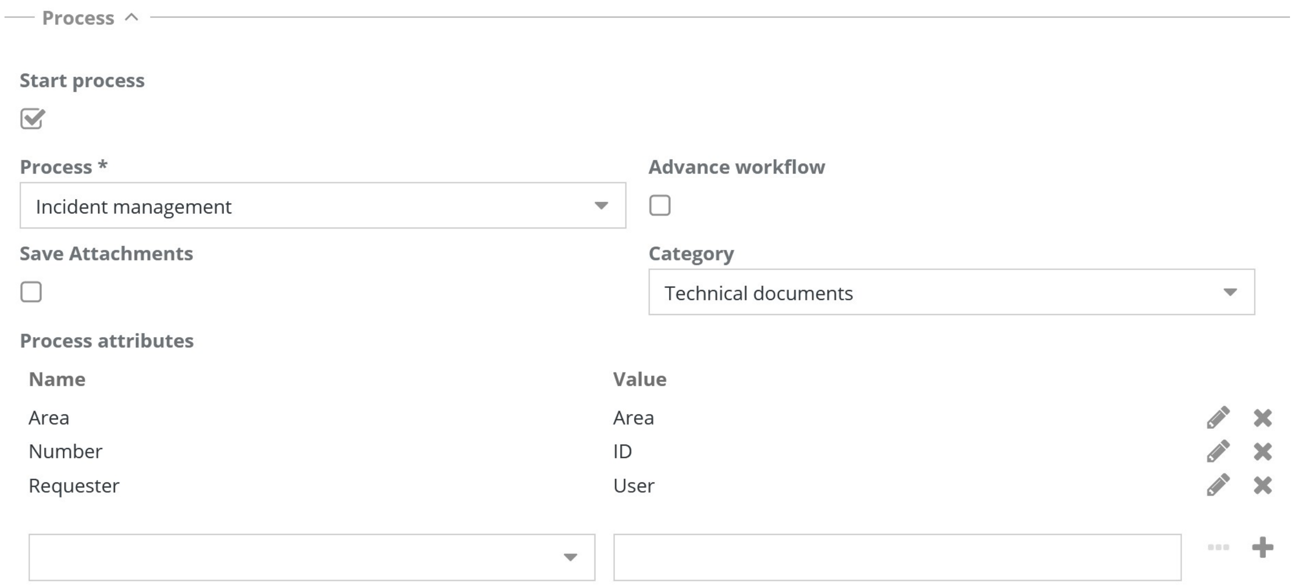
```
Email body

[key]Requester[/key]<value>John Smith</value>
[key]Description[/key]<value>The printer will not turn on</value>
[key]Priority[/key]<value>Medium</value>
```

24.1.5. Page 5 of the wizard

The following information is required:

- Start process: it enables the process starting
- Process: name of the starting process (chosen in the list of available processes)
- Advance workflow: the process can be advanced to the following step
- Attachment filing into DMS: it enables the attachment filing into DMS, linked to the process
- Category: it requires to select the category that should be associated with the attachment (all the entries of all the documental categories are available, if configured for the selected process)
- Mapping of fields: indication of the email fields that have to be reported in the starting process



Process

Start process

☑

Process *

Incident management

Advance workflow

□

Save Attachments

□

Category

Technical documents

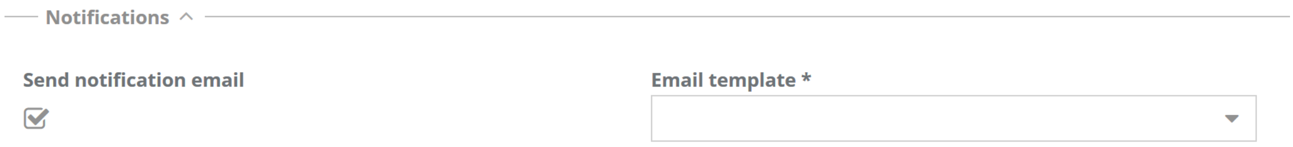
Process attributes

Name	Value
Area	Area
Number	ID
Requester	User

24.1.6. Page 6 of the wizard

The following information is required:

- Send notification email: it enables the notification sending
- Email template: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)



Notifications

Send notification email

☑

Email template *

24.2. Send email task

This function allows you to schedule the automatic and recurring sending of an email, with the option to attach a report.

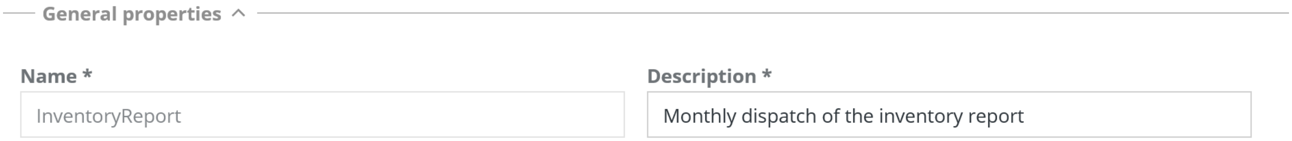
You can predefine the value criteria of the report parameters, obtaining in this way the sending of reports which depend on the time of the task execution.

The wizard includes three pages, described below.

24.2.1. Page 1 of the wizard

The following information is required:

- Name: task name
- Description: task description



General properties

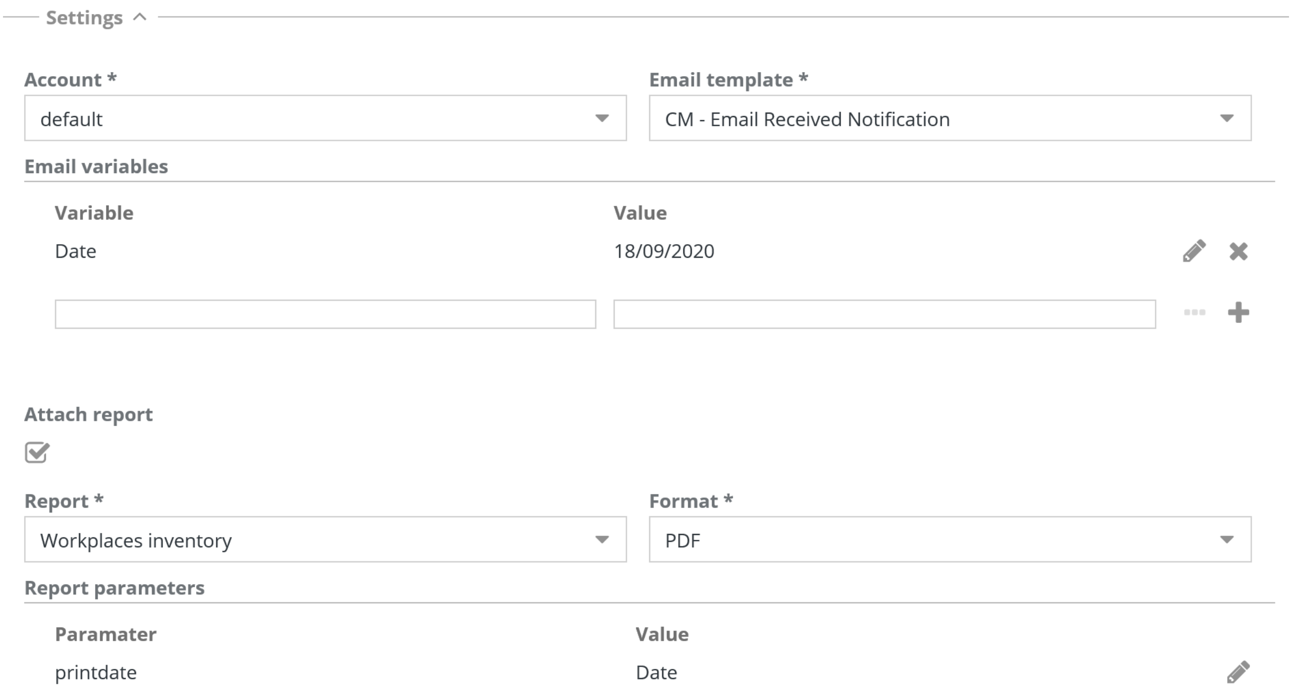
Name *: InventoryReport

Description *: Monthly dispatch of the inventory report

24.2.2. Wizard page 2

The following information is required:

- Account: email recipient
- Template email: you can use this template when you send email (see chapter related to the email management)
- Email variables: you can define context variables, which can be used for the calculation of the report parameters
- Attached report: it enables a report attached to the email
- Report: allows to select the report to send
- Format: allows to specify the format of the report, choosing from the provided ones
- Report parameters: the system presents the list of parameters provided by the selected report and allows compilation



Settings

Account *
default

Email template *
CM - Email Received Notification

Email variables

Variable	Value
Date	18/09/2020

Attach report
☑

Report *
Workplaces inventory

Format *
PDF

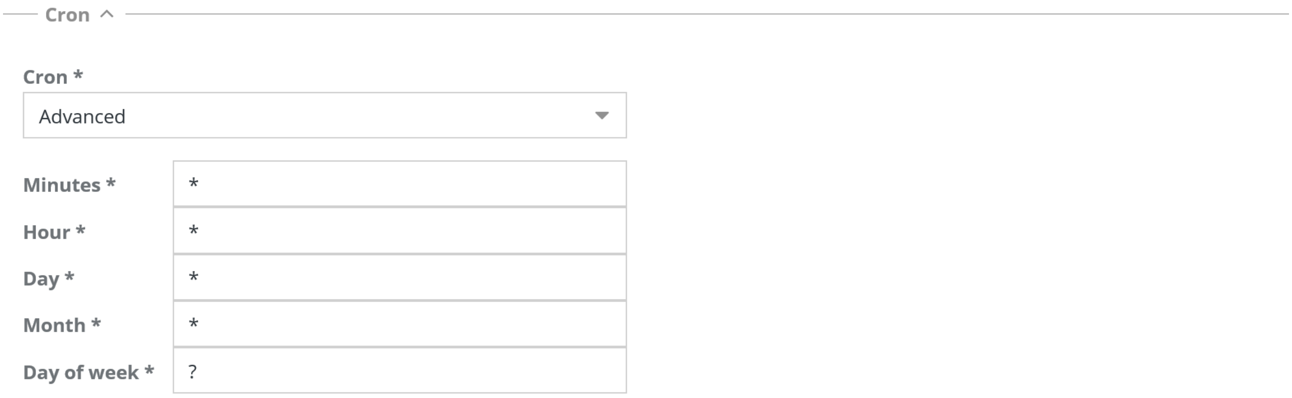
Report parameters

Paramater	Value
printdate	Date

24.2.3. Wizard page 3

The information related to the scheduling ways of the task is required:

- Simple mode: with the options “every hour”, “every day”, “every month”, “every year”
- Advanced mode: with the same options and the same syntax as the schedule tool “cron” available in Linux



Cron

Cron *
Advanced

Minutes *	*
Hour *	*
Day *	*
Month *	*
Day of week *	?

24.3. Task import / export scheduling by file

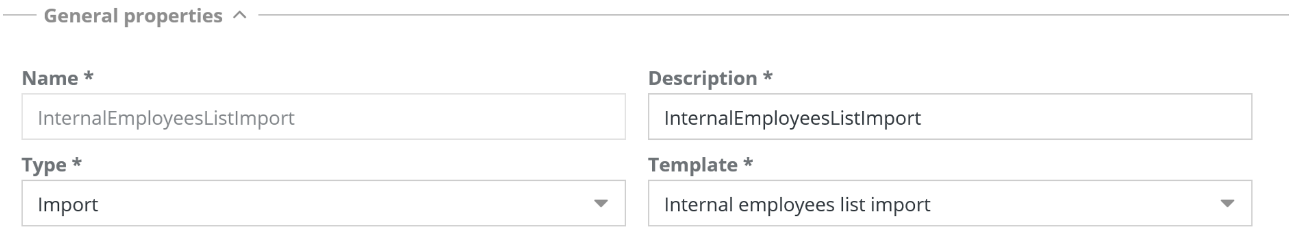
The task allows you to execute data import or export operations in batch mode through file, using specific templates described in the related chapter.

The wizard includes four pages described below.

24.3.1. Page 1 of the wizard

The following information is required:

- Name: task name
- Description: task description
- Type: import or export
- Template: choice of import / export template



General properties

Name *: InternalEmployeesListImport

Description *: InternalEmployeesListImport

Type *: Import

Template *: Internal employees list import

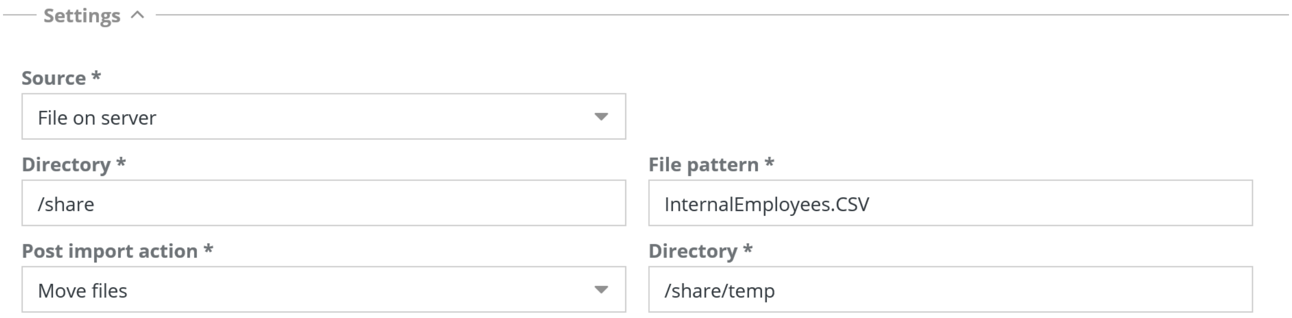
24.3.2. Page 2 of the wizard

The following information is required:

- Source: File in the server or URL

In case the source is a file in the server, the following information is required:

- Directory: server or shared folder including the file
- File pattern: name or pattern for the file recognition
- Post import action: it deletes or disables the file (it is renamed with a predefined pattern) or move the file (in this case the new folder is required) or no action



Settings

Source *: File on server

Directory *: /share

File pattern *: InternalEmployees.CSV

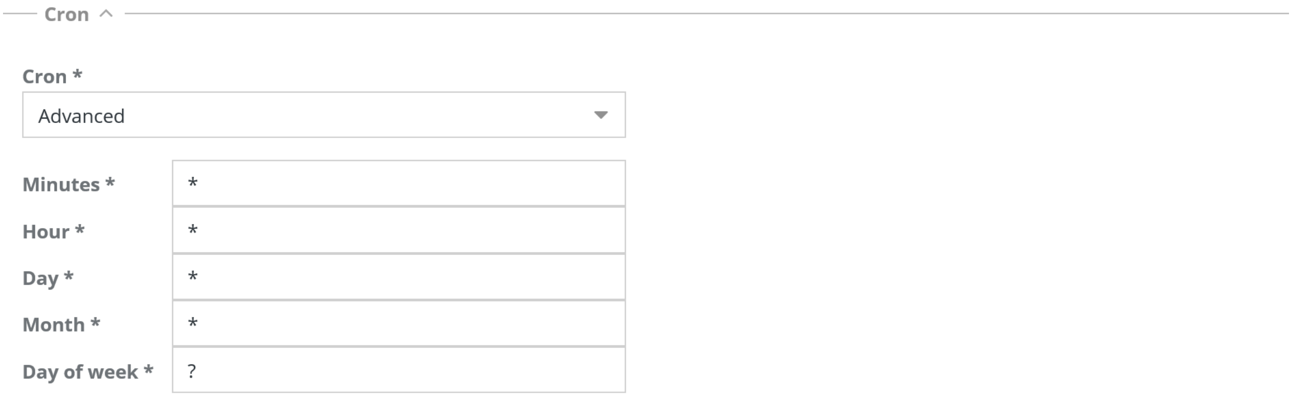
Post import action *: Move files

Directory *: /share/temp

24.3.3. Page 3 of the wizard

The information related to the scheduling ways of the task is required:

- Simple mode: with the options “every hour”, “every day”, “every month”, “every year”
- Advanced mode: with the same options and the same syntax as the schedule tool “cron” available in Linux



Cron *
Advanced

Minutes *	*
Hour *	*
Day *	*
Month *	*
Day of week *	?

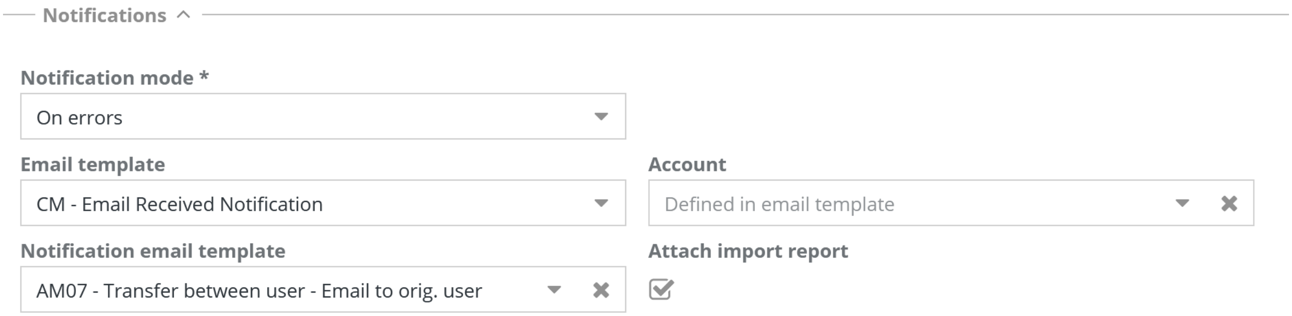
24.3.4. Page 4 of the wizard

The following information is required:

- Notification mode: in case of errors or always or never

In the first two cases the following information is required:

- Execution email template: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Template email of error notification: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Account: it requires to select one of the accounts preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Attach report: requests to attach the report, actions and possible errors to the mail



Notification mode *
On errors

Email template
CM - Email Received Notification

Account
Defined in email template

Notification email template
AM07 - Transfer between user - Email to orig. user

Attach import report
☑

24.4. Task import scheduling by database charts

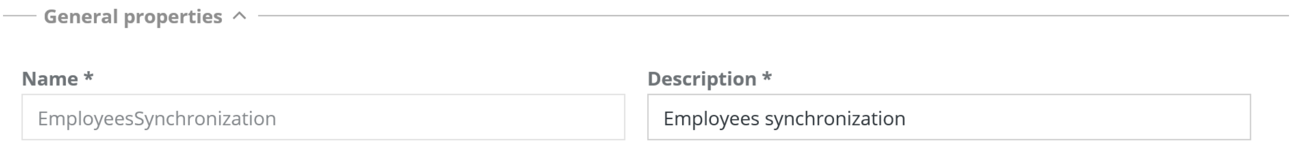
The task allows you to execute data import or export operations in batch mode from relation database charts, using specific database templates described in the related chapter of this manual.

The wizard includes four pages described below.

24.4.1. Page 1 of the wizard

The following information is required:

- Name: task name
- Description: task description


General properties

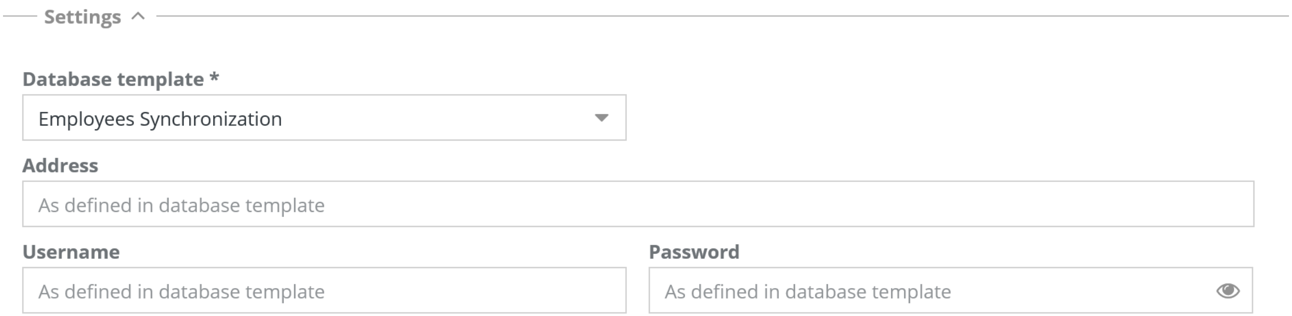
Name *
EmployeesSynchronization

Description *
Employees synchronization

24.4.2. Page 2 of the wizard

The following information is required:

- Database template: choice of the import database template to apply
- Address: connection string to the external database (the one set in the template can be kept as default)
- Username: access username to the external database (the one set in the template can be kept as default)
- Password: access password to the external database (the one set in the template can be kept as default)


Settings

Database template *
Employees Synchronization

Address
As defined in database template

Username
As defined in database template

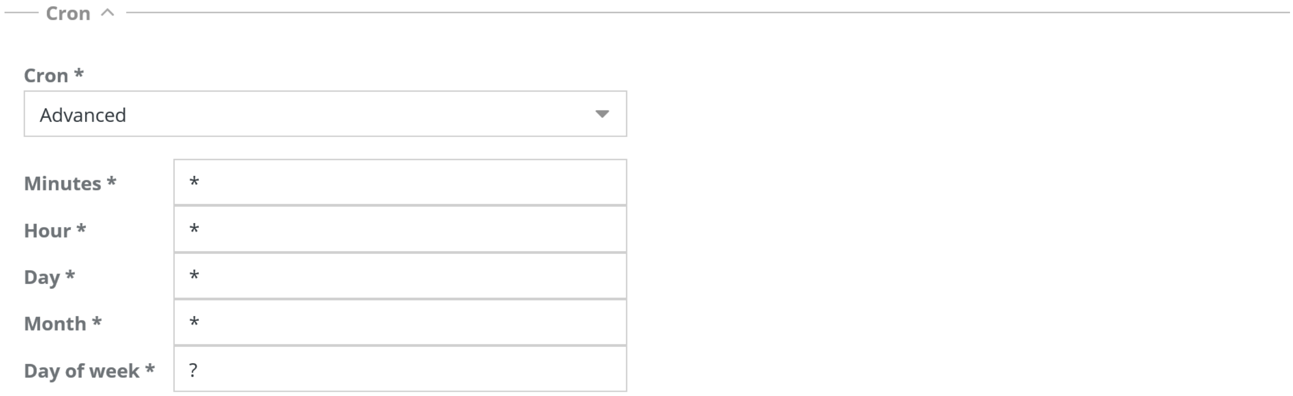
Password
As defined in database template

24.4.3. Wizard page 3

The information related to the scheduling ways of the task is required:

- Simple mode: with the options “every hour”, “every day”, “every month”, “every year”
- Advanced mode: with the same options and the same syntax as the schedule tool “cron”

available in Linux



Cron

Cron *: Advanced

Minutes *: *

Hour *: *

Day *: *

Month *: *

Day of week *: ?

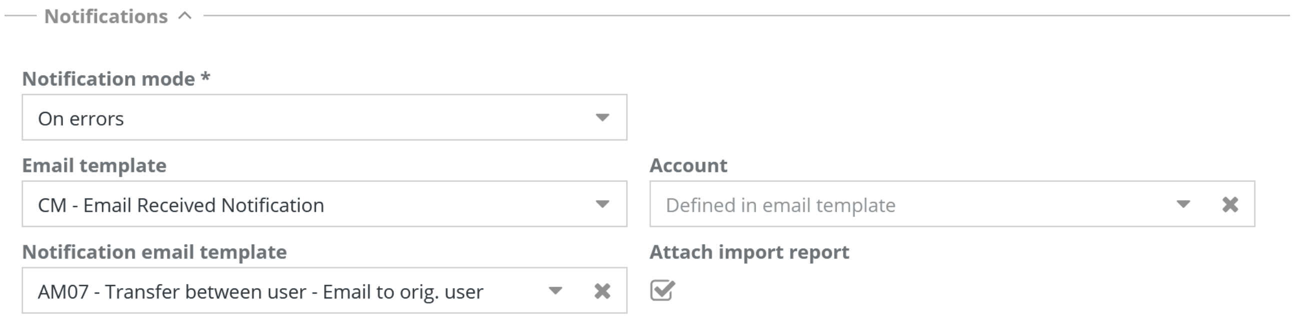
24.4.4. Wizard page 4

The following information is required:

- Notification mode: in case of errors or always or never

In the first two cases the following information is required:

- Execution email template: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Template email of error notification: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Account: it requires to select one of the accounts preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Attach report: requests to attach the report, actions and possible errors to the mail



Notifications

Notification mode *: On errors

Email template: CM - Email Received Notification

Account: Defined in email template

Notification email template: AM07 - Transfer between user - Email to orig. user

Attach import report: ☑

24.5. Task import scheduling by GIS planimetries

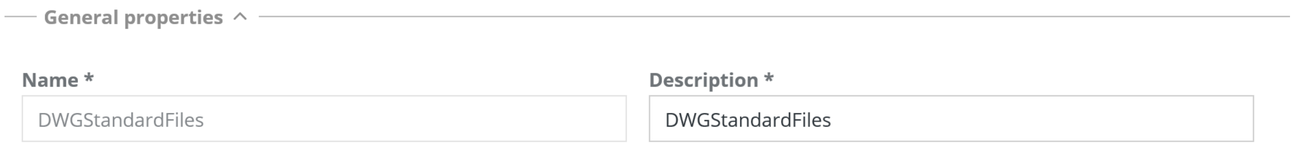
The task allows you to execute data import or export operations in batch mode from GIS planimetries, using specific database templates described in the related chapter of this manual.

The wizard includes four pages described below.

24.5.1. Page 1 of the wizard

The following information is required:

- Name: task name
- Description: task description



General properties ^

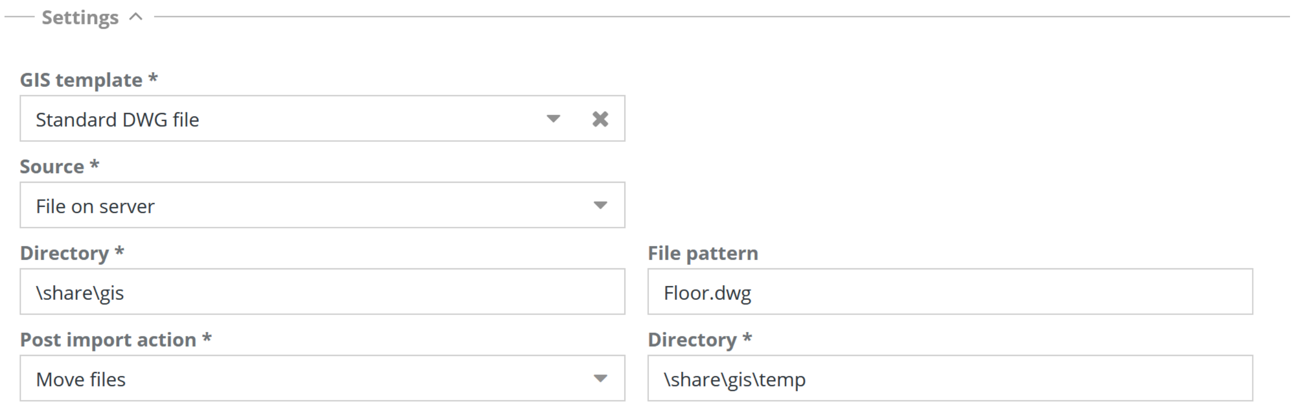
Name *: DWGStandardFiles

Description *: DWGStandardFiles

24.5.2. Page 2 of the wizard

The following information is required:

- GIS template: choice of the GIS import template to apply
- Source: File in the server or URL
- Directory: server folder where you can find the DWG file to import
- File pattern: name or pattern for the DWG file recognition
- Post import action: it deletes or disables the file (it is renamed with a predefined pattern) or move the file (in this case the new folder is required) or no action



Settings ^

GIS template *: Standard DWG file

Source *: File on server

Directory *: \share\gis

File pattern: Floor.dwg

Post import action *: Move files

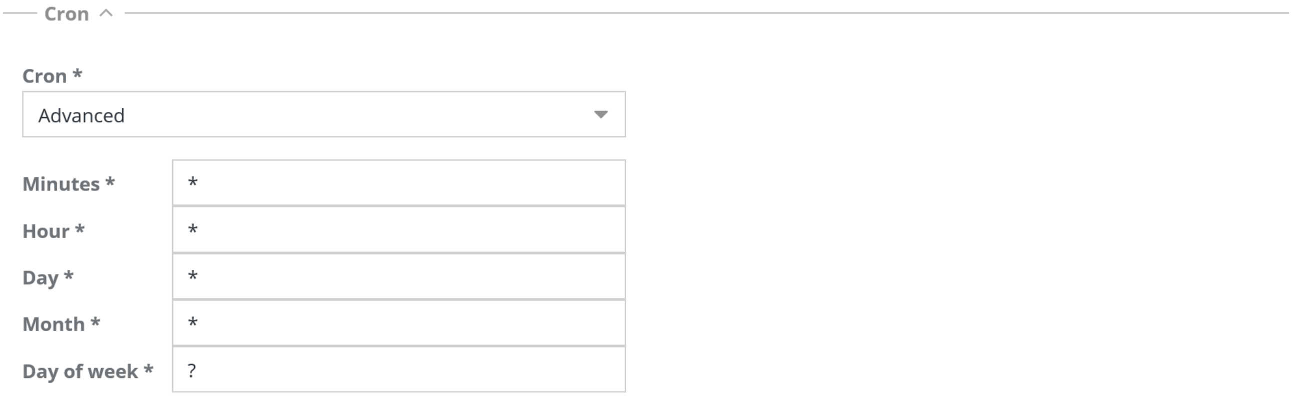
Directory *: \share\gis\temp

24.5.3. Wizard page 3

The information related to the scheduling ways of the task is required:

- Simple mode: with the options “every hour”, “every day”, “every month”, “every year”

- Advanced mode: with the same options and the same syntax as the schedule tool “cron” available in Linux



Cron

Cron *: Advanced

Minutes *: *

Hour *: *

Day *: *

Month *: *

Day of week *: ?

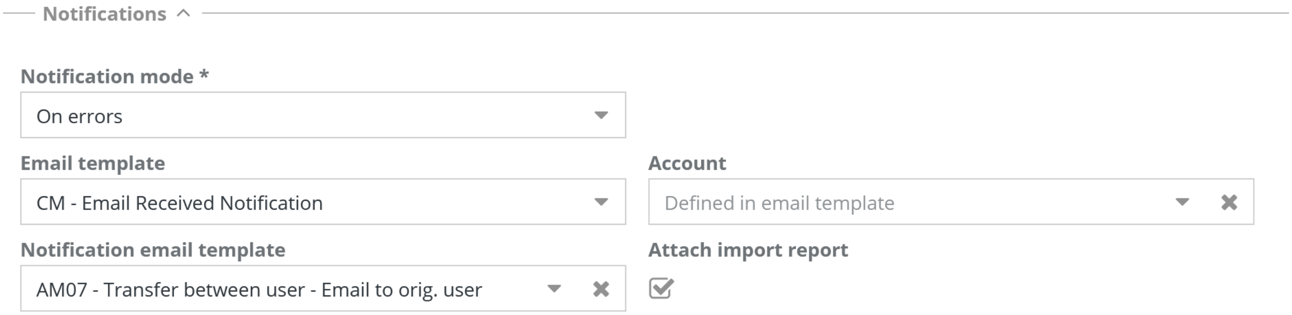
24.5.4. Wizard page 4

The following information is required:

- Notification mode: in case of errors or always or never

In the first two cases the following information is required:

- Execution email template: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Template email of error notification: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Account: it requires to select one of the accounts preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Attach report: requests to attach the report, actions and possible errors to the mail



Notifications

Notification mode *: On errors

Email template: CM - Email Received Notification

Account: Defined in email template

Notification email template: AM07 - Transfer between user - Email to orig. user

Attach import report: ☑

24.6. Task import scheduling by IFC file (BIM models)

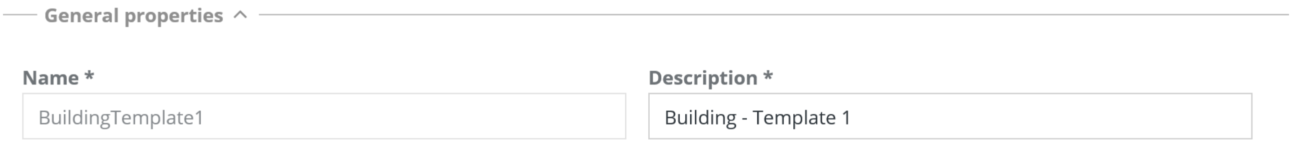
The task allows you to execute data import or export operations in batch mode from IFC exchange files (format to export 3D BIM models), using specific database templates described in the related chapter of this manual.

The wizard includes four pages described below.

24.6.1. Page 1 of the wizard

The following information is required:

- Name: task name
- Description: task description



Name *
BuildingTemplate1

Description *
Building - Template 1

24.6.2. Page 2 of the wizard

The following information is required:

- BIM template: choice of the import template to apply
- Import from: provides "File" or "Project" options

In case you import from a file, the following information is required:

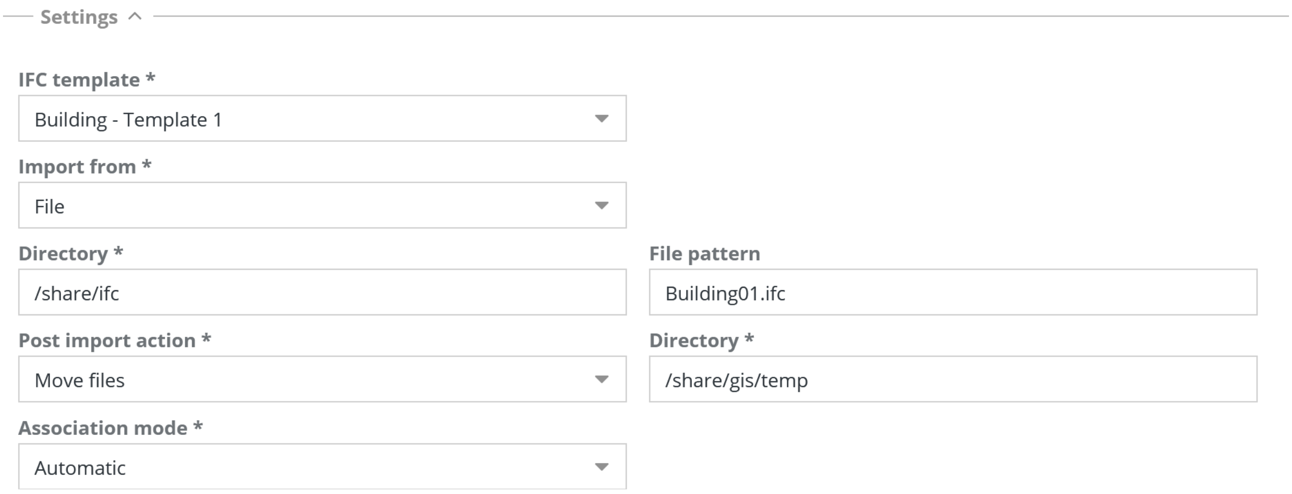
- Directory: server or shared folder including the file
- File pattern: name or pattern for the file recognition
- Post import action: it deletes or disables the file (it is renamed with a predefined pattern) or move the file (in this case the new folder is required) or no action

In case you import from a project, the following information is required:

- Project name: for a description of BIM

In both cases you need:

- Association mode: automatic or static



Settings

IFC template *: Building - Template 1

Import from *: File

Directory *: /share/ifc

File pattern: Building01.ifc

Post import action *: Move files

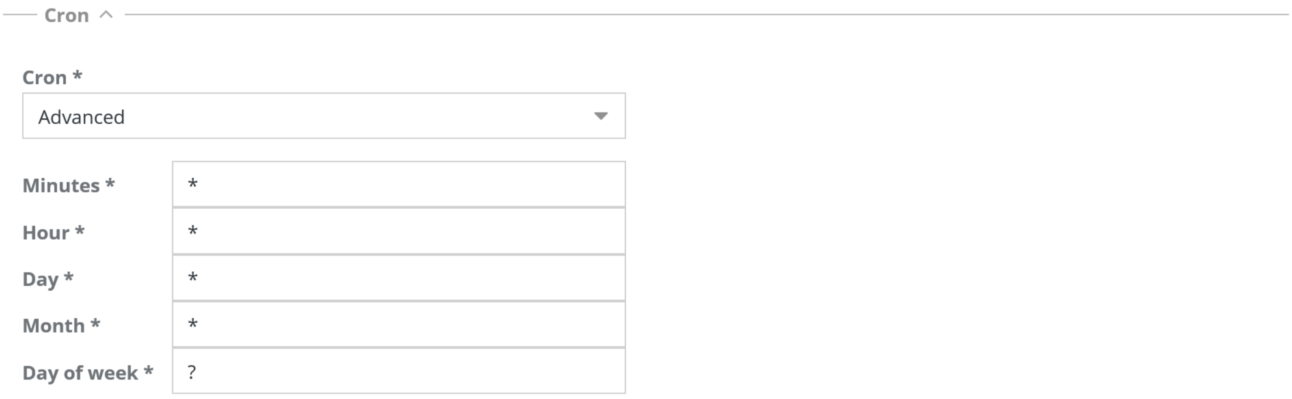
Directory *: /share/gis/temp

Association mode *: Automatic

24.6.3. Wizard page 3

The information related to the scheduling ways of the task is required:

- Simple mode: with the options “every hour”, “every day”, “every month”, “every year”
- Advanced mode: with the same options and the same syntax as the schedule tool “cron” available in Linux



Cron

Cron *: Advanced

Minutes *: *

Hour *: *

Day *: *

Month *: *

Day of week *: ?

24.6.4. Wizard page 4

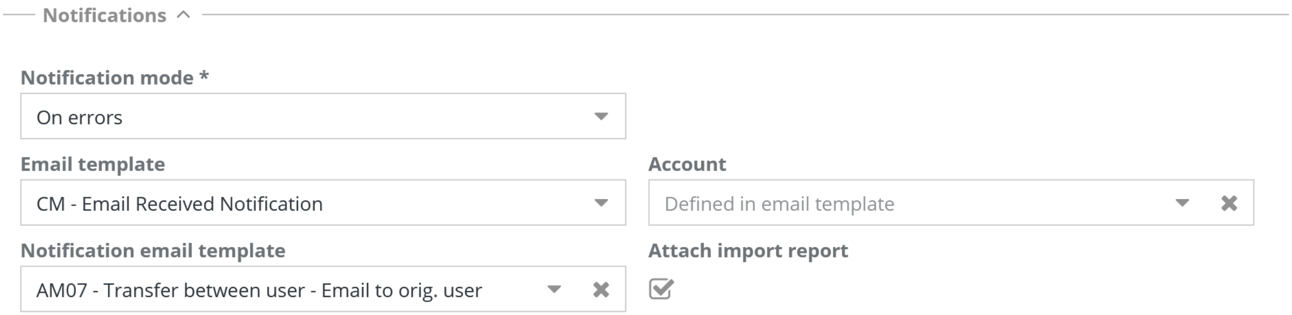
The following information is required:

- Notification mode: in case of errors or always or never

In the first two cases the following information is required:

- Execution email template: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Template email of error notification: it requires to select one of the templates preconfigured with the proper function (see chapter relate to the e-mail management menu)

- Account: it requires to select one of the accounts preconfigured with the proper function (see chapter relate to the e-mail management menu)
- Attach report: requests to attach the report, actions and possible errors to the mail



Notifications

Notification mode *: On errors

Email template: CM - Email Received Notification

Account: Defined in email template

Notification email template: AM07 - Transfer between user - Email to orig. user

Attach import report: ☑

24.7. Scheduling task for starting workflow

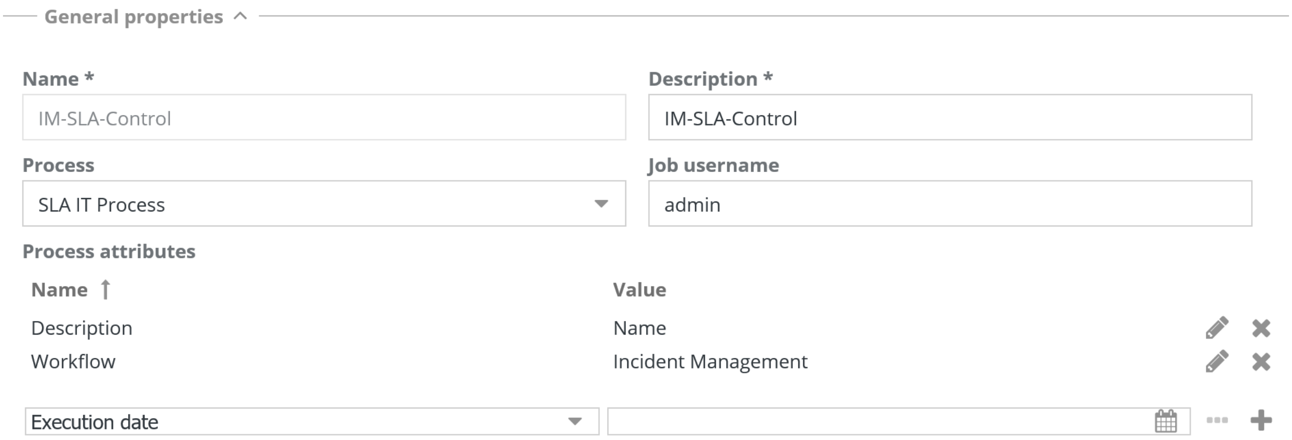
This task allows you to schedule a workflow configured in CMDBuild.

The wizard includes two pages described below.

24.7.1. Page 1 of the wizard

The following information is required:

- Name: task name
- Description: task description
- Process: starting workflow
- Job username: to be used to start the workflow (from which permissions derive, according to the predefined group)
- Process attributes: list of attributes and values to initialize some starting process attributes



General properties

Name *	Description *
IM-SLA-Control	IM-SLA-Control
Process	Job username
SLA IT Process	admin

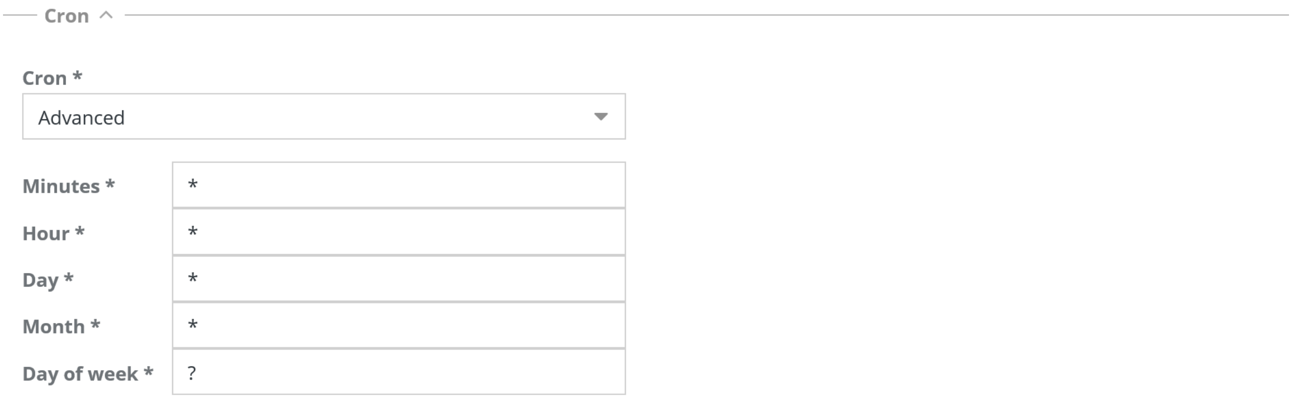
Process attributes

Name	Value
Description	Name
Workflow	Incident Management
Execution date	

24.7.2. Page 2 of the wizard

The information related to the scheduling ways of the task is required:

- Simple mode: with the options “every hour”, “every day”, “every month”, “every year”
- Advanced mode: with the same options and the same syntax as the schedule tool “cron” available in Linux



Cron

Cron *	Advanced
Minutes *	*
Hour *	*
Day *	*
Month *	*
Day of week *	?

25. Service bus

CMDBuild is a web enterprise system, implemented in SOA architecture (Service Oriented Architecture) and designed to grant interoperability solutions.

In the most efficient implementations, an architecture like this is supported by a BUS Service (or ESB, Enterprise Service Bus), a software infrastructure (middleware) for the interoperability among applications and systems through standard services for coordination, security, messaging, routing and transformations.

The CMDBuild Service Bus includes:

- management of data flows and messages (receipt, routing, elaboration, sending)
- distributed management of the charge related to some batch processes
- input/output modules with features of coordination among data sending / receiving / sync processes
- error and notification management
- “webhook” features for the subscription of a notification service in case of certain events in the CMDB
- supporting the “chat” service for the communication among CMDBuild users

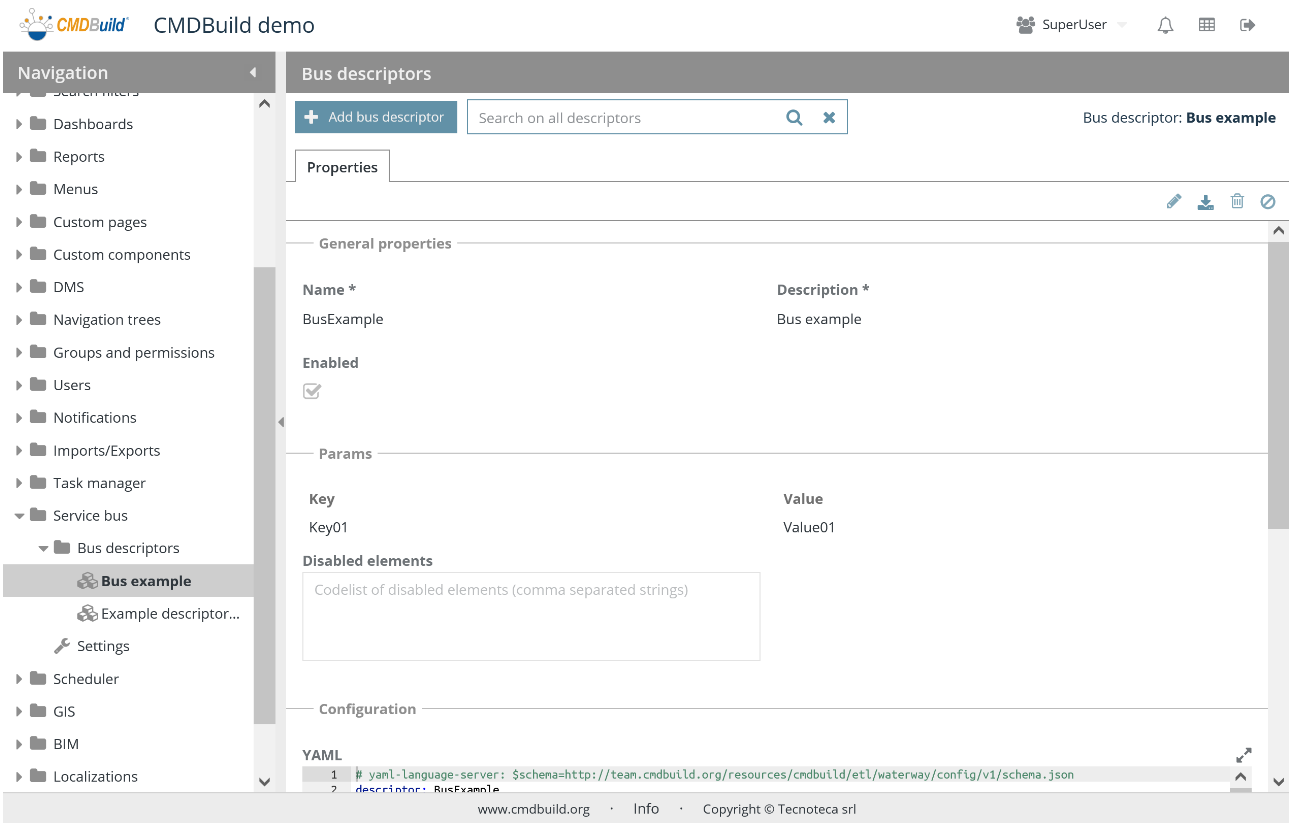
The CMDBuild Service BUS is designed with a plugin architecture, where standard basic components available for everybody, standard advanced components supplied with the subscription and custom components can coexist.

25.1. Bus descriptor

CMDBuild Service Bus behavior is defined through objects, called Bus descriptors, that describes data flow and the transformation and elaboration logic.

Each descriptor includes:

- a schema that describes components corresponding to execution steps and other resources
- a YAML script including custom code and/or calls to CMDBuild API or other services



Bus descriptor: **Bus example**

General properties

Name *: BusExample

Description *: Bus example

Enabled: ☑

Params

Key	Value
Key01	Value01

Disabled elements: Codelist of disabled elements (comma separated strings)

Configuration

YAML

```
# yaml-language-server: $schema=http://team.cmdbuild.org/resources/cmdbuild/etl/waterway/config/v1/schema.json
descriptor: BusExample
```

There is the possibility of performing the following operations:

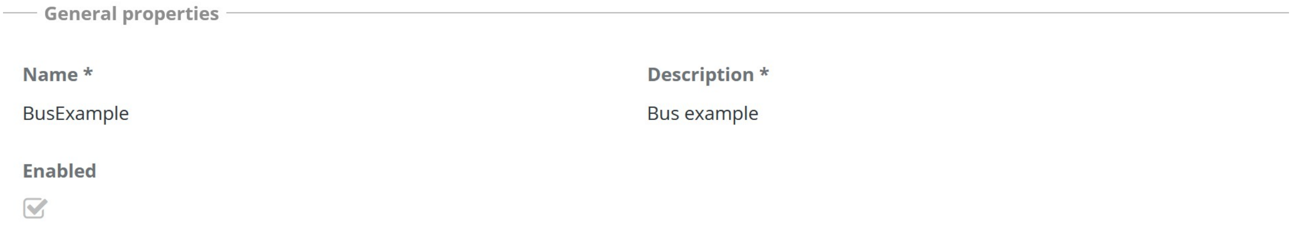
- Add a new rule of deadline generation	✚ Add bus descriptor
- Do searches among inserted rules	Search on all descriptors 🔍 ✖
- Edit the selected bus descriptor	✎
- Delete the selected bus descriptor	🗑
- Disable the selected bus descriptor (i.e. set the status as non-active)	⊘

Each bus descriptor needs some parameters to be compiled (metadata), as mentioned.

25.1.1. General properties

The following information is required:

- Name: bus descriptor name
- Description: bus descriptor description
- Active: it states whether the bus descriptor is active and can be used



General properties

Name *	**Description ***
BusExample	Bus example

Enabled

☑

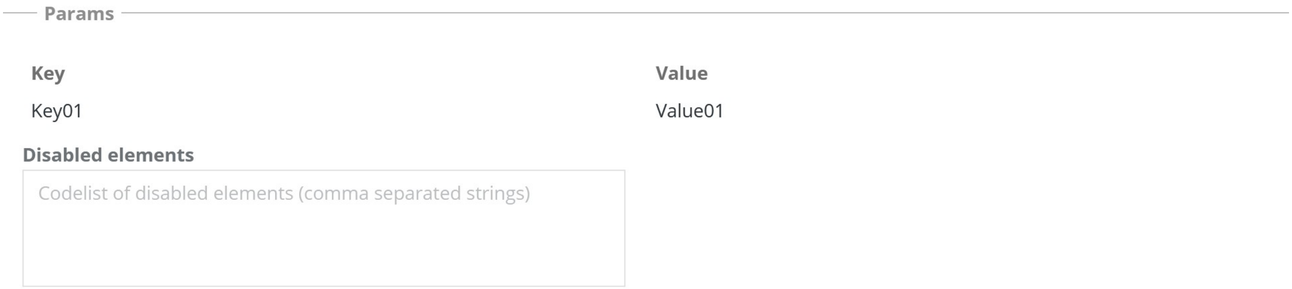
25.1.2. Parameters

You can specify a list of parameters as follows:

- Key: parameter key given to the bus descriptor (in order to parameterize its behavior)
- Value: value given to the Bus descriptor

The following additional parameter is also required:

- Disabled elements: you are required to state any components of the Bus descriptor you want to disable in order to debug or for other needs



Params

Key	Value
Key01	Value01

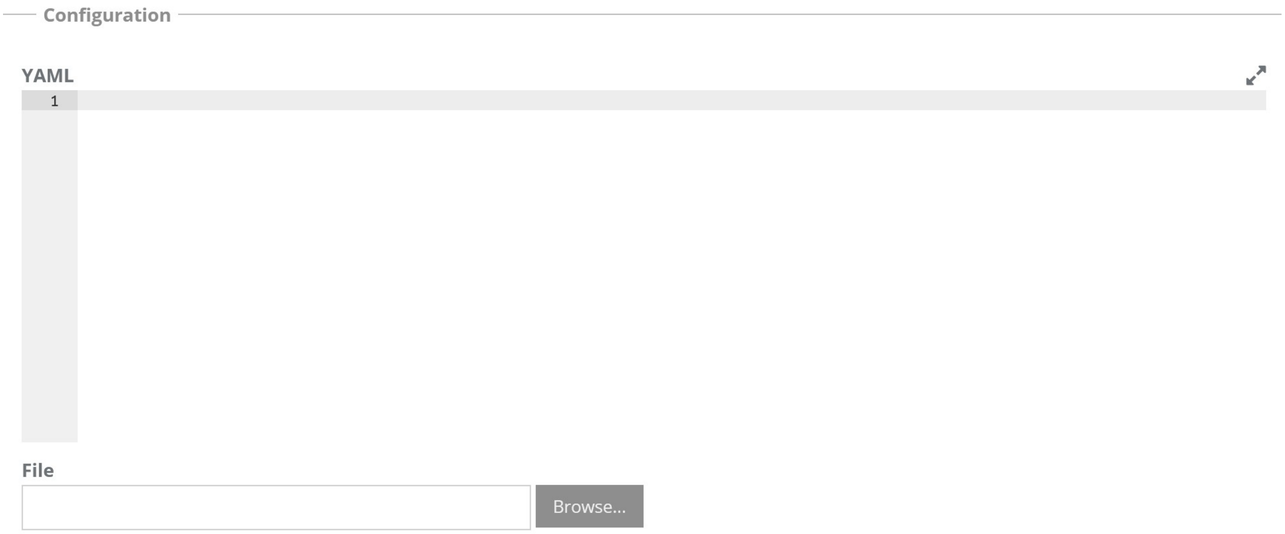
Disabled elements

Codelist of disabled elements (comma separated strings)

25.1.3. Configuration

The following information is required:

- YAML code: Bus descriptor body, YAML format
- File: file including the Bus descriptor body, in YAML format, an alternative to the previous one

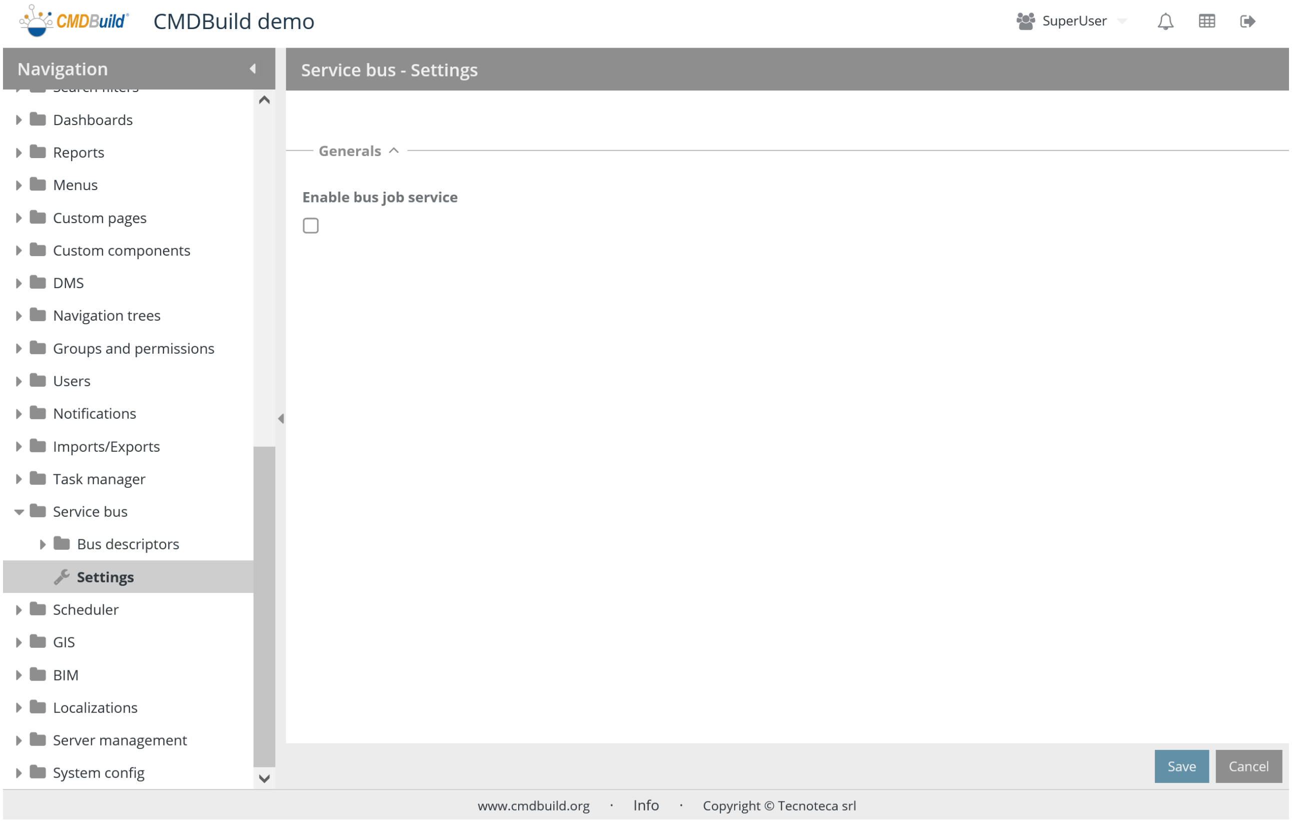


Configuration
YAML
1
File
Browse...

25.2. Configurations

The function allows to set all configuration parameters of the CMDBuild Service Bus.

At the moment you can only enable or disable the Service Bus.



Enable bus job service

□

26. Schedule

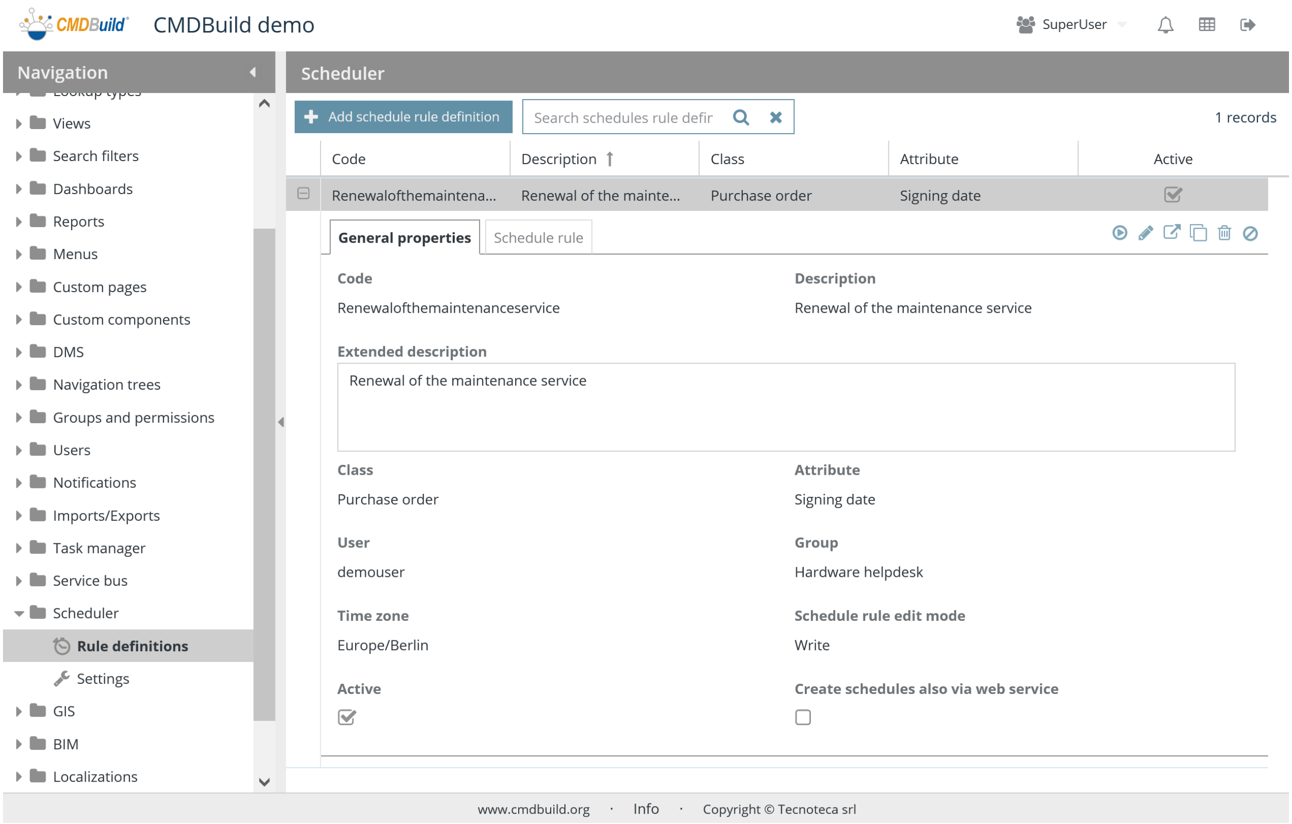
The scheduling helps operators to check procedures related to, for example, contracts with customers and suppliers, orders, certifications, warranties, recurring operations on assets, etc.

Deadlines can be generated automatically from the system when inserting new data cards including data attributes marked as "deadline", but also can be inserted manually into the resulting Scheduling file.

In the first case the Administration Module allows you to define generation criteria associated to any type of "Data" attribute of any class, specifying the ones that can be displayed or edited from operators.

26.1. Rule definition

This function allows you to create new rules for the generation of deadlines and to edit the properties of the old ones.



Code	Description	Class	Attribute	Active
Renewalofthemaintena...	Renewal of the mainte...	Purchase order	Signing date	☑

Code: Renewalofthemaintenanceservice

Description: Renewal of the maintenance service

Extended description: Renewal of the maintenance service

Class: Purchase order

Attribute: Signing date

User: demouser

Group: Hardware helpdesk

Time zone: Europe/Berlin

Schedule rule edit mode: Write

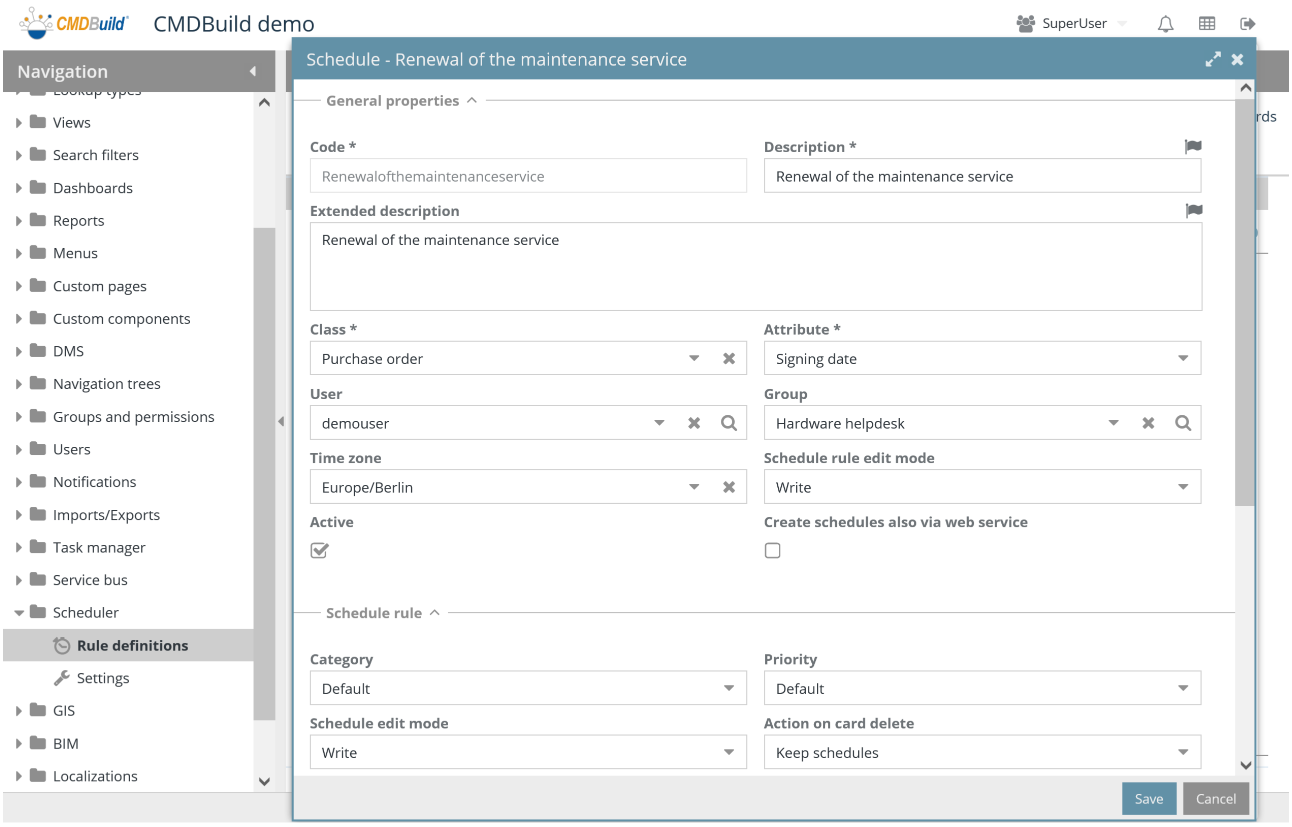
Active: ☑

Create schedules also via web service: ☐

There is the possibility of performing the following operations:

• Add a new rule of deadline generation	Add schedule rule definition
• Do searches among inserted rules	Search schedules rule defir

- Force the rule application to the cards already inserted	⏵
- Edit the selected rule	✎
- Display the complete information related to the selected rule	↗
- Delete the selected rule	🗑
- Disable the selected rule (i.e. set the status as non-active)	⊘



Schedule - Renewal of the maintenance service

General properties

Code *: Renewalofthemaintenanceservice
Description *: Renewal of the maintenance service
Extended description: Renewal of the maintenance service
Class *: Purchase order
Attribute *: Signing date
User: demouser
Group: Hardware helpdesk
Time zone: Europe/Berlin
Schedule rule edit mode: Write
Active: ☑
Create schedules also via web service: ☐

Schedule rule

Category: Default
Priority: Default
Schedule edit mode: Write
Action on card delete: Keep schedules

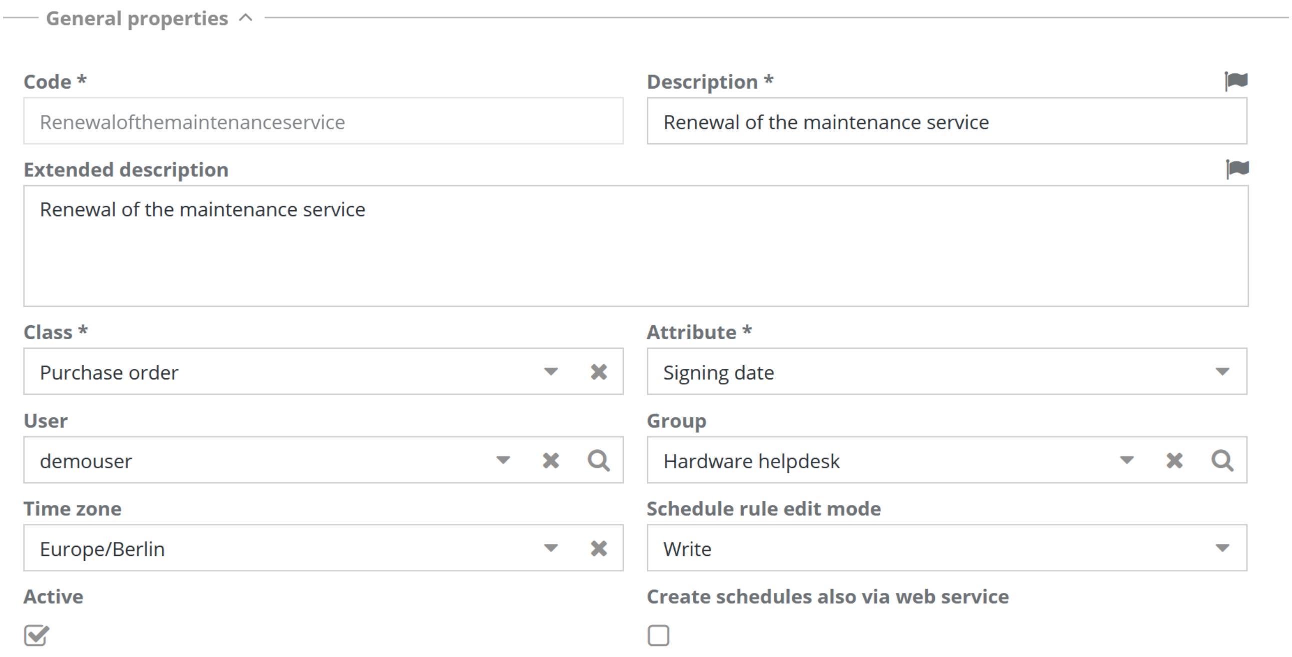
Every rule of generation of deadlines needs certain parameters to be compiled (metadata), as described below.

26.1.1. General properties

The following information is required:

- Code
- Description
- Description
- Class: this class contains Data attributes to be marked as "Expiration"

- Attribute: this attribute generates the expiration according to defined rules
- User: in case it is specified, the expiration visibility will be limited to that user
- Group: in case it is specified, the expiration visibility will be limited to that group
- Time zone: it specifies the time zone considered in the generation and notification of the expiration(s)
- Schedule rule edit mode, for operators that insert into the system data cards that generate expirations:
    - Hidden: the expiration will be generated without the operator's intervention
    - Read: the operator can only view some parameters that generate the expiration
    - Write: the operator can also edit some parameters that generate the expiration
- Active: states if the rule is active
- Creates deadlines also via webservice: automatically executed also operating via webservice



General properties

Code *: Renewalofthemaintenanceservice

Description *: Renewal of the maintenance service

Extended description: Renewal of the maintenance service

Class *: Purchase order

Attribute *: Signing date

User: demouser

Group: Hardware helpdesk

Time zone: Europe/Berlin

Schedule rule edit mode: Write

Active: ☑

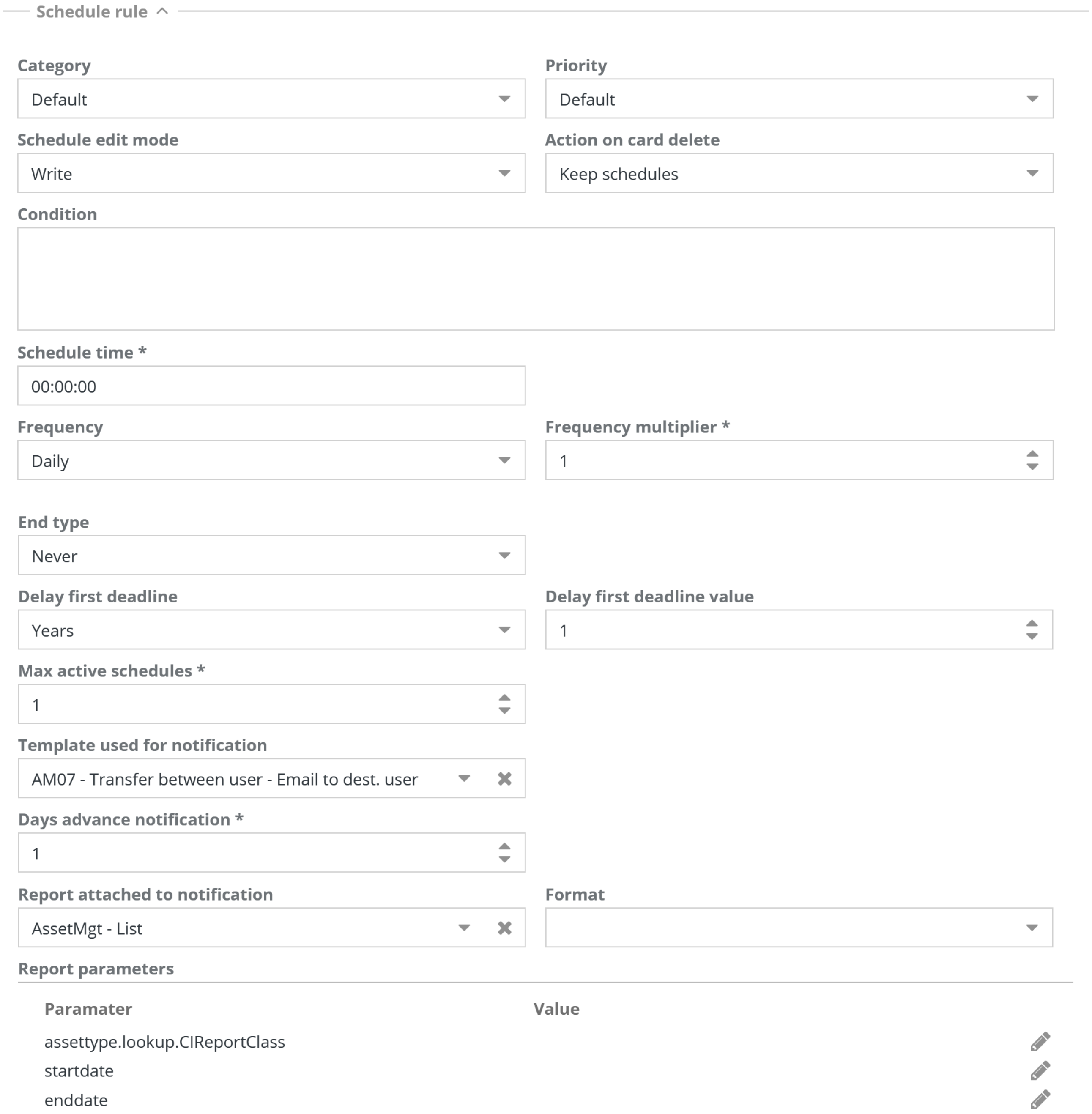
Create schedules also via web service: □

26.1.2. Schedule rule

The following information is required:

- Category: the expiration is classified with the specified category (the values can be customized through a specific lookup list)
- Priority: the expiration is classified with the specified priority (the values can be customized through a specific lookup list)
- Schedule edit mode, during the insertion / change of a data card that generated expirations:
    - Hidden: the expirations during the generation will not be shown to the operator

  - Read: the expirations during the generation will be shown to the operation in only-read mode
  - Write: the expirations during the generation can be changed by the operator
- Action on card delete:
  - Unlink: it unlinks the expiration from the linked data card
  - Delete: it deletes the expiration
- Condition: the expiration will be generated only in case the specified condition is verified
- Schedule time: you can specify a time in the expiration date
- Frequency, which takes the following values:
  - Once
  - Daily
  - Weekly
  - Monthly
  - Yearly
- Frequency multiplier: it allows you to specify a frequency multiplier, for example, every three months, every two years, etc.
- End type, which takes the following values:
  - Never
  - End date, in case it appears, you have to valorize the "End date" field
  - Number of events, in case it appears, you have to valorize the "Number of occurrences" field
- Daily before deadline, which can be specified in:
  - Years
  - Months
  - Days
- Daily before deadline value: number of days / months / years of the postponement
- Max active schedules: it allows you to limit the number of schedules which are open at the same time in the scheduling (once an expiration is closed, the next one will be automatically generated)
- Notification template: email template to use for notification
- Days advance notification: it allows you to specify the expiration notification (it appears only after having chosen the notification template)
- Attached notification report: it allows to attach a report to the notification mail
- Format: report format (PDF or CSV)
- Report parameters: defining of the parameters provided by the report



Schedule rule

Category: Default

Priority: Default

Schedule edit mode: Write

Action on card delete: Keep schedules

Condition:

Schedule time *: 00:00:00

Frequency: Daily

Frequency multiplier *: 1

End type: Never

Delay first deadline: Years

Delay first deadline value: 1

Max active schedules *: 1

Template used for notification: AM07 - Transfer between user - Email to dest. user

Days advance notification *: 1

Report attached to notification: AssetMgt - List

Format:

Report parameters

Paramater	Value
assettype.lookup.CIReportClass	
startdate	
enddate	

26.2. Configurations

The function allows to set all configuration parameters necessary to the scheduling file:

- general properties
- list of values
- list of categories
- list of priorities

26.2.1. General properties

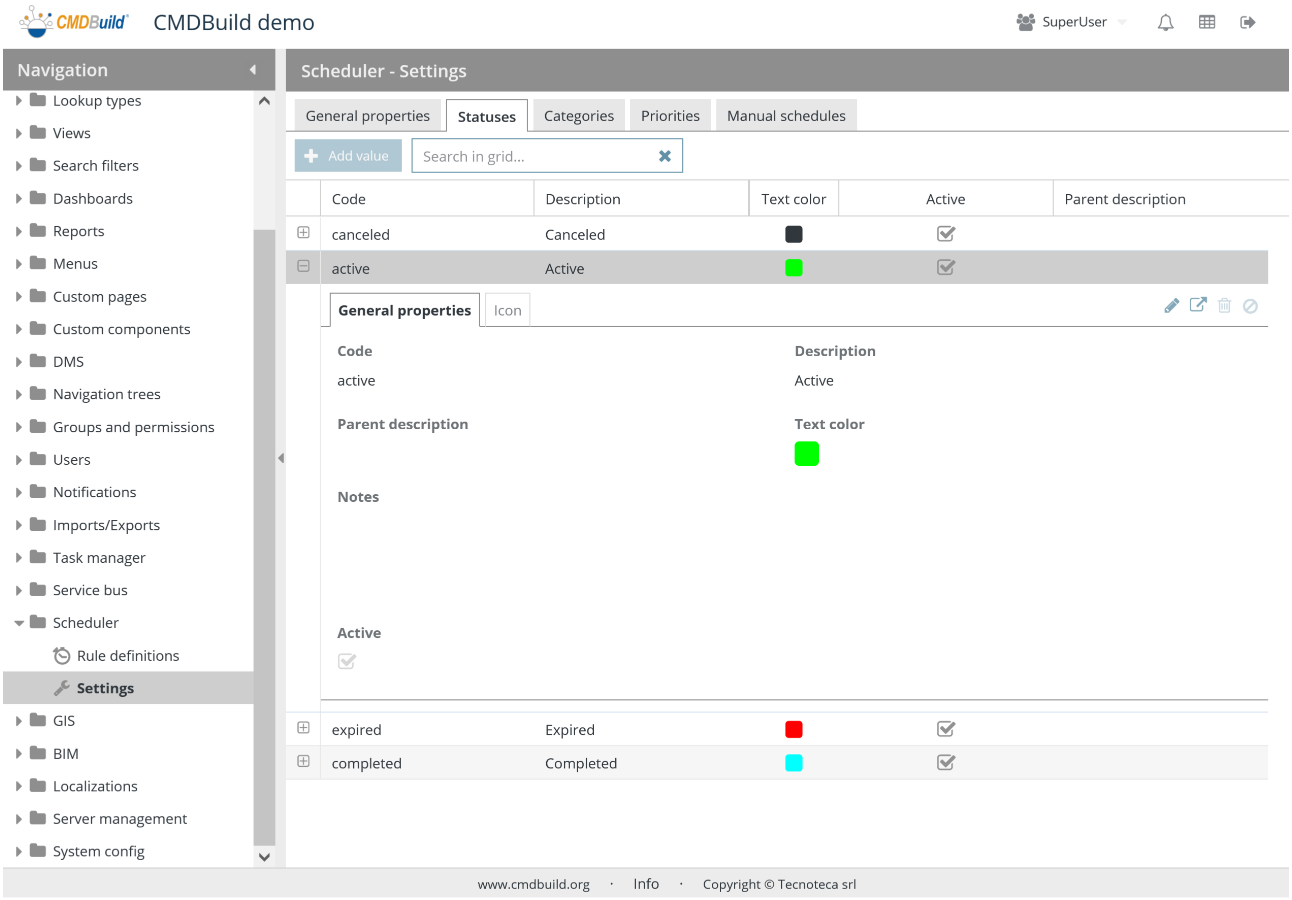
The following information is required:

- Active: it allows to activate the functionality of the scheduling file

Active

☑

26.2.2. List of values



Code	Description	Text color	Active	Parent description
canceled	Canceled	⬛	☑	
active	Active	🟩	☑	
expired	Expired	🟥	☑	
completed	Completed	🟦	☑	

Code
active

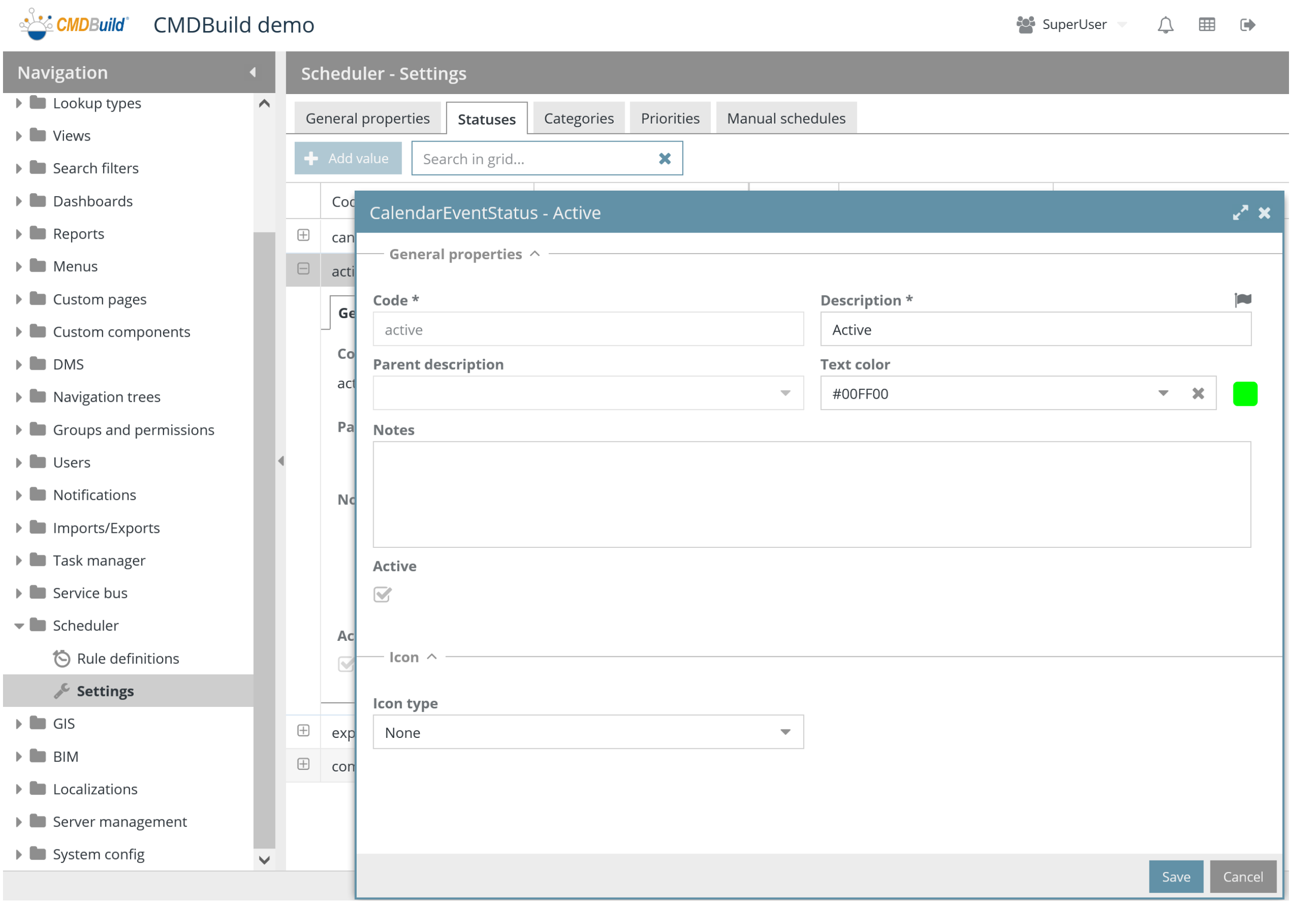
Description
Active

Parent description

Text color
🟩

Notes

Active
☑



CalendarEventStatus - Active

General properties

Code *: active

Description *: Active

Parent description:

Text color: #00FF00 🟩

Notes:

Active: ☑

Icon

Icon type: None

The management of states of the scheduling table is the same as the one seen in the Lookup lists. Please read the relative chapter for further details.

Since the scheduling file requests predefined states, it is possible to modify the description or colors, but it is impossible to insert new states or edit / disable the one available.

26.2.3. List of categories

The management of categories used by the scheduling table, customizable by your necessities, is the same as the one seen in the Lookup lists. Please read the relative chapter for further details.

26.2.4. Priority list

The management of priorities used by the scheduling table, customizable by your necessities, is the same as the one seen in the Lookup lists. Please read the relative chapter for further details.

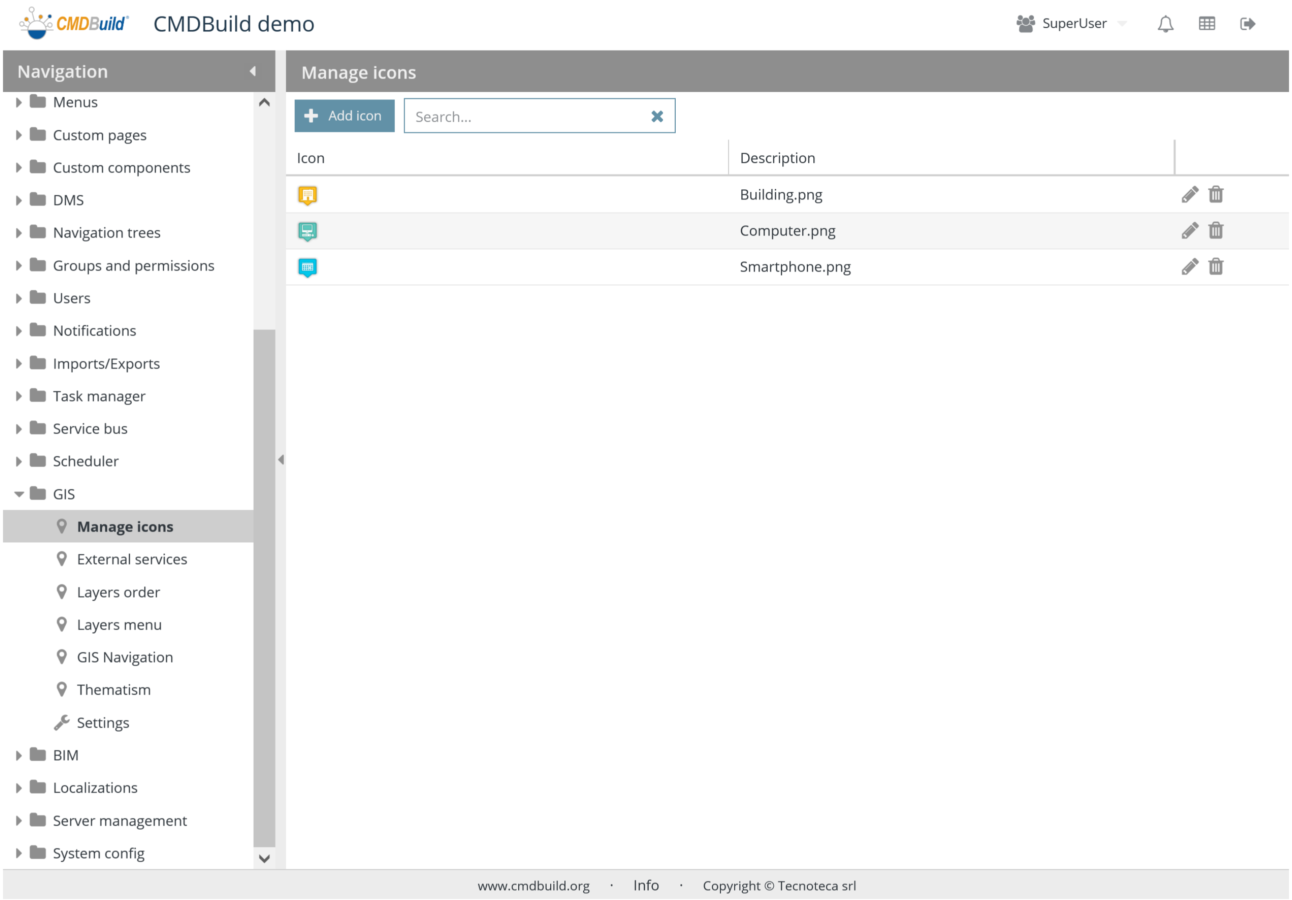
27. GIS

CMDBuild allows you to georeference on geographical maps and on planimetries those cards for which a geographical attribute has been configured.

Geographical attributes can be "point", "open polygonal" and "closed polygonal".

27.1. Icons management

This feature allows you to create and update a list of icons that are used to represent markers (see the management of geographical attributes in the menu Classes, paragraph 6.5).



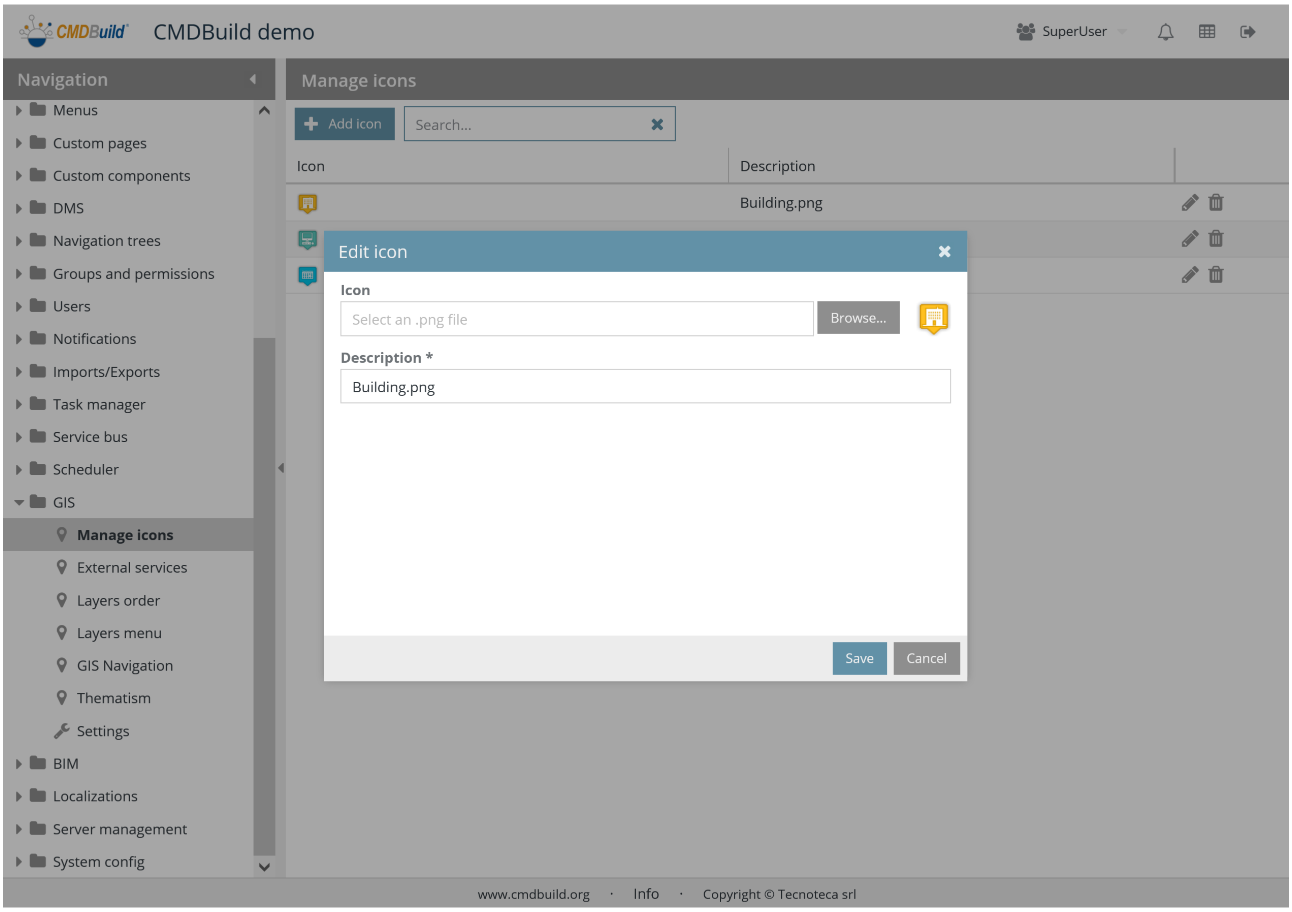
Icon	Description
	Building.png
	Computer.png
	Smartphone.png

There is the possibility of performing the following operations:

- Add a new icon	➕ Add icon
- research the icon in the list	Search... 🔍 ✖
- Edit the selected icon	✎

• Delete the selected icon	🗑

The upload of a new icon and the change of information regarding an old one are done through the following popup window.



Icon: Select an .png file

Description *: Building.png

The following information is required:

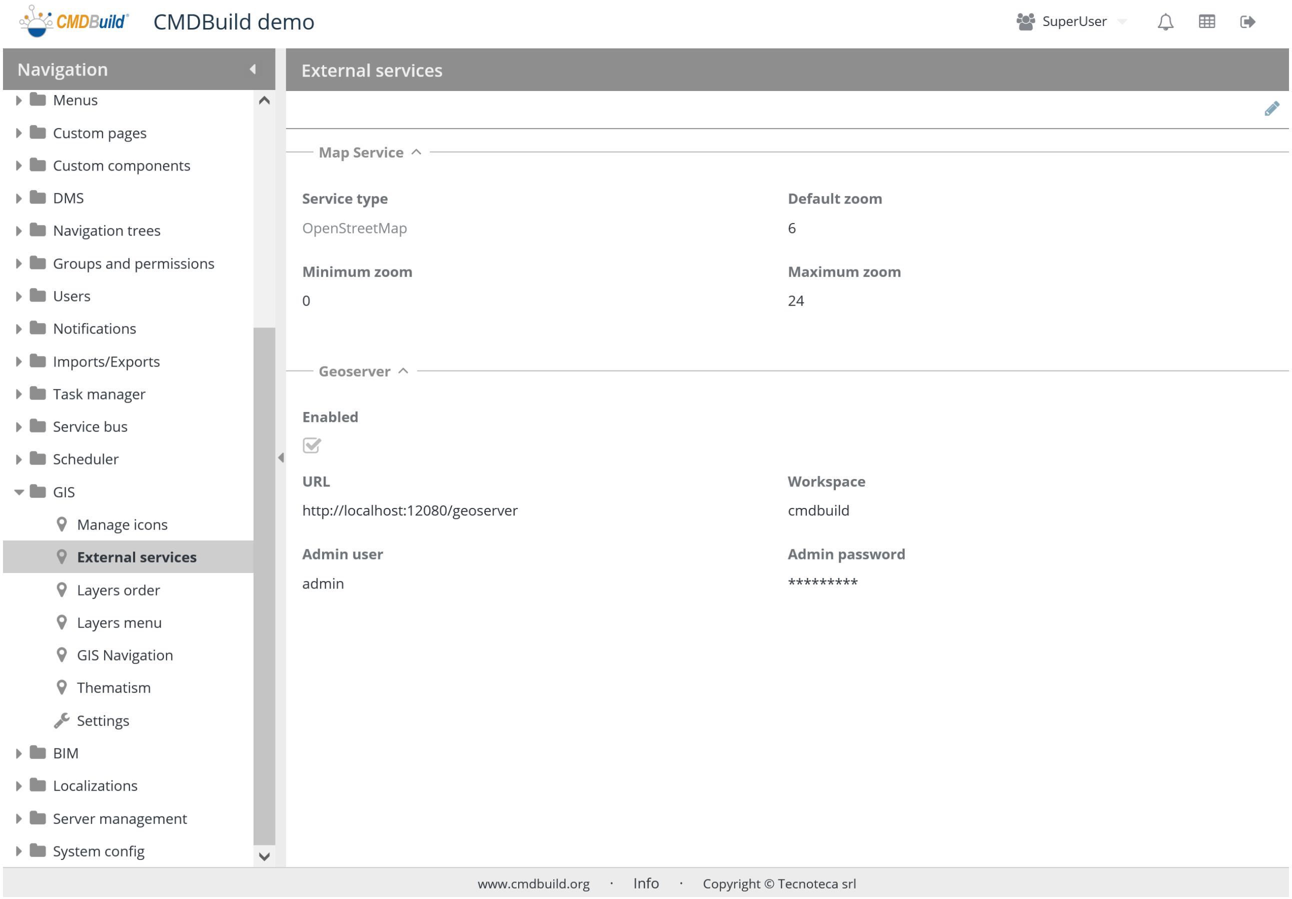
- Icon: file to upload, with PNG extension and 32x32 or 64x64 pixel
- Description: icon description

27.2. External services

This feature allows you to activate the external services necessary for the backgrounds useful to represent and georefer the cards.

The external services currently supported are:

- OpenStreetMap
- The GIS GeoServer, usually installed locally, through which you can load vectorial files like Shape (planimetries, etc.) or raster



Map Service

Service type: OpenStreetMap

Default zoom: 6

Minimum zoom: 0

Maximum zoom: 24

Geoserver

Enabled: ☑

URL: http://localhost:12080/geoserver

Workspace: cmdbuild

Admin user: admin

Admin password: *********

There is the possibility of performing the following operations:

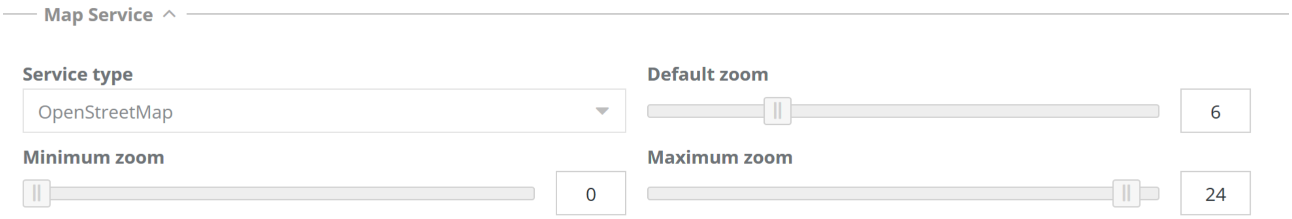
• edit parameters	✎

27.2.1. Map service

The following information is required:

- Service type: currently OpenStreetMap is available
- Default zoom: default zoom level required to display the map

- Minimum zoom: minimum zoom level required to display the map
- Maximum zoom: maximum zoom level required to display the map



Map Service

Service type: OpenStreetMap

Default zoom: 6

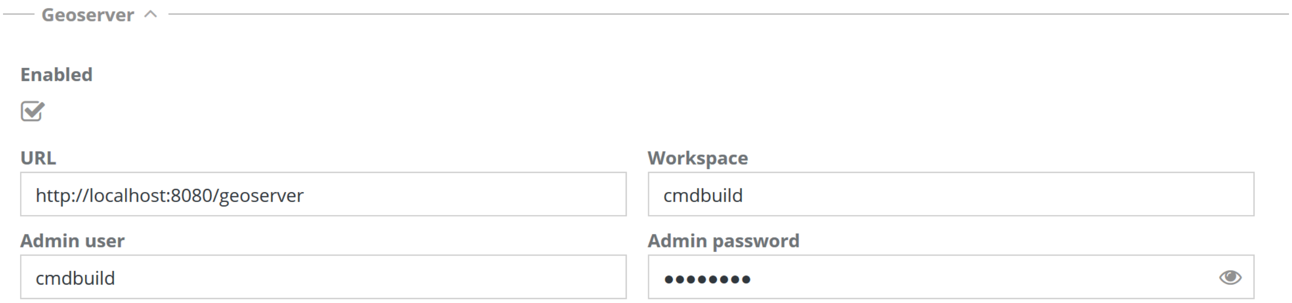
Minimum zoom: 0

Maximum zoom: 24

27.2.2. Geoserver

The following information is required:

- Enabled: it shows whether Geoserver is enabled or not
- URL: Geoserver address
- Workspace: workspace name
- admin username
- admin password



Geoserver

Enabled ☑

URL: http://localhost:8080/geoserver

Workspace: cmdbuild

Admin user: cmdbuild

Admin password: ••••••••

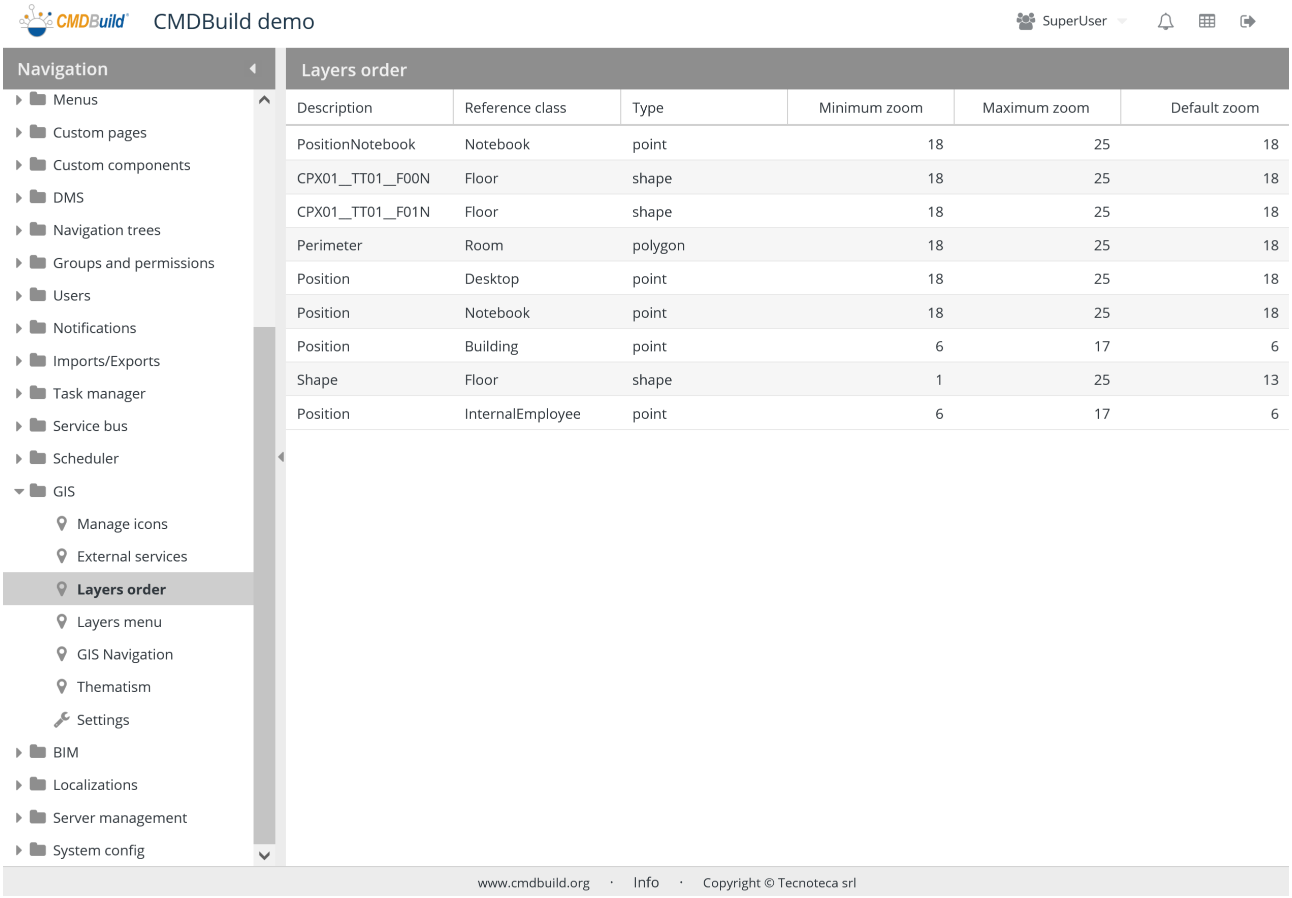
27.3. Layer order

This feature allows you to set layers order, so that they can be shown in the data Management Module.

Please note that CMDBuild manages the layers related to the geographical Attributes (asset locations, polygons, room surface) within the PostgreSQL database using the spacial extension PostGIS.

27.3.1. Features

The layers in the list correspond to one of the geographical attributes configured in the classes.



Layers order

Description	Reference class	Type	Minimum zoom	Maximum zoom	Default zoom
PositionNotebook	Notebook	point	18	25	18
CPX01__TT01__F00N	Floor	shape	18	25	18
CPX01__TT01__F01N	Floor	shape	18	25	18
Perimeter	Room	polygon	18	25	18
Position	Desktop	point	18	25	18
Position	Notebook	point	18	25	18
Position	Building	point	6	17	6
Shape	Floor	shape	1	25	13
Position	InternalEmployee	point	6	17	6

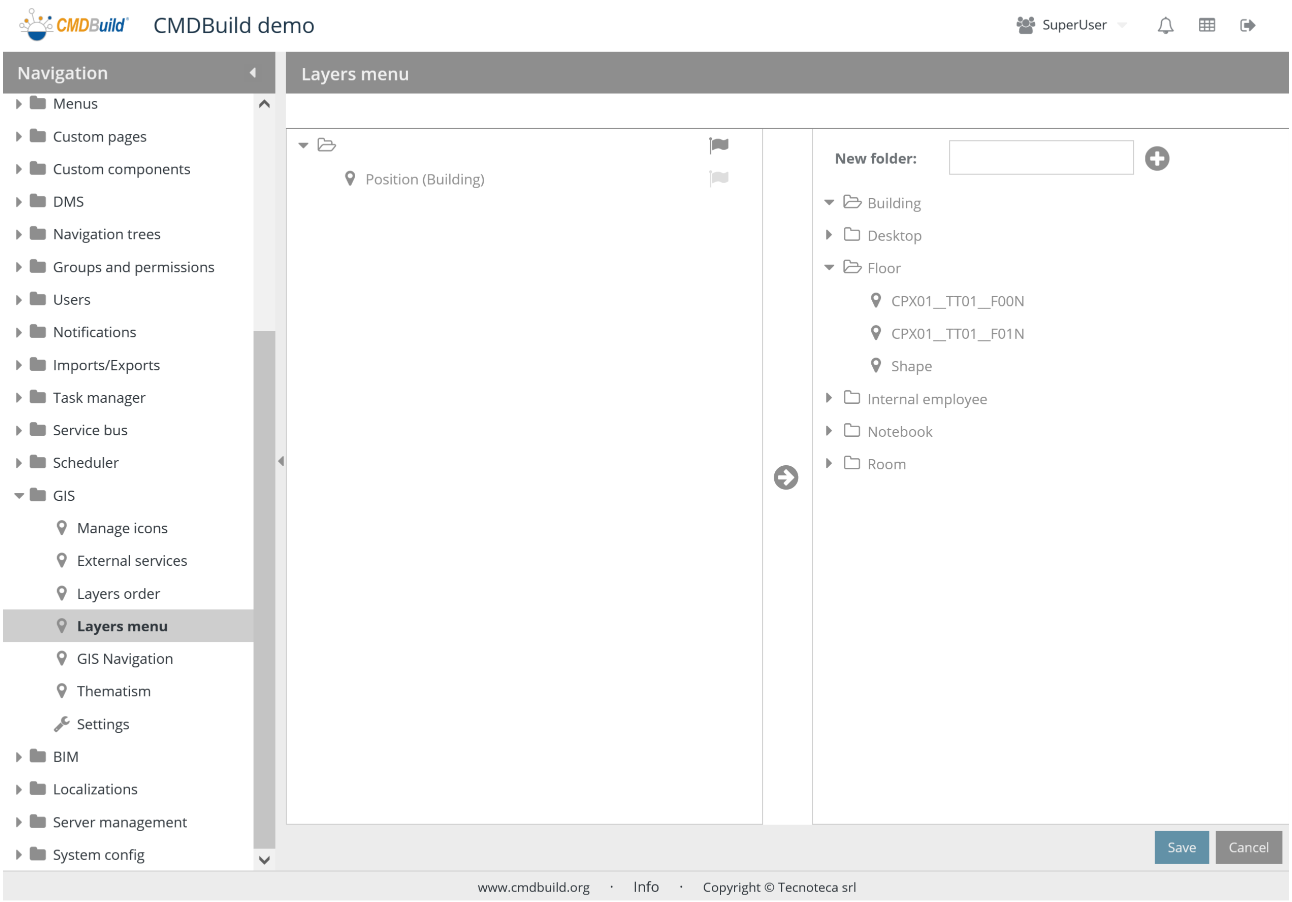
The layers order can be modified by moving the lines in the grid (drag and drop).

27.4. Layers menu

This task allows you to define the menu for the layer consultation configured in CMDBuild.

27.4.1. Features

This function allows you to configure a menu that is similar the CMDBuild general one. It is displayed in the Layers tab of the Management Module map.



The configuration modes are the same described for the general menu of the application, with the possibility of creating folders and positioning the available layers through the "drag & drop" function.

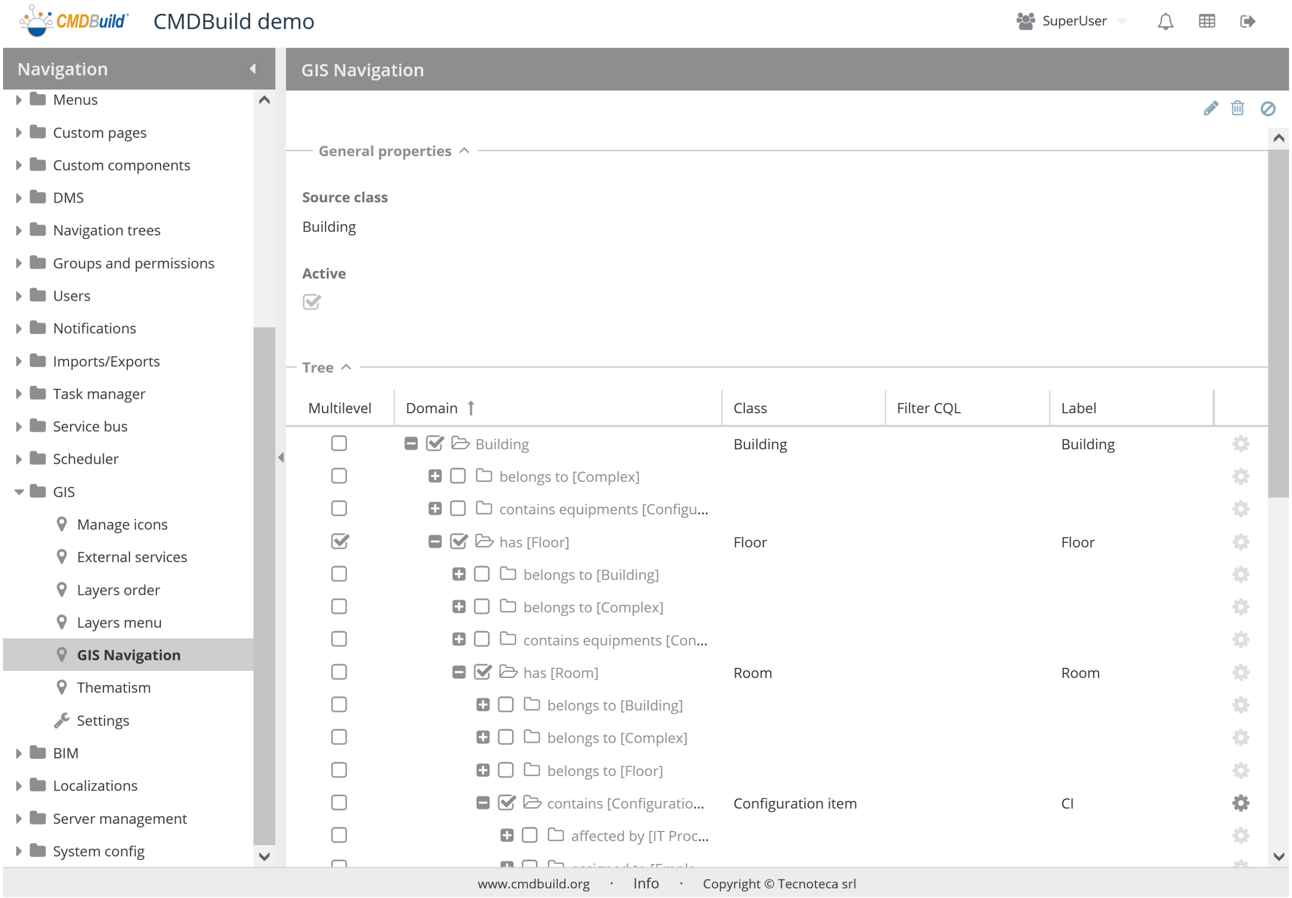
27.5. GIS navigation tree

The "Map" page in the Management Module requires the configuration of a GIS navigation tree that leads the operator in the navigation among the various types of cards georefered on the territory and/or on plans.

This means that the structure of the hierarchical menu has to be defined. This menu will present the list of elements of each level (for example, by expanding the card of a floor, you will see the cards related to the rooms and inside these the cards related to the assets) and you have the possibility of highlighting them on the map.

27.5.1. Features

The function allows you to configure the hierarchical navigation menu among the items georefered on the map.



General properties

Source class
Building

Active
☑

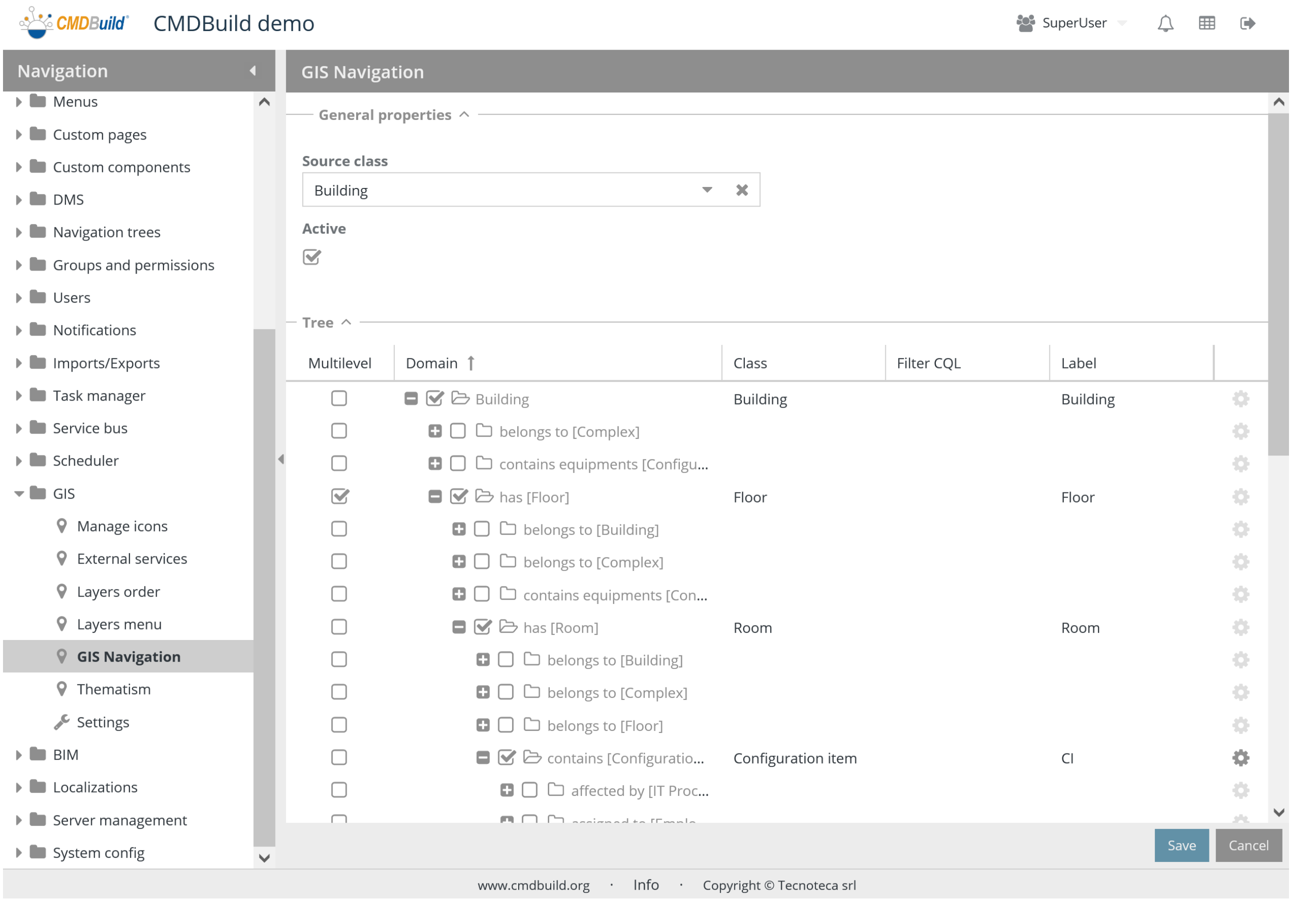
Tree

Multilevel	Domain	Class	Filter CQL	Label
□	⊟ ☑ Building	Building		Building
□	⊞ □ belongs to [Complex]			
□	⊞ □ contains equipments [Configu...			
☑	⊟ ☑ has [Floor]	Floor		Floor
□	⊞ □ belongs to [Building]			
□	⊞ □ belongs to [Complex]			
□	⊞ □ contains equipments [Con...			
□	⊟ ☑ has [Room]	Room		Room
□	⊞ □ belongs to [Building]			
□	⊞ □ belongs to [Complex]			
□	⊞ □ belongs to [Floor]			
□	⊟ ☑ contains [Configuratio...	Configuration item		CI
□	⊞ □ affected by [IT Proc...			

There is the possibility of performing the following operations:

• edit the GIS navigation tree	✎

- delete the GIS navigation tree	🗑
- disable the GIS navigation tree (i.e. set the status as non-active)	⊘



General properties

Source class: Building

Active: ☑

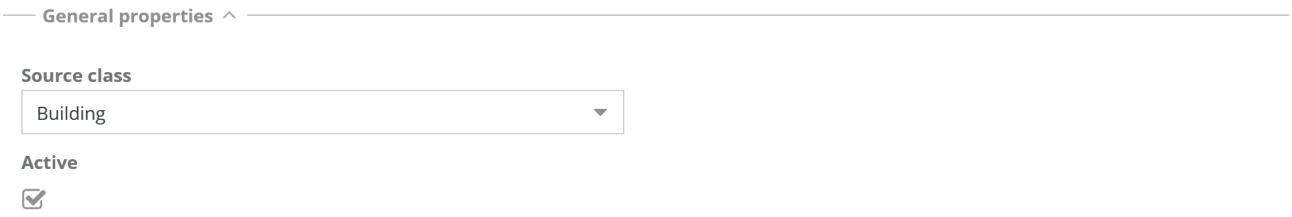
Tree

Multilevel	Domain	Class	Filter CQL	Label
☐	⊟ ☑ Building	Building		Building
☐	⊞ ☐ belongs to [Complex]			
☐	⊞ ☐ contains equipments [Configu...			
☑	⊟ ☑ has [Floor]	Floor		Floor
☐	⊞ ☐ belongs to [Building]			
☐	⊞ ☐ belongs to [Complex]			
☐	⊞ ☐ contains equipments [Con...			
☐	⊟ ☑ has [Room]	Room		Room
☐	⊞ ☐ belongs to [Building]			
☐	⊞ ☐ belongs to [Complex]			
☐	⊞ ☐ belongs to [Floor]			
☐	⊟ ☑ contains [Configuratio...	Configuration item		CI
☐	⊞ ☐ affected by [IT Proc...			

27.5.2. General properties

The following information is required:

- Original class: root of the tree (in general "Building" or "Complex")
- Active: the GIS navigation tree is active and can be used



General properties

Source class: Building

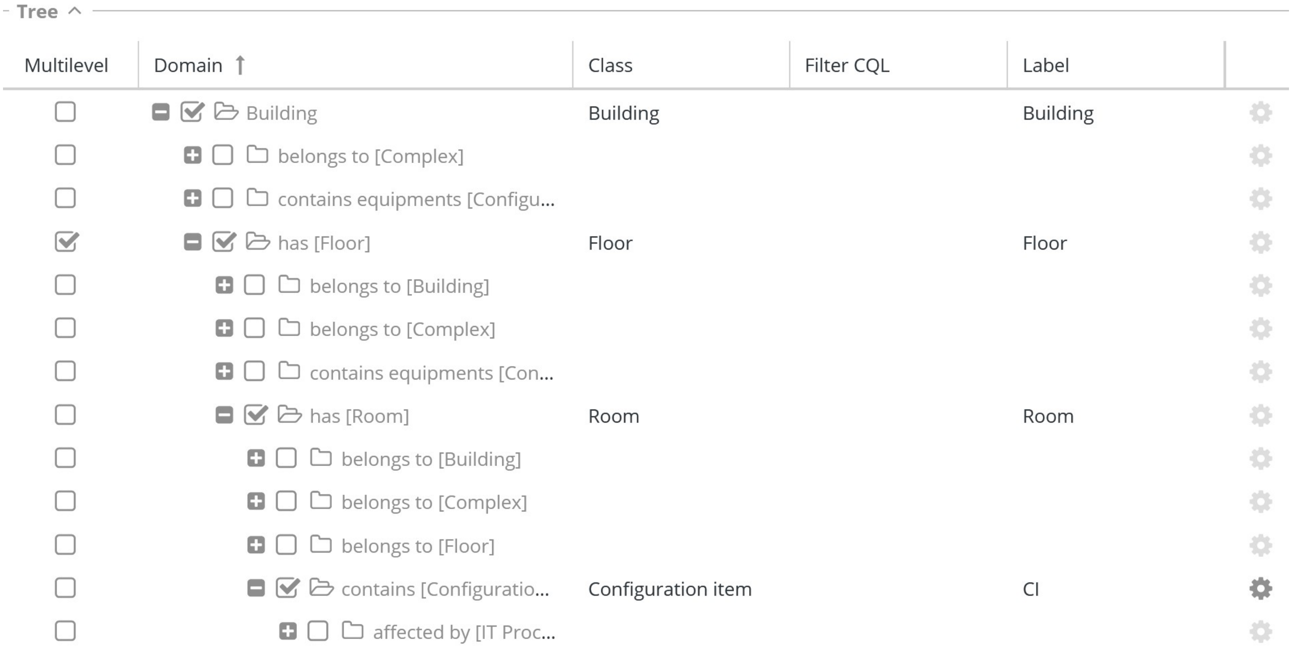
Active: ☑

27.5.3. Navigation tree

Starting from the root class 1:N the class tree is displayed and for each class the following are required:

- Multilevel: it is useful to identify the "vertical development" class, which is the one capable of overlapping and needs to be displayed mutually exclusive (usually related to the "Floor" class).
- Class: it allows you to select the classes of the GIS navigation tree you want to configure
- CQL filter: it allows you to set filters to exclude some cards of the selected classes
- Alias to be used as labels

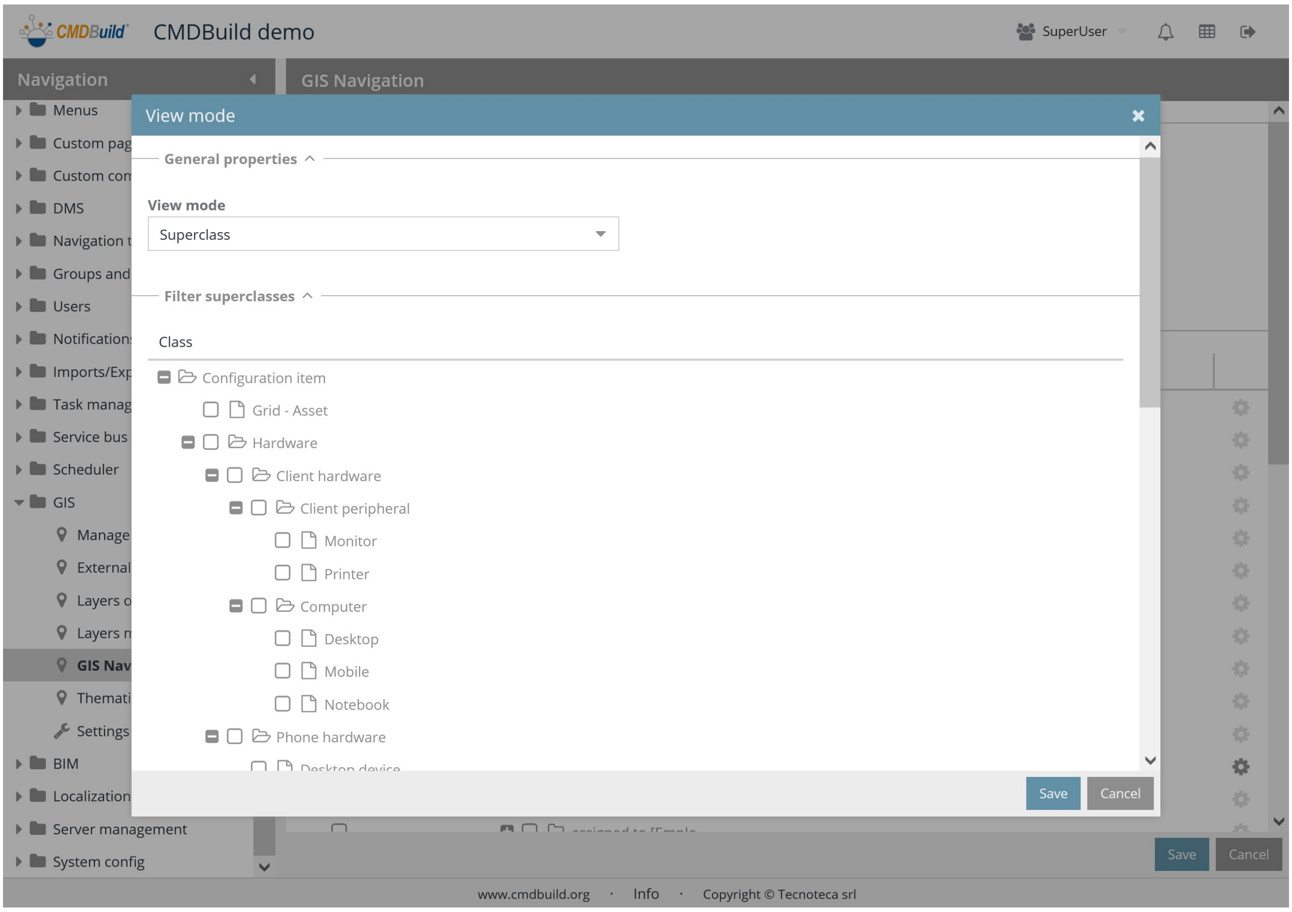
Providing the necessary domains, a typical navigation tree is the one made up of the following classes: Building => Floor => Room => Asset.



Tree

Multilevel	Domain ↑	Class	Filter CQL	Label	
□	⊟ ☑ Building	Building		Building	⚙
□	⊞ □ belongs to [Complex]				⚙
□	⊞ □ contains equipments [Configu...				⚙
☑	⊟ ☑ has [Floor]	Floor		Floor	⚙
□	⊞ □ belongs to [Building]				⚙
□	⊞ □ belongs to [Complex]				⚙
□	⊞ □ contains equipments [Con...				⚙
□	⊟ ☑ has [Room]	Room		Room	⚙
□	⊞ □ belongs to [Building]				⚙
□	⊞ □ belongs to [Complex]				⚙
□	⊞ □ belongs to [Floor]				⚙
□	⊟ ☑ contains [Configuratio...	Configuration item		CI	⚙
□	⊞ □ affected by [IT Proc...				⚙

Through the icon in the column on the right (cogwheel), enabled only for superclasses, you can filter the subclasses you want to include in the menu and manage a custom label from this function.



View mode: Superclass

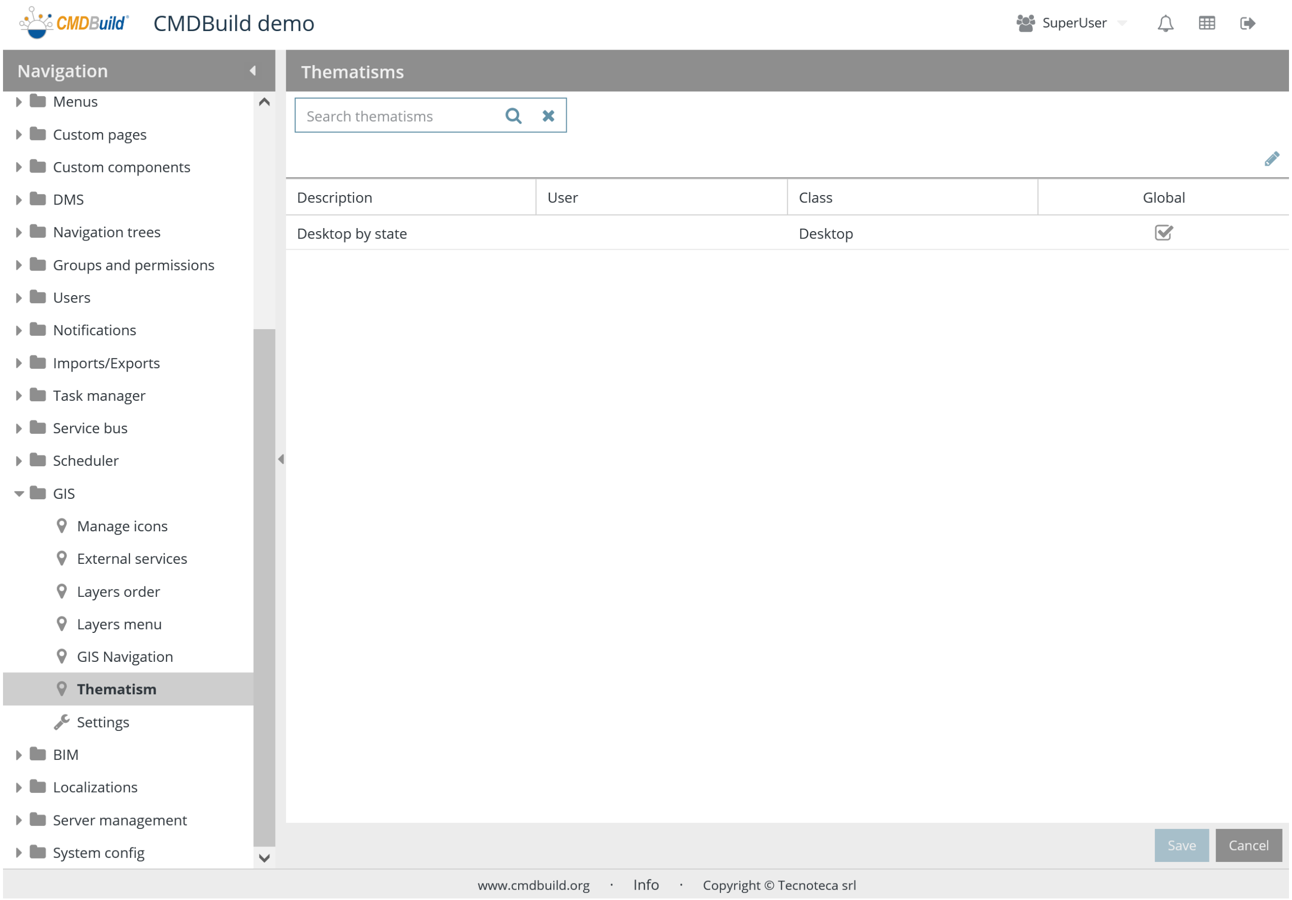
27.6. Thematic maps

From the GIS interface of the data Management Module you can configure thematic maps on displayed data, which highlight the output of analysis functions on the visible entities with various colors or different symbols (e.g. the rooms of a plan can be colored according to their intended use, computers can be shown according to the number of breakdowns, etc.).

Thematic maps can be also saved by their creator, remaining available in the following accesses.

27.6.1. Features

This function allows the administrator to view the thematic maps configured by some users and make them visible to everybody.

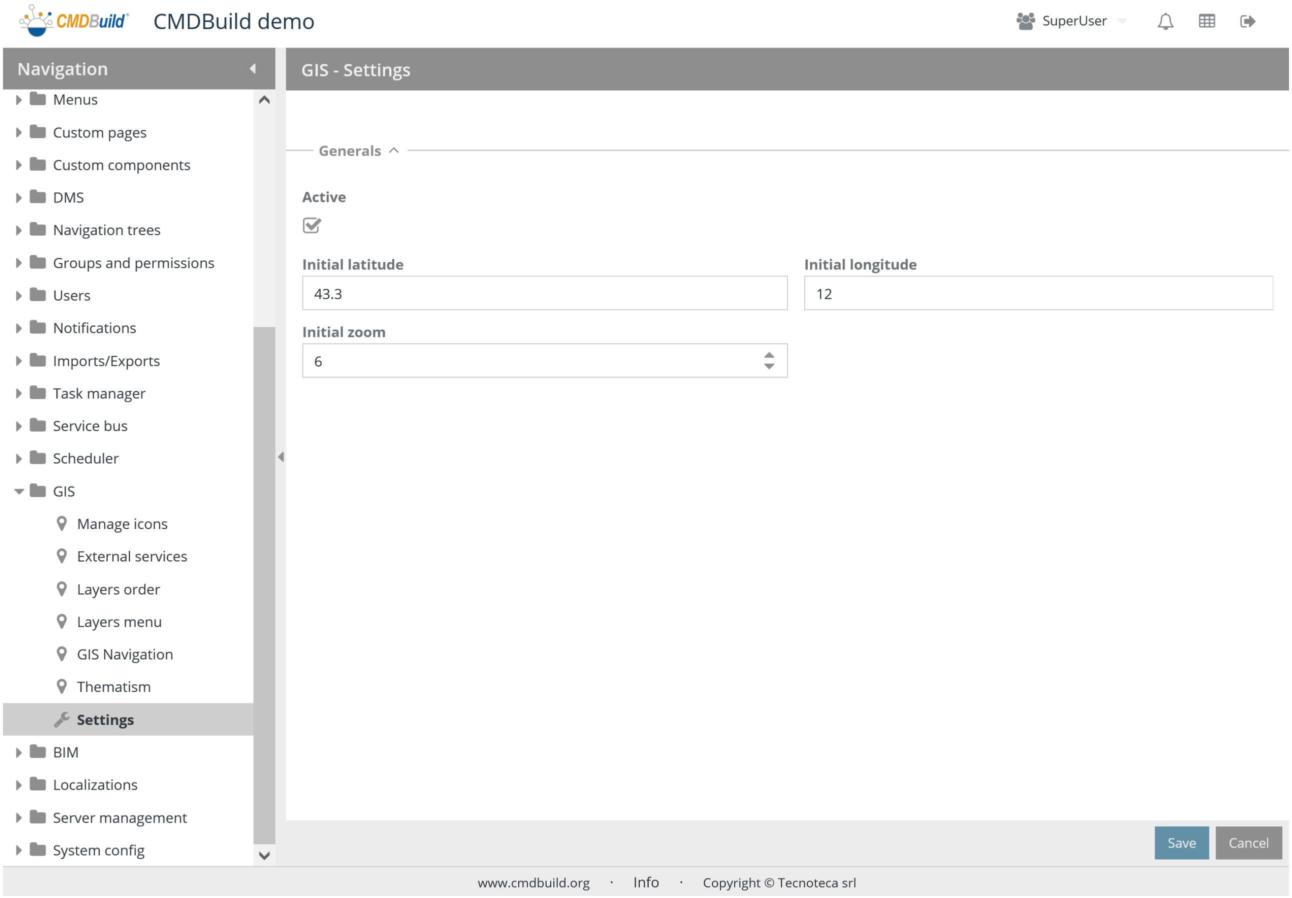


Description	User	Class	Global
Desktop by state		Desktop	☑

For a complete description of the configuration mode, please see the related section of the User Manual.

27.7. Configurations

This function allows you to set some configuration parameters of 2D georeference functions available in the CMDBuild system, both on maps and on vector maps.



Active
☑

Initial latitude: 43.3

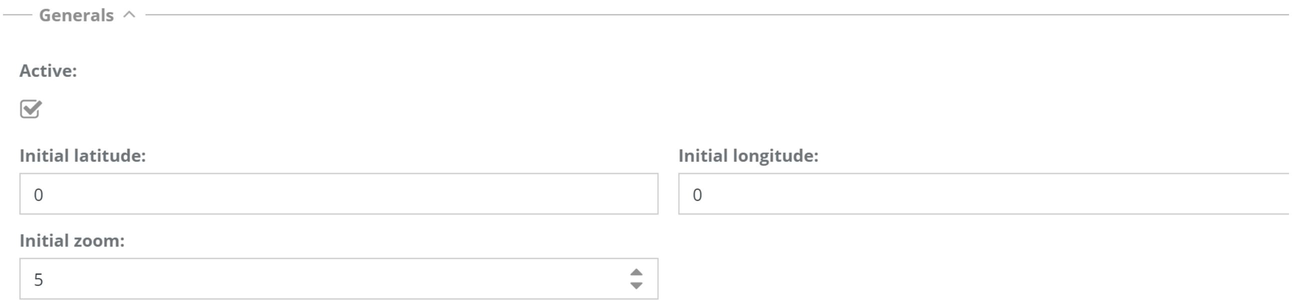
Initial longitude: 12

Initial zoom: 6

27.7.1. General properties

The following information is required:

- Active: it allows you to activate 2D georeference features
- Initial latitude: it allows you yo specify the map latitude at startup
- Initial longitude: it allows you yo specify the map longitude at startup
- Initial zoom level: it allows you to specify the zoom level at startup



Generals

Active: ☑

Initial latitude: 0	Initial longitude: 0
Initial zoom: 5	

28. BIM

CMDBuild supports the BIM paradigm (Building Information Modeling), which is a conceptual solution with the aim of managing the building design procedure, from the conception of the building to its construction, use and maintenance, and finally to its demolition, if any.

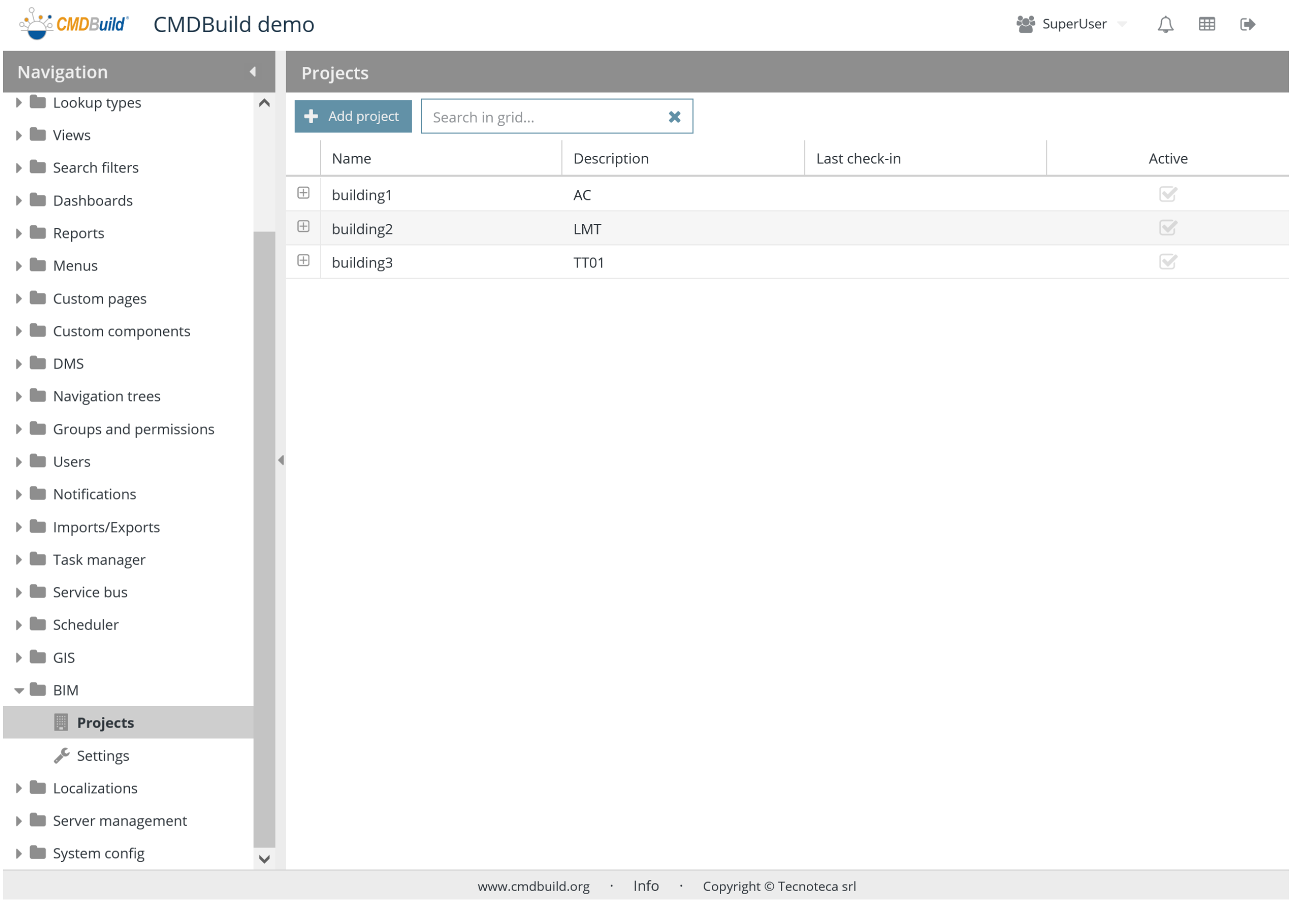
CMDBuild is able to sync automatically the chosen information from a BIM repository by using the open standard format IFC (Industry Foundation Classes) and display in a realistic way interiors and assets thanks to a 3D interactive viewer, integrated in the user interface.

CMDBuild uses as embedded repository the open source tool BiMserver.

28.1. Projects

This function allows to manage BIM projects managed in CMDBuild. Every project is usually associated to a building or to a built up area.

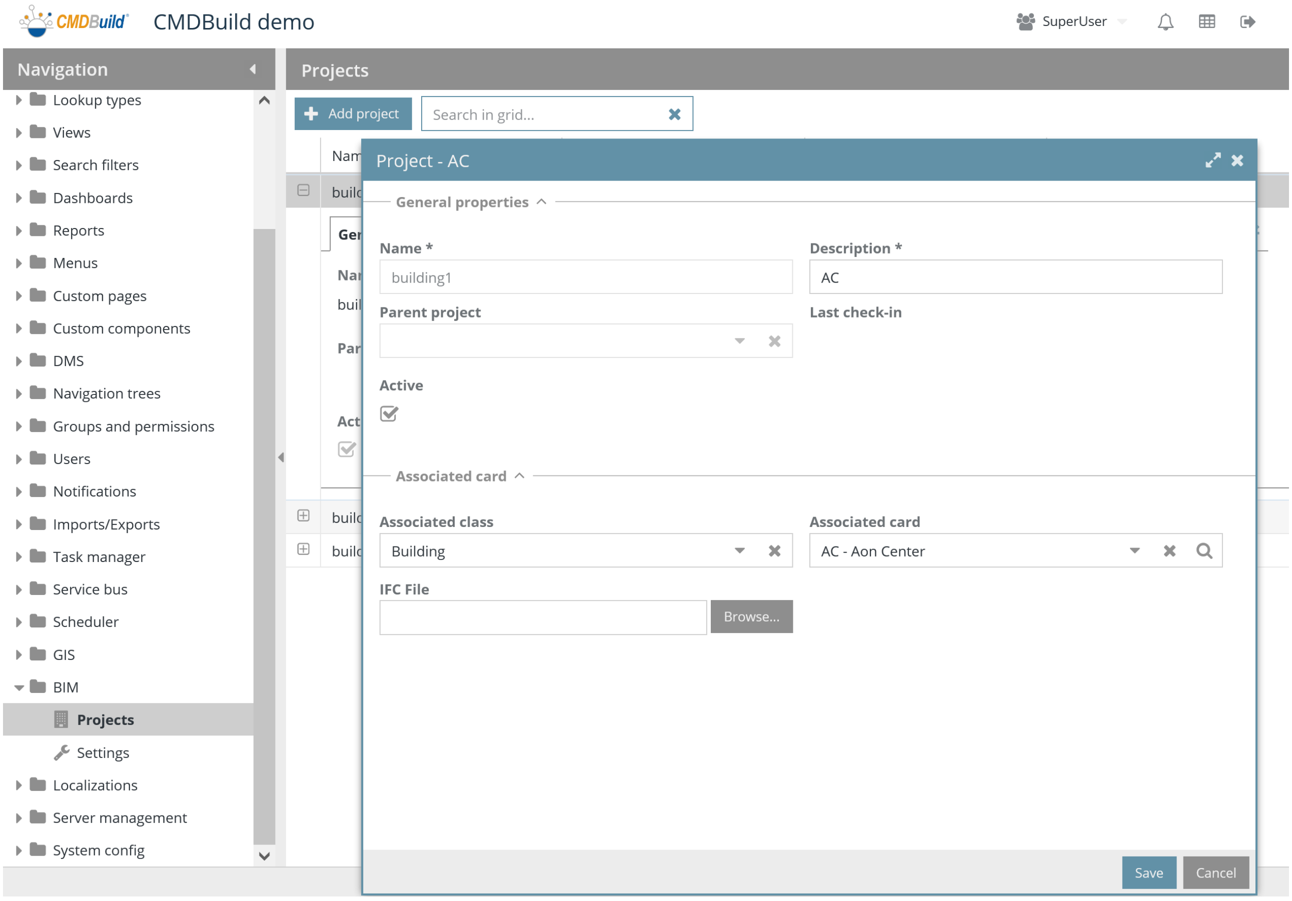
A BIM project coincides to a IFC file in various versions.



Name	Description	Last check-in	Active
building1	AC		✓
building2	LMT		✓
building3	TT01		✓

There is the possibility of performing the following operations:

• create a new BIM project	✚ Add project
• do searches on configured projects	Search... 🔍 ✖
• edit the selected project	✎
• display the complete information related to the selected project	↗
• download the selected project	⤓
• clone the selected project	⧉
• delete the selected project	🗑
• disable the selected project (i.e. set the status as non-active)	⊘
• Convert IFC format to XKT (supported by Xeokit)	⇄



Project - AC

General properties

Name * building1

Description * AC

Parent project

Last check-in

Active ☑

Associated card

Associated class Building

Associated card AC - Aon Center

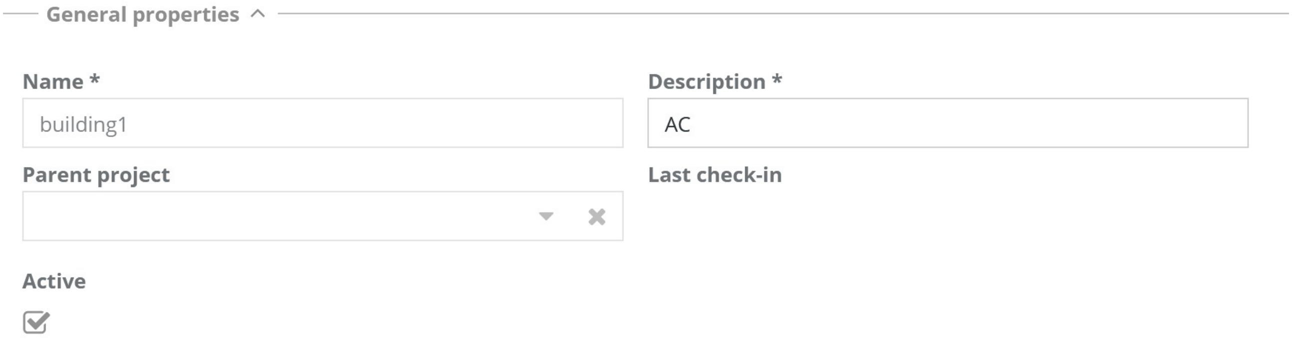
IFC File

Each BIM project needs some parameters to be compiled (metadata), as mentioned.

28.1.1. General properties

The following information is required:

- Name: project name
- Description: description of the project
- Parent project: it allows you to split a BIM model in various elements, having a parent model and several specific submodels for different plants, which can be automatically charged.
- Last check-in: last check-in date
- Active: the project is active and can be used



General properties

Name *: building1

Description *: AC

Parent project:

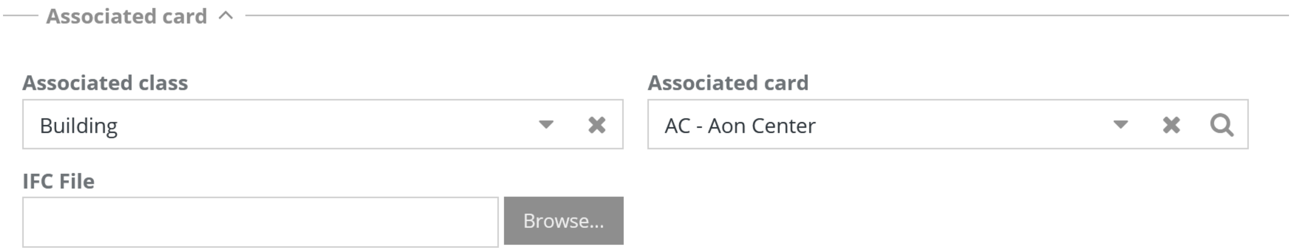
Last check-in:

Active: ☑

28.1.2. Linked cards

The following information is required:

- Related class: it requires to select the CMDBuild class for the project connection; it is usually "Building" or "Complex of buildings"
- Related card: it requires to select the CMDBuild class selected in the previous field; it is usually a building or a complex of buildings
- IFC file: it requires the uploading of the IFC file to in order to file in into BiMserver



Associated card

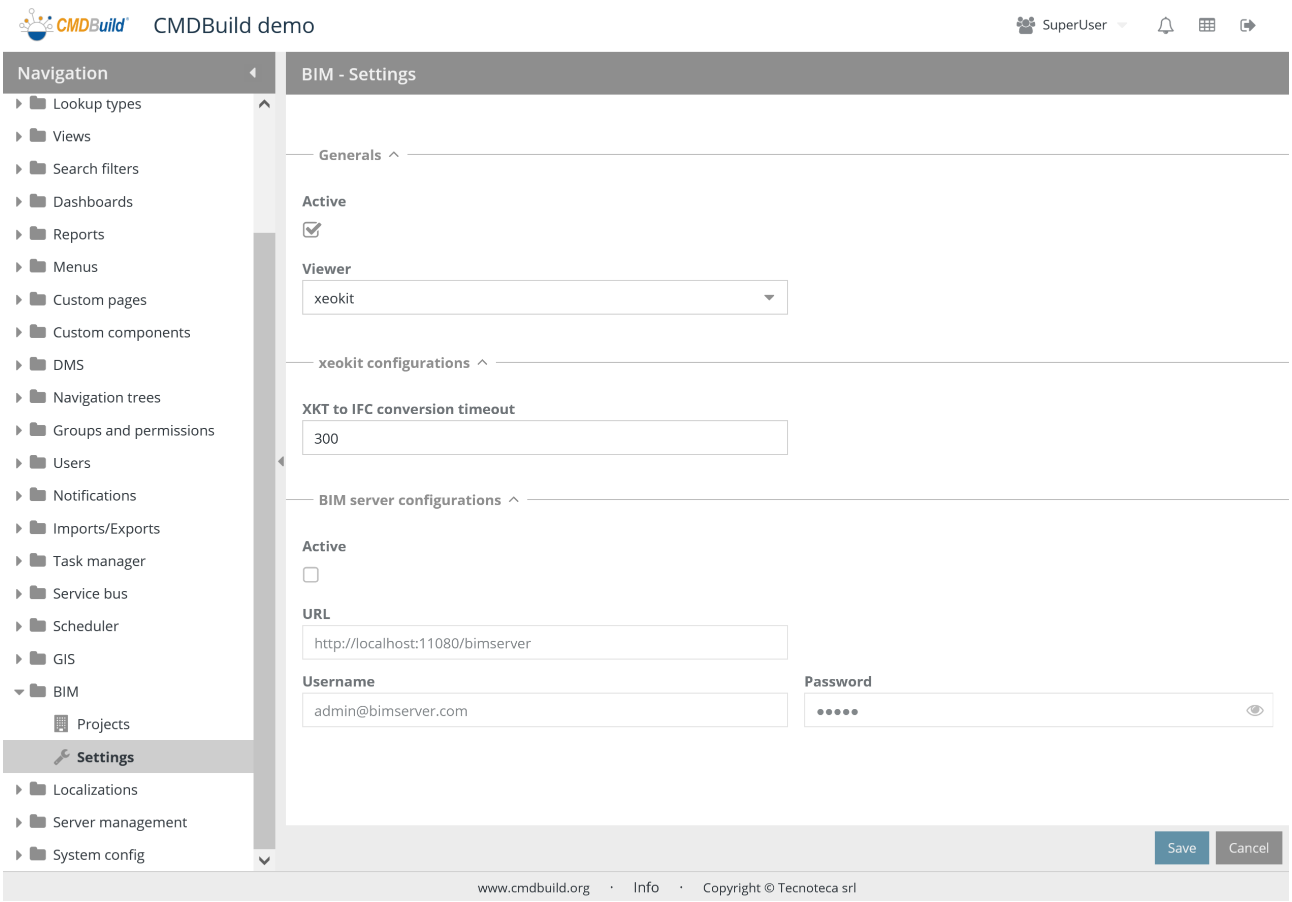
Associated class: Building

Associated card: AC - Aon Center

IFC File:

28.2. Configurations

This function allows you to set some configuration parameters of 3D georeference functions available in the CMDBuild system, related to the interface with BIM systems (Building Information Modeling) through IFC file (Industry Foundation Classes).



BIM - Settings

Generals

Active
☑

Viewer
xeokit

xeokit configurations

XKT to IFC conversion timeout
300

BIM server configurations

Active
□

URL
http://localhost:11080/bimserver

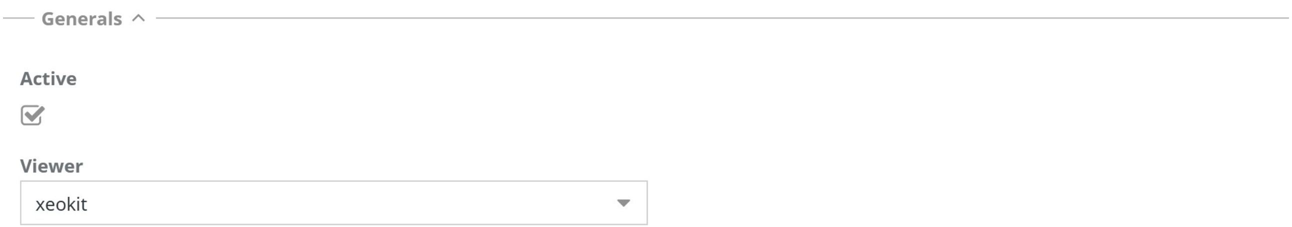
Username
admin@bimserver.com

Password
•••••

28.2.1. General properties

The following information is required:

- Active: it allows you to activate 3D georeference features
- Viewer: BIMSurfer or xeokit



Generals

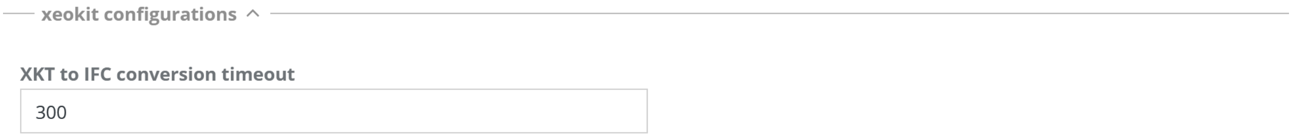
Active
☑

Viewer
xeokit

28.2.2. Xeokit configuration

The following information is required:

- XKT to IFC conversion timeout: maximal timeout for conversions from XTK format (native of xeokit) to IFC

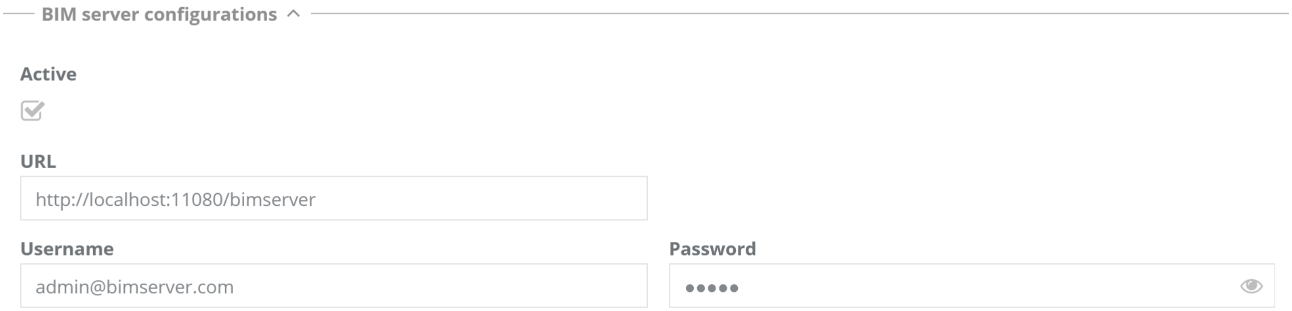


XKT to IFC conversion timeout: 300

28.2.3. BIMServer configuration

The following information is required:

- Active: active BIMServer
- URL: URL of the BIMServer service
- Username: service user used to access the BiMserver
- Password: service user's password



Active: ☑

URL: http://localhost:11080/bimserver

Username: admin@bimserver.com

Password: ●●●●●

29. Localization

In CMDBuild there are two localization typologies:

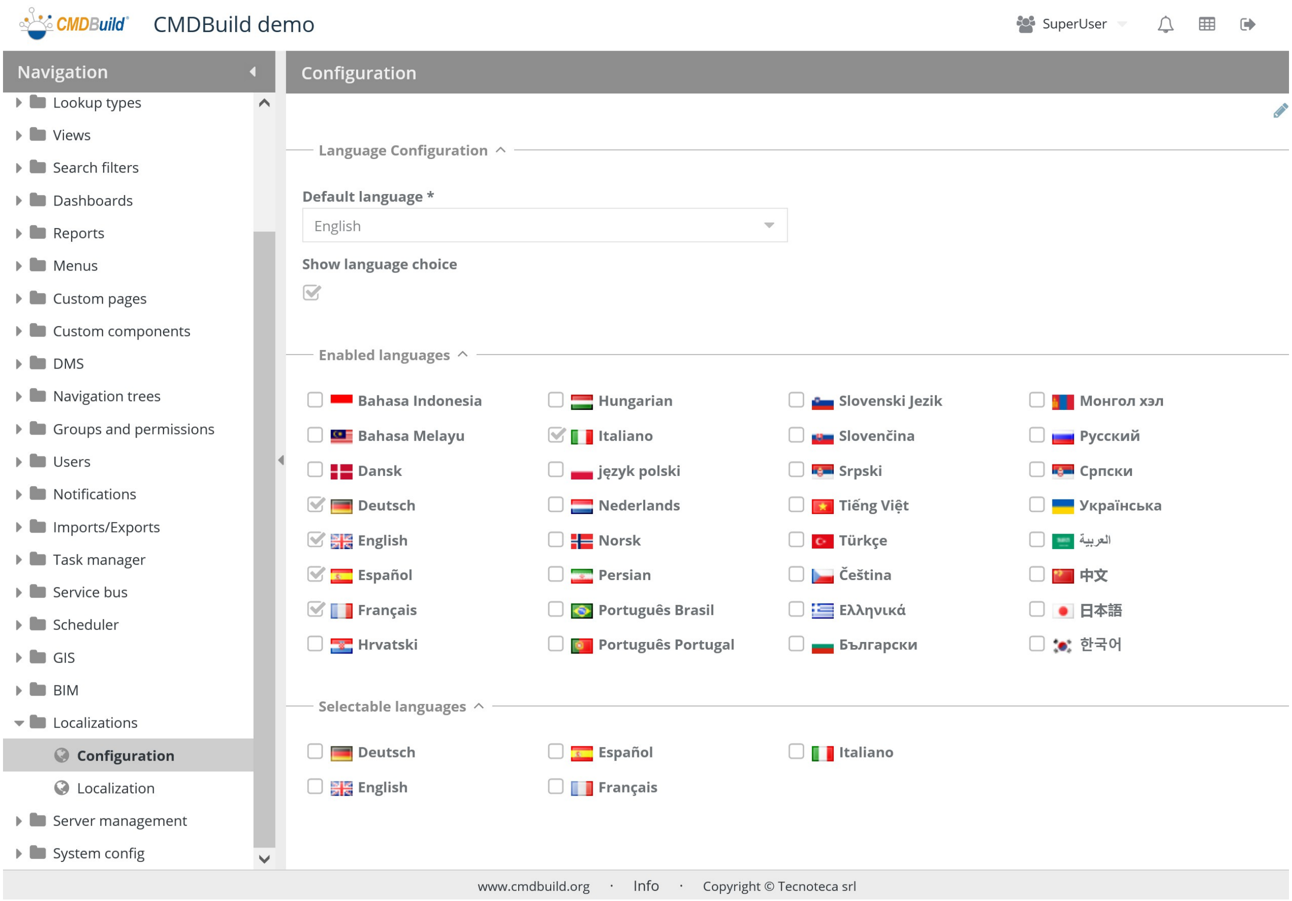
- a first-level localization, related to the only base CMDBuild interface (menu texts and standard buttons, headers, footers, etc.)
- a second-level localization, related to all application elements (class names, attributes, domains, lookups, menu entries, etc.)

The first-level localization is managed through external system files.

The second-level localization is managed in this Administration Manual, by working on each page where there are elements configured with texts to be localized, or globally through the features described in this chapter.

29.1. Configuration

In the configuration page you can select the default language of the CMDBuild default instance, decide to permit or not to choose the language when logging in and select the languages for the current instance, among those with the first-level localization.



Configuration

Language Configuration

Default language *
English

Show language choice
☑

Enabled languages

□ Bahasa Indonesia
□ Bahasa Melayu
□ Dansk
☑ Deutsch
☑ English
☑ Español
☑ Français
□ Hrvatski
□ Hungarian
☑ Italiano
□ język polski
□ Nederlands
□ Norsk
□ Persian
□ Português Brasil
□ Português Portugal
□ Slovenski Jezik
□ Slovenčina
□ Srpski
□ Tiếng Việt
□ Türkçe
□ Čeština
□ Ελληνικά
□ Български
□ Монгол хэл
□ Русский
□ Српски
□ Українська
□ العربية
□ 中文
□ 日本語
□ 한국어

Selectable languages

□ Deutsch
□ English
□ Español
□ Français
□ Italiano

There is the possibility of performing the following operations:

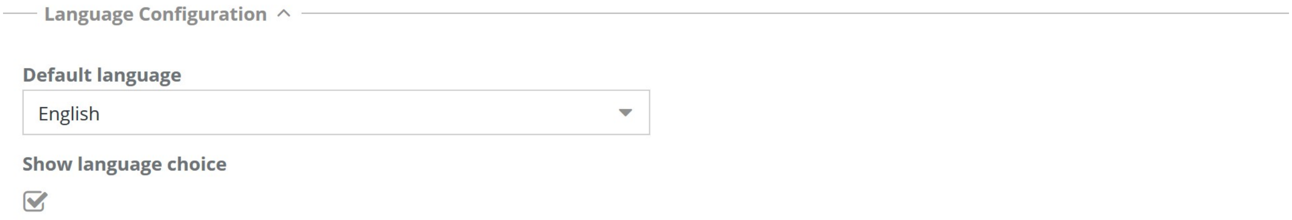
• enter the edit mode to edit the configuration parameters	✎

The parameters to configure the localization are divided in the two sections described as follows.

29.1.1. Language configuration

The following information is required:

- Default language: default language for the CMDBuild instance
- Show language selection: it requires the list of languages when logging-in



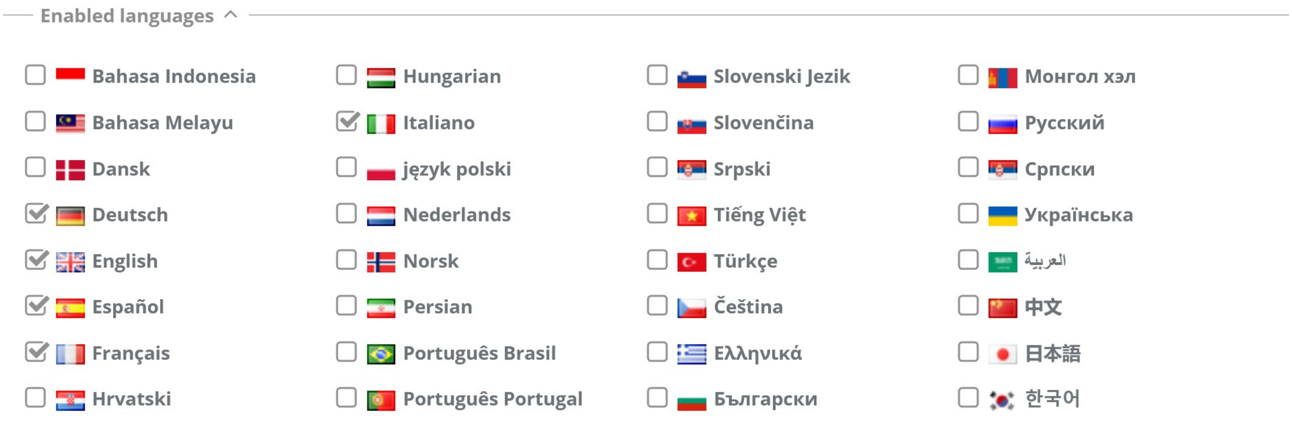
Language Configuration ^

Default language
English ▾

Show language choice
☑

29.1.2. Enabled languages

Then you can select the languages enabled for the localization, among those with the first-level localization.

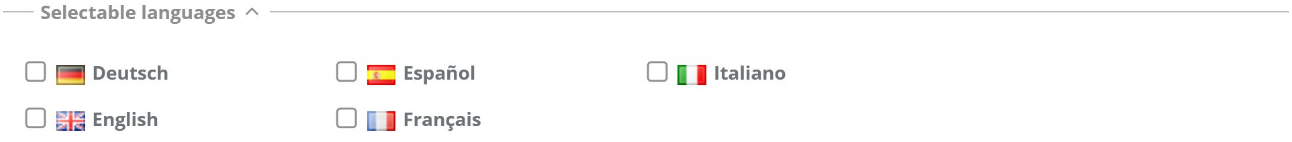


Enabled languages ^

☐ Bahasa Indonesia
☐ Bahasa Melayu
☐ Dansk
☑ Deutsch
☑ English
☑ Español
☑ Français
☐ Hrvatski
☐ Hungarian
☑ Italiano
☐ język polski
☐ Nederlands
☐ Norsk
☐ Persian
☐ Português Brasil
☐ Português Portugal
☐ Slovenski Jezik
☐ Slovenčina
☐ Srpski
☐ Tiếng Việt
☐ Türkçe
☐ Čeština
☐ Ελληνικά
☐ Български
☐ Монгол хэл
☐ Русский
☐ Српски
☐ Українська
☐ العربية
☐ 中文
☐ 日本語
☐ 한국어

29.1.3. Selectable languages

You can select also the languages available at login in the application.



Selectable languages ^

☐ Deutsch
☐ English
☐ Español
☐ Français
☐ Italiano

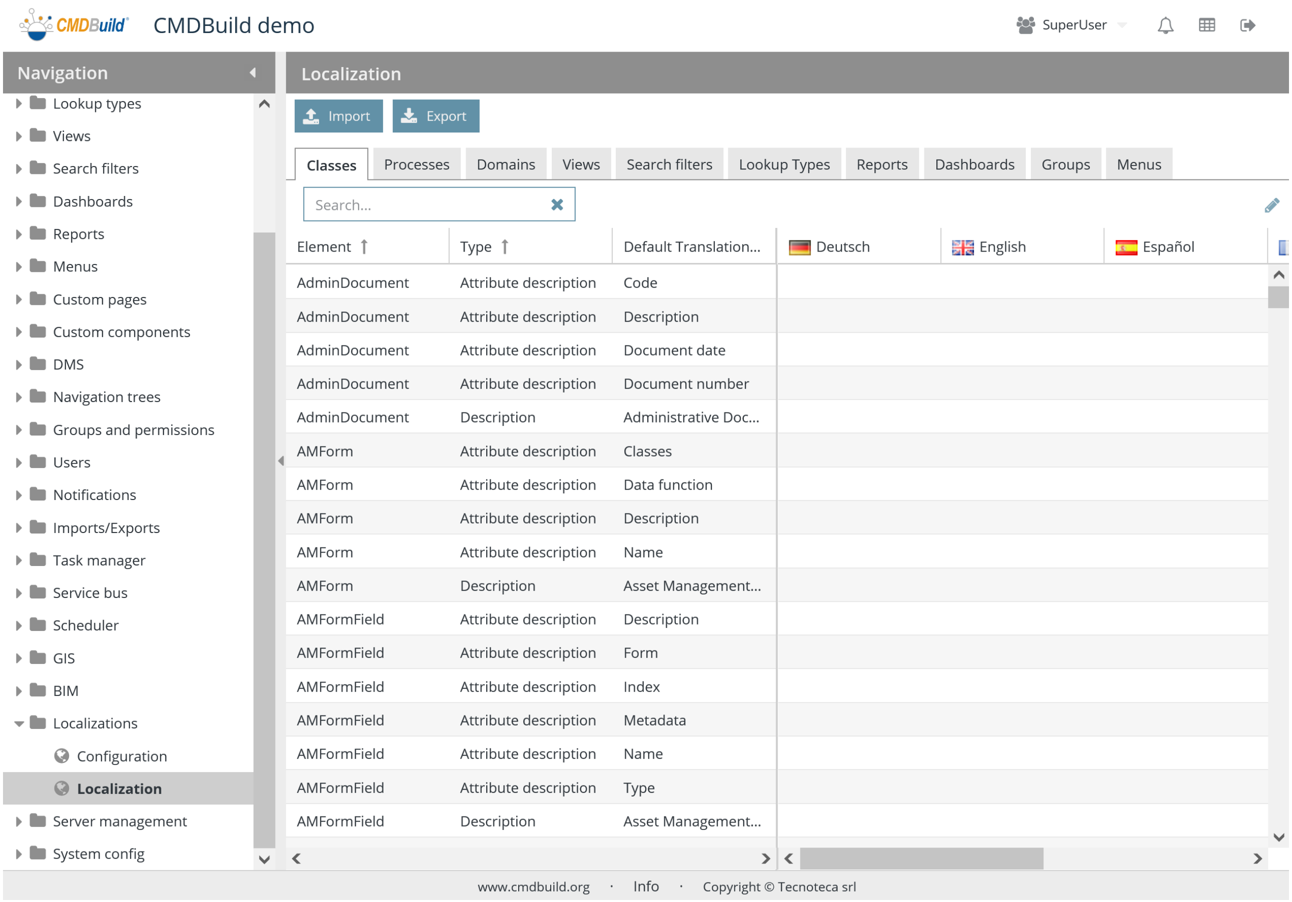
29.2. Localization

The localization interface offers a single intervention point in order to work on every translation:

- Classes: description of classes, description of attribute groups, description of attributes
- Processes: description of processes, description of attribute groups, description of attributes
- Domains: domain description, direct description, inverse description, Master Detail label
- Views: view description
- Search filters: search filter description
- Lookup: lookup list heading description
- Report: report description
- Dashboard: dashboard description, labels of histogram axis, legend
- Groups: description of groups of users
- Menu: folder description, description of element inserted into the menu

Each kind of element is managed in a proper TAB of the interface.

From the localization page you can import any further localizations provided by contributors or export your own localizations you want to make available to third parties.



Element	Type	Default Translation...	Deutsch	English	Español
AdminDocument	Attribute description	Code			
AdminDocument	Attribute description	Description			
AdminDocument	Attribute description	Document date			
AdminDocument	Attribute description	Document number			
AdminDocument	Description	Administrative Doc...			
AMForm	Attribute description	Classes			
AMForm	Attribute description	Data function			
AMForm	Attribute description	Description			
AMForm	Attribute description	Name			
AMForm	Description	Asset Management...			
AMFormField	Attribute description	Description			
AMFormField	Attribute description	Form			
AMFormField	Attribute description	Index			
AMFormField	Attribute description	Metadata			
AMFormField	Attribute description	Name			
AMFormField	Attribute description	Type			
AMFormField	Description	Asset Management...			

There is the possibility of performing the following operations:

• perform a search	Search... 🔍 ✖
• import a localization in CSV format	Import
• export a localization in one or more languages in CSV format	Export
• enter in edit mode to edit the localizations	✎

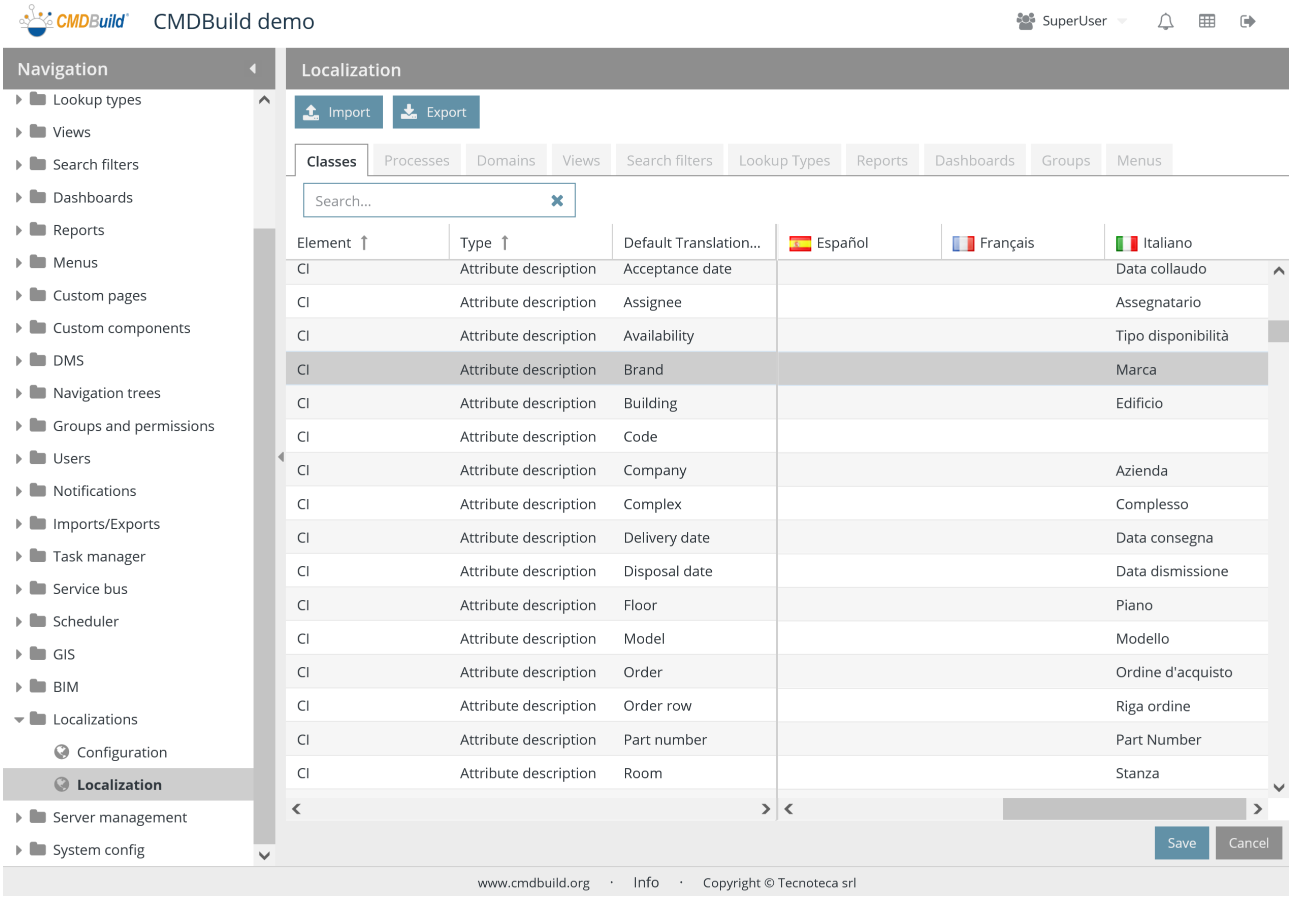
For each language enabled to the localization, there is a column where you can view, add and change the translation of that element.

29.2.1. TAB Classes

Elements to translate:

- class description
- attribute description
- description of attributes groups (fieldset)

Here's a screenshot of the user interface of classes localization.



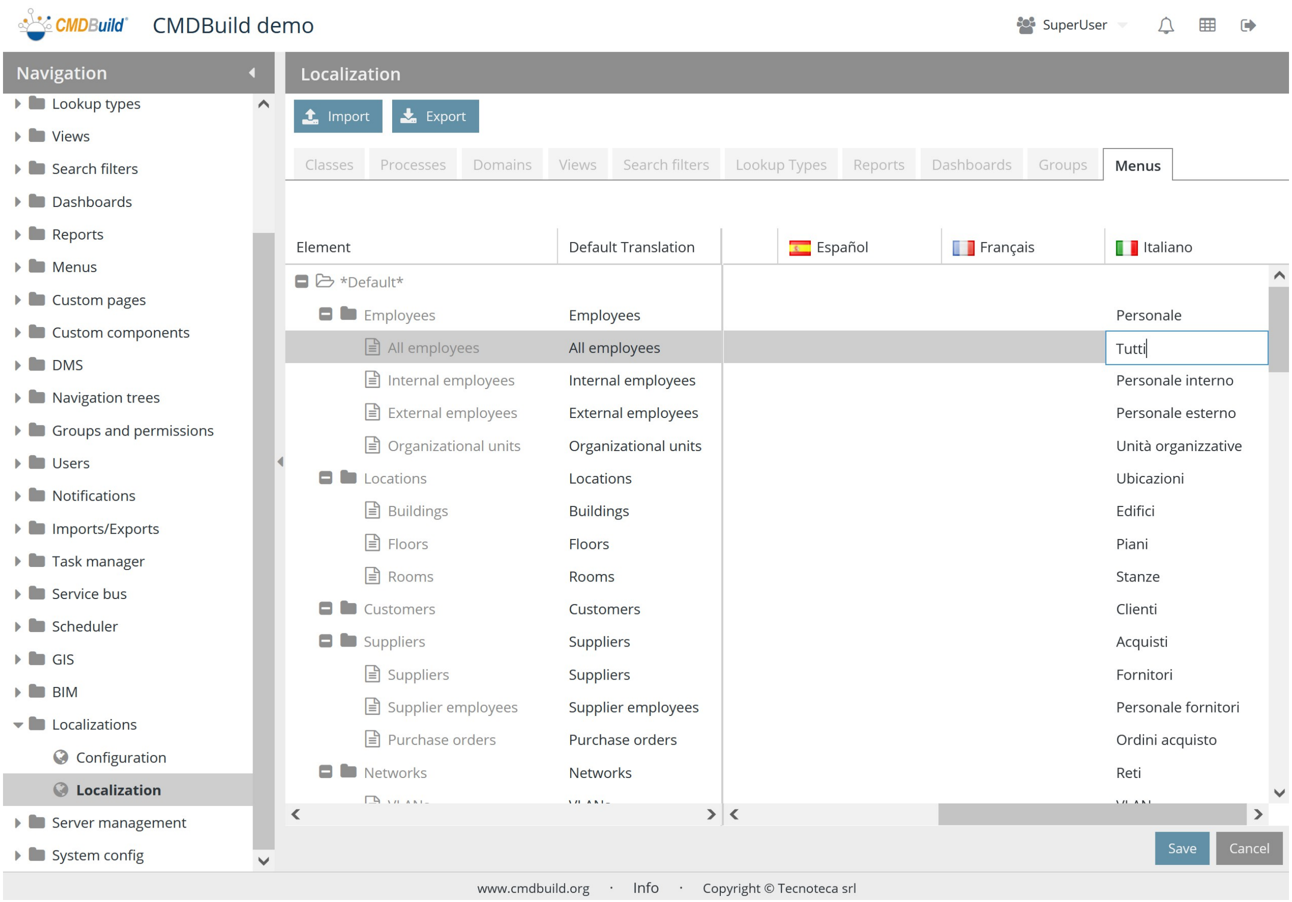
Element	Type	Default Translation...	Español	Français	Italiano
CI	Attribute description	Acceptance date			Data collaudo
CI	Attribute description	Assignee			Assegnatario
CI	Attribute description	Availability			Tipo disponibilità
CI	Attribute description	Brand			Marca
CI	Attribute description	Building			Edificio
CI	Attribute description	Code			
CI	Attribute description	Company			Azienda
CI	Attribute description	Complex			Complesso
CI	Attribute description	Delivery date			Data consegna
CI	Attribute description	Disposal date			Data dismissione
CI	Attribute description	Floor			Piano
CI	Attribute description	Model			Modello
CI	Attribute description	Order			Ordine d'acquisto
CI	Attribute description	Order row			Riga ordine
CI	Attribute description	Part number			Part Number
CI	Attribute description	Room			Stanza

The management of other tabs is similar, but the Menu tab is described in the following paragraph.

29.2.2. Menu tab

Elements to translate are the names of the single menu entries.

Here's a screenshot of the user interface of menu localization.



Element	Default Translation	Español	Français	Italiano
Default				
Employees	Employees			Personale
All employees	All employees			Tutti
Internal employees	Internal employees			Personale interno
External employees	External employees			Personale esterno
Organizational units	Organizational units			Unità organizzative
Locations	Locations			Ubicazioni
Buildings	Buildings			Edifici
Floors	Floors			Piani
Rooms	Rooms			Stanze
Customers	Customers			Clienti
Suppliers	Suppliers			Acquisti
Suppliers	Suppliers			Fornitori
Supplier employees	Supplier employees			Personale fornitori
Purchase orders	Purchase orders			Ordini acquisto
Networks	Networks			Reti

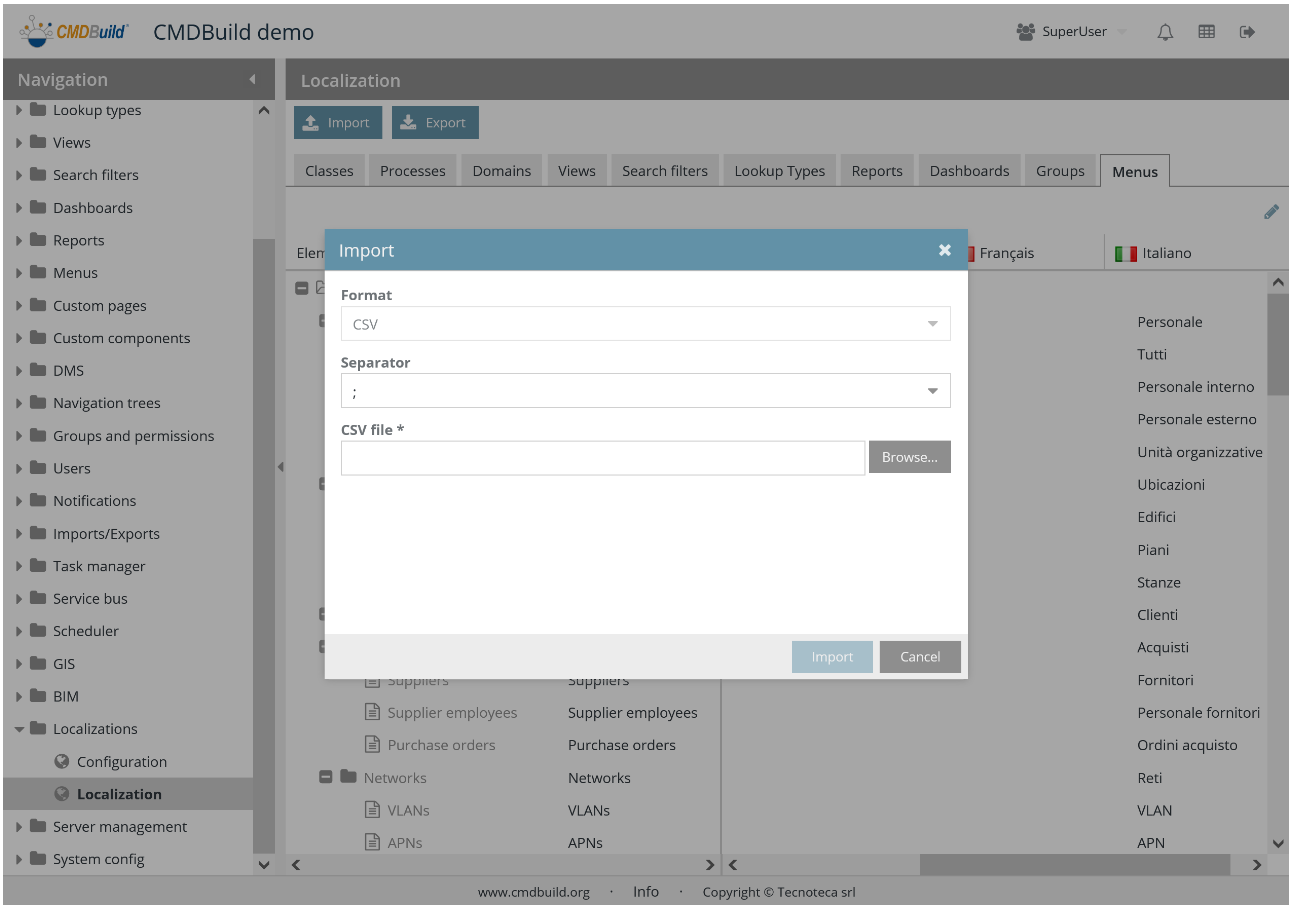
29.3. Import

The import feature allows you to import a localization provided by Tecnoteca or other contributors.

In the page from where you can import there are the following fields:

- Format: CSV (not-editable)
- Separator: it can be selected among the symbols “,” or “;” or “|”
- File

Here's a sample of interface screenshot.



Import

Format: CSV

Separator: ;

CSV file *

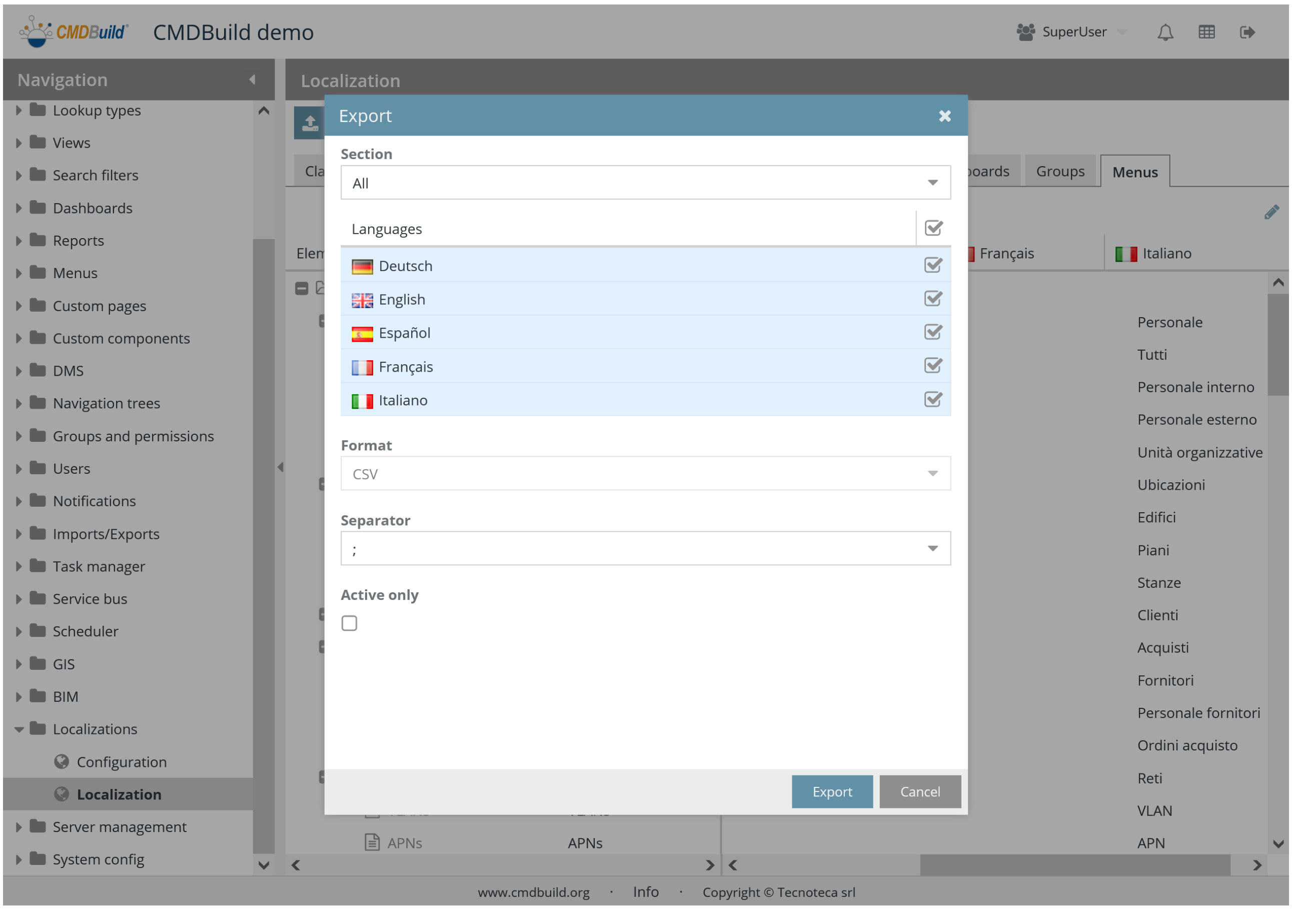
29.4. Export

The export feature allows you to export your own localization that you want to make available to other users.

In the page from where you can export there are the following fields:

- Section: you can choose among All, Classes, Processes, Domains, Views, Search filters, Lookups, Reports, Menu
- Languages: through "check" you can select the languages you want to export (among the enabled ones)
- Format: CSV (not-editable)
- Separator: it can be selected among the symbols “,” or “;” or “|”
- Only active: it allows you to export only localizations of active elements

Here's a sample of interface screenshot.



Export

Section: All

Languages ☑

Deutsch	☑
English	☑
Español	☑
Français	☑
Italiano	☑

Format: CSV

Separator: ;

Active only □

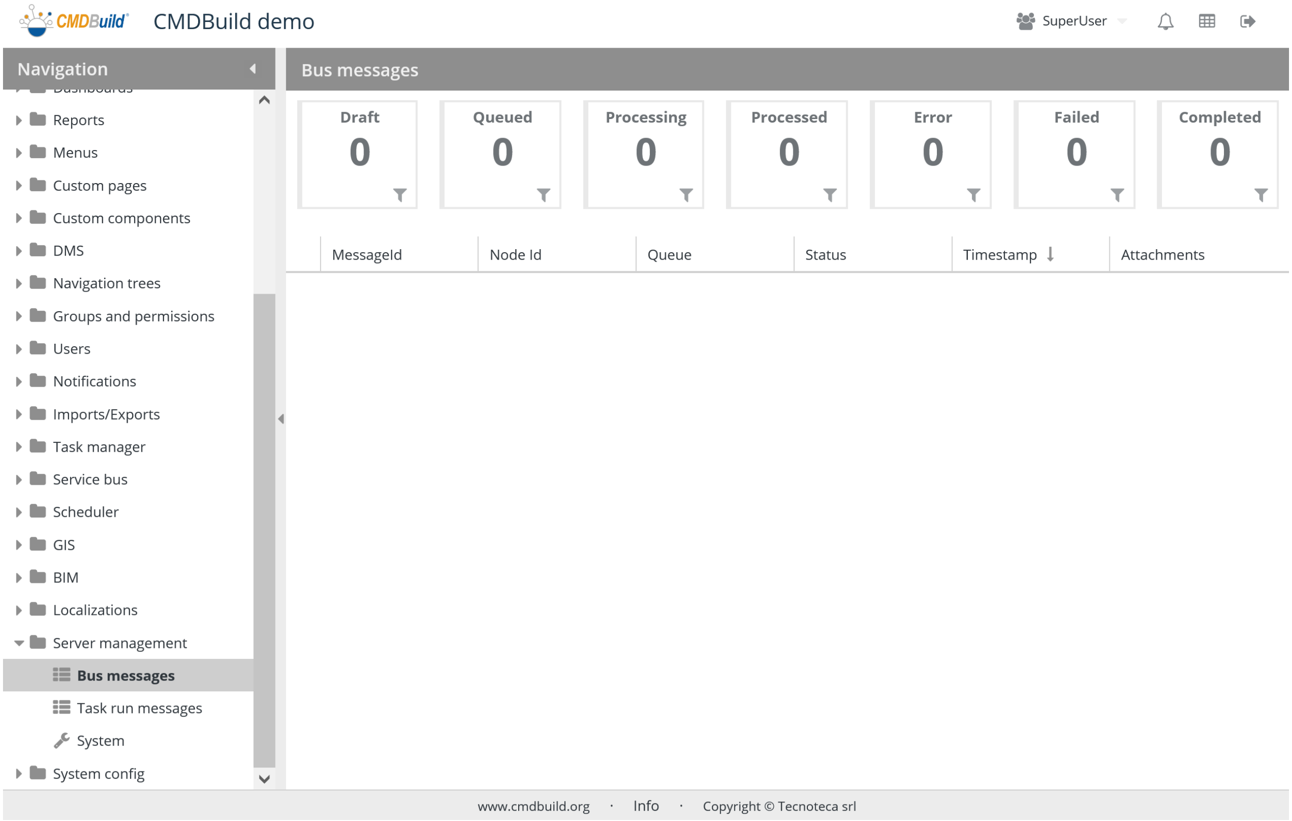
30. Server configuration

CMDBuild server configuration pages allow you to keep controlled some aspects of processing in progress and active services.

30.1. Bus messages

This function allows to display messages exchanged among services through the CMDBuild Service Bus, summarized on the top.

In the grid you can see the list of messages with details.



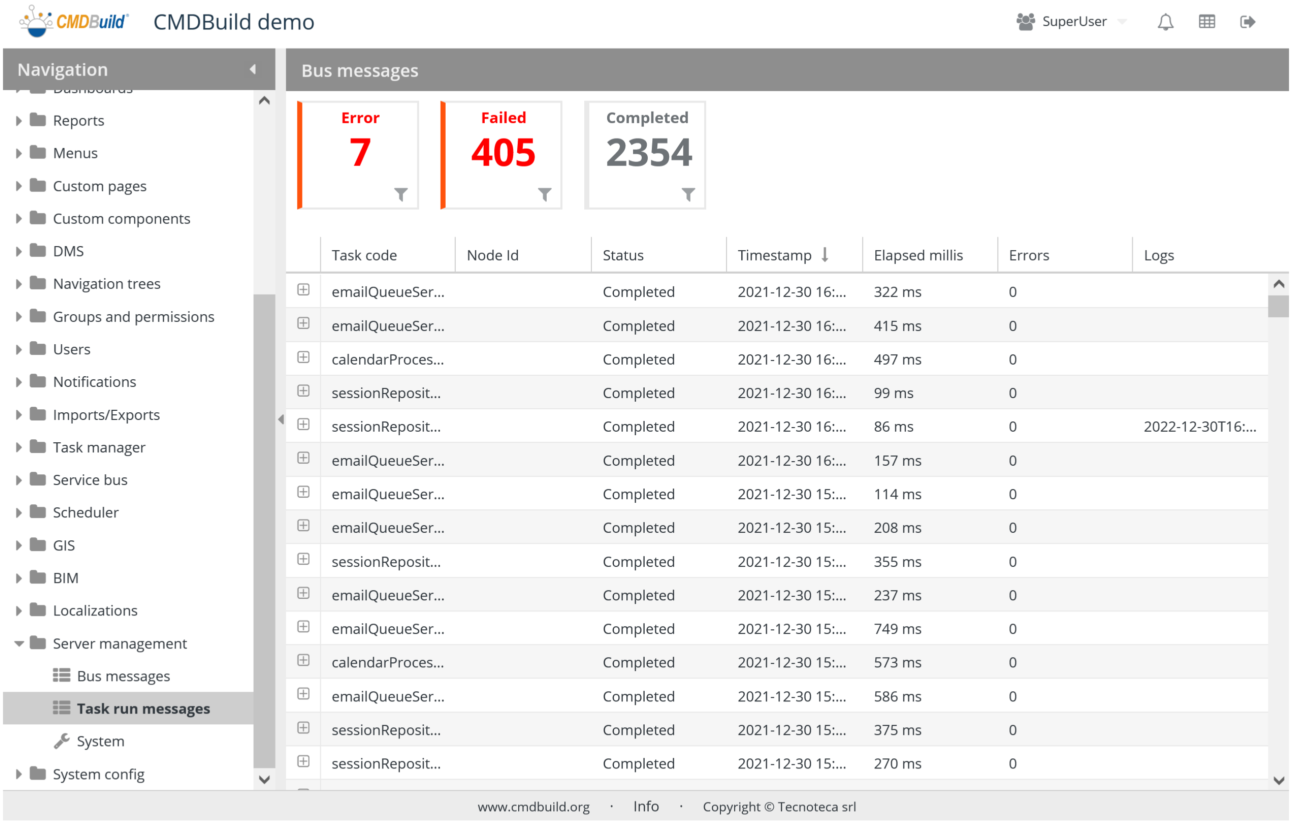
Draft	Queued	Processing	Processed	Error	Failed	Completed
0	0	0	0	0	0	0

MessageId	Node Id	Queue	Status	Timestamp	Attachments

30.2. Messages of running tasks

This function allows to display messages produced by tasks running in the system, summarized on the top.

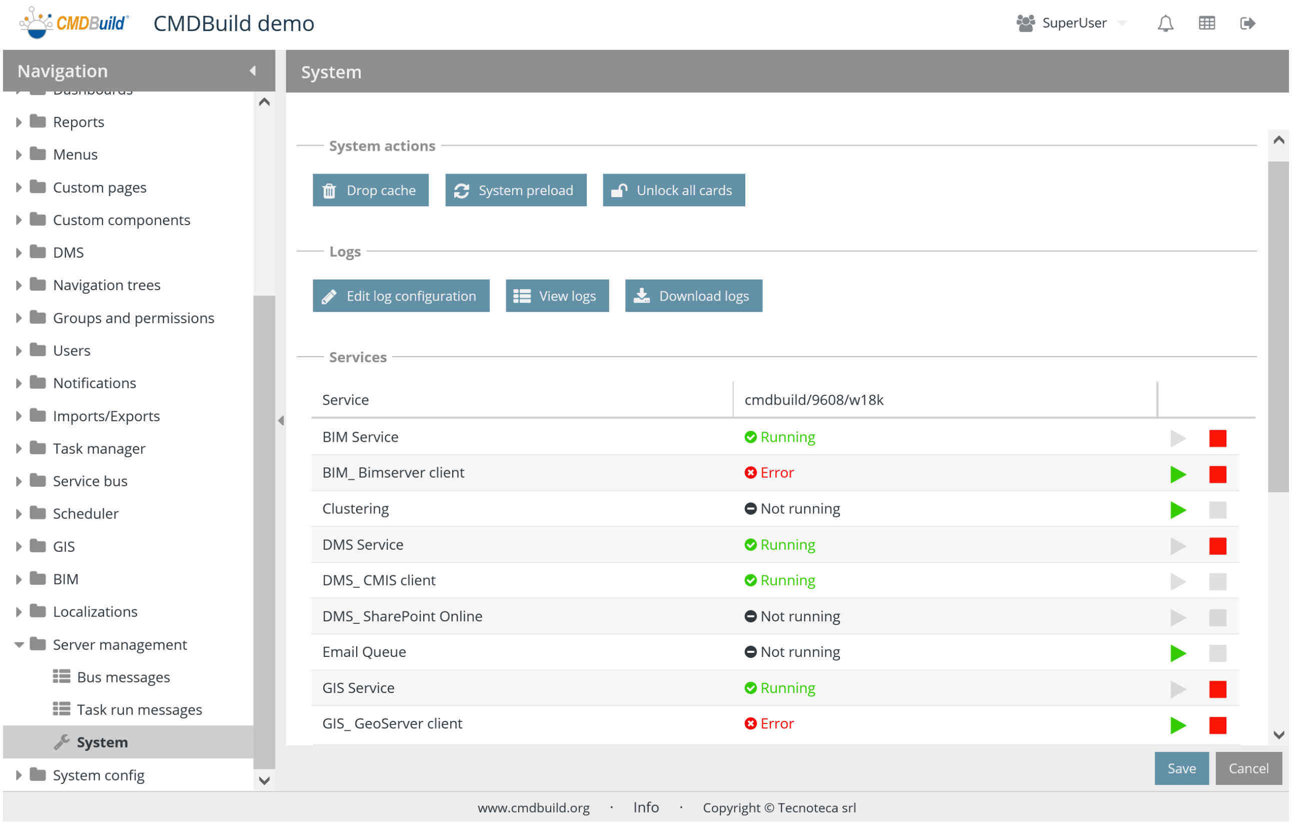
In the grid you can see the list of messages with details.



Task code	Node Id	Status	Timestamp	Elapsed millis	Errors	Logs
emailQueueSer...		Completed	2021-12-30 16:...	322 ms	0	
emailQueueSer...		Completed	2021-12-30 16:...	415 ms	0	
calendarProces...		Completed	2021-12-30 16:...	497 ms	0	
sessionReposit...		Completed	2021-12-30 16:...	99 ms	0	
sessionReposit...		Completed	2021-12-30 16:...	86 ms	0	2022-12-30T16:...
emailQueueSer...		Completed	2021-12-30 16:...	157 ms	0	
emailQueueSer...		Completed	2021-12-30 15:...	114 ms	0	
emailQueueSer...		Completed	2021-12-30 15:...	208 ms	0	
sessionReposit...		Completed	2021-12-30 15:...	355 ms	0	
emailQueueSer...		Completed	2021-12-30 15:...	237 ms	0	
emailQueueSer...		Completed	2021-12-30 15:...	749 ms	0	
calendarProces...		Completed	2021-12-30 15:...	573 ms	0	
emailQueueSer...		Completed	2021-12-30 15:...	586 ms	0	
sessionReposit...		Completed	2021-12-30 15:...	375 ms	0	
sessionReposit...		Completed	2021-12-30 15:...	270 ms	0	

30.3. Server status

It allows to display the server status and run some system operations.

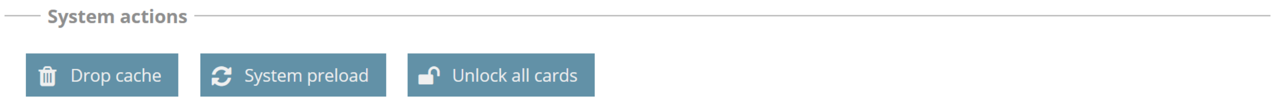


Service	cmdbuild/9608/w18k		
BIM Service	Running	▶	■
BIM_ Bimserver client	Error	▶	■
Clustering	Not running	▶	■
DMS Service	Running	▶	■
DMS_ CMIS client	Running	▶	■
DMS_ SharePoint Online	Not running	▶	■
Email Queue	Not running	▶	■
GIS Service	Running	▶	■
GIS_ GeoServer client	Error	▶	■

30.3.1. System actions

It allows you to access some buttons in order to perform service functions:

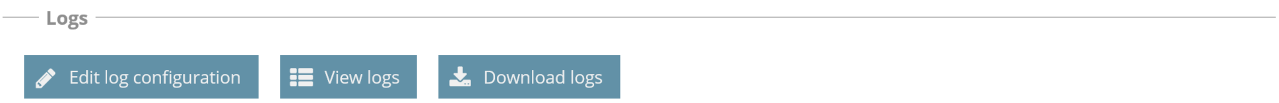
- Drop cache: used to clear CMDBuild cache; useful when elements of the PostgreSQL database are modified outside the application (LookUp rows, database views, system functions, etc.) and you don't want to restart Tomcat to refresh
- System preload: it can be used to load the cache information after a cancellation
- Unlock all cards: used to unlock the locked cards and processes. This is useful to anticipate the lock release



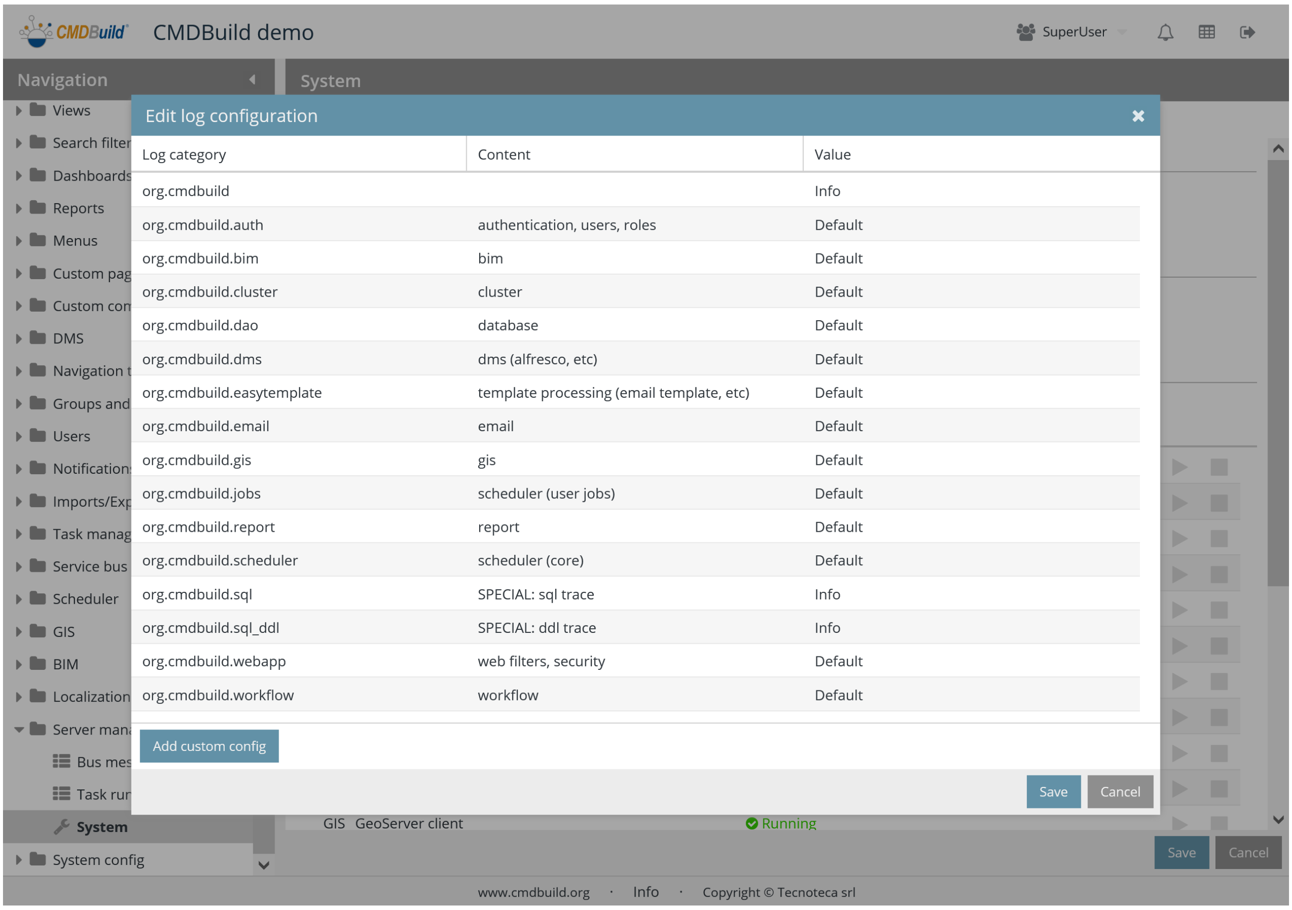
30.3.2. Log management

It allows you to perform the following operations on log files:

- Edit log configurations: allows to view and edit the file log configurations
- View log: allows to view CMDBuild log file
- Download log: allows to download several log files used by CMDBuild

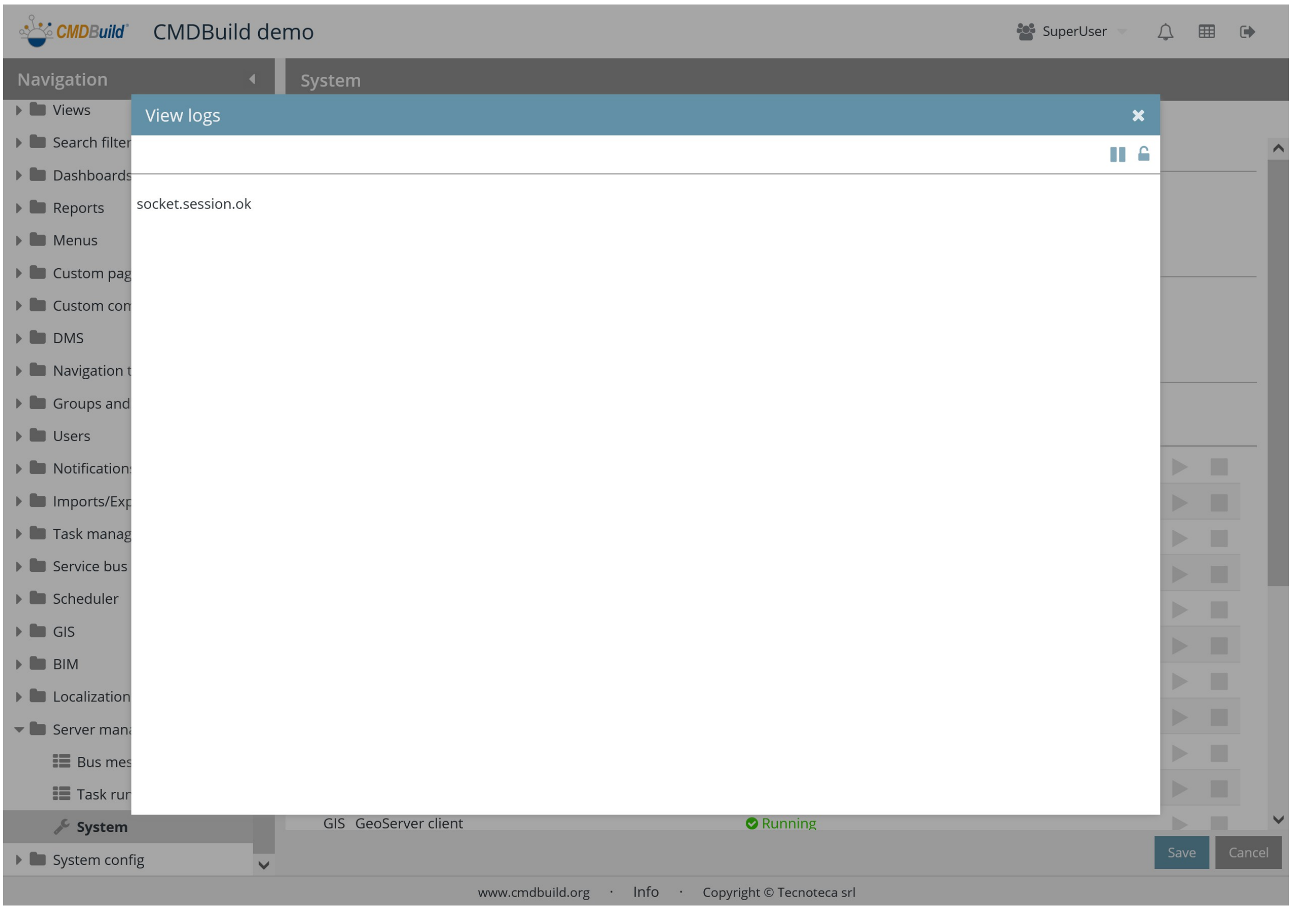


The button "Edit log configurations" opens the following popup window:



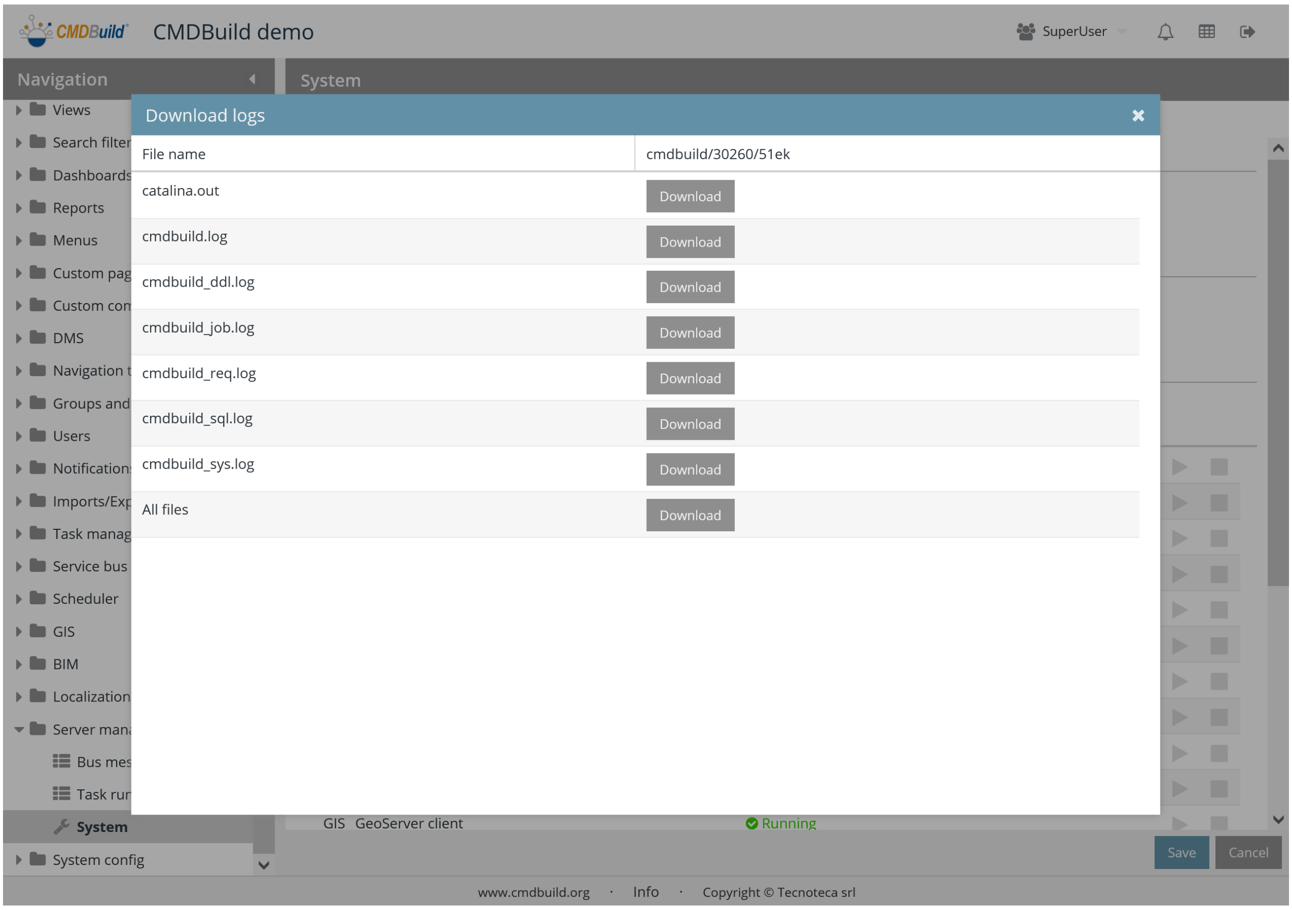
Log category	Content	Value
org.cmdbuild		Info
org.cmdbuild.auth	authentication, users, roles	Default
org.cmdbuild.bim	bim	Default
org.cmdbuild.cluster	cluster	Default
org.cmdbuild.dao	database	Default
org.cmdbuild.dms	dms (alfresco, etc)	Default
org.cmdbuild.easytemplate	template processing (email template, etc)	Default
org.cmdbuild.email	email	Default
org.cmdbuild.gis	gis	Default
org.cmdbuild.jobs	scheduler (user jobs)	Default
org.cmdbuild.report	report	Default
org.cmdbuild.scheduler	scheduler (core)	Default
org.cmdbuild.sql	SPECIAL: sql trace	Info
org.cmdbuild.sql_ddl	SPECIAL: ddl trace	Info
org.cmdbuild.webapp	web filters, security	Default
org.cmdbuild.workflow	workflow	Default

The button "View log" opens the following popup window:



```
socket.session.ok
```

The button "Download log"opens the following popup window:

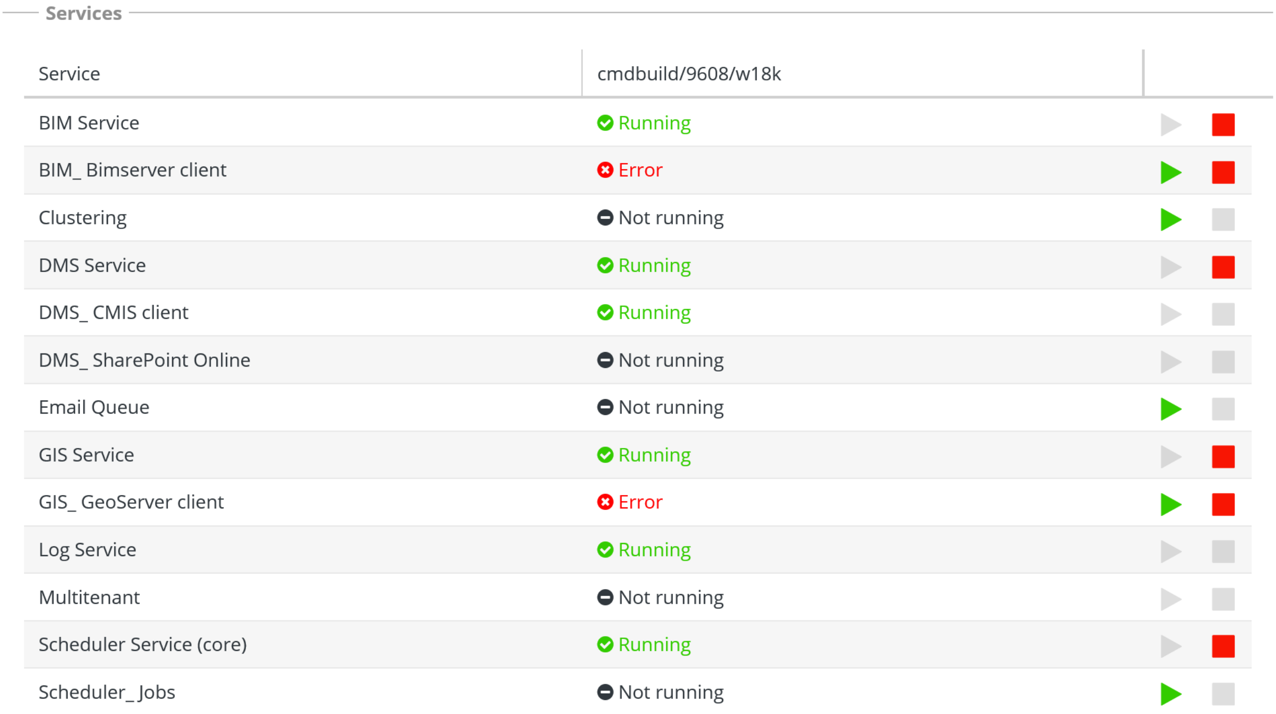


Download logs

File name	cmdbuild/30260/51ek
catalina.out	Download
cmdbuild.log	Download
cmdbuild_ddl.log	Download
cmdbuild_job.log	Download
cmdbuild_req.log	Download
cmdbuild_sql.log	Download
cmdbuild_sys.log	Download
All files	Download

30.3.3. Service management

It displays the current status of the services used by CMDBuild and allows the interruptions for some of them.



Services

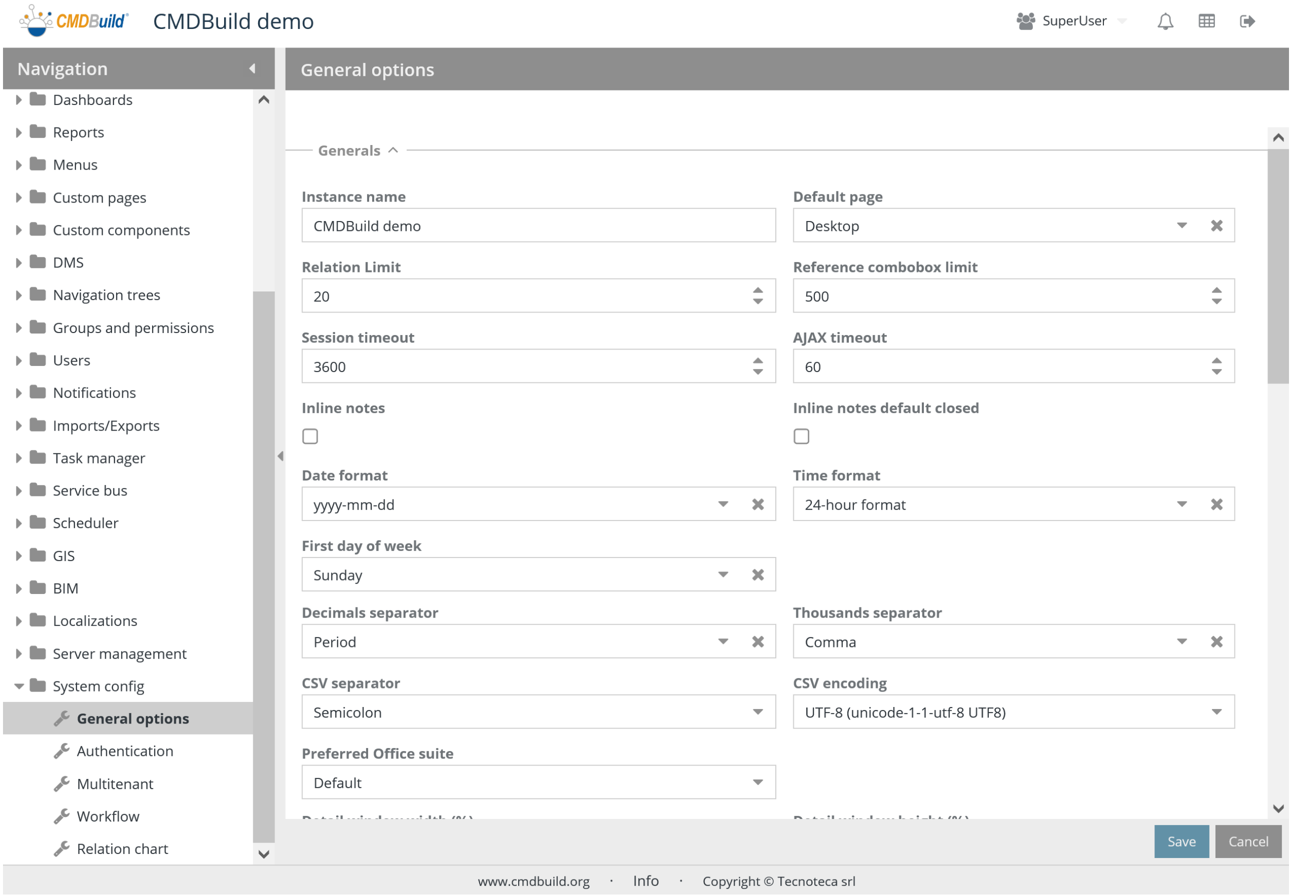
Service	cmdbuild/9608/w18k		
BIM Service	✓ Running	▷	■
BIM_ Bimserver client	✖ Error	▶	■
Clustering	⊖ Not running	▶	□
DMS Service	✓ Running	▷	■
DMS_ CMIS client	✓ Running	▷	□
DMS_ SharePoint Online	⊖ Not running	▷	□
Email Queue	⊖ Not running	▶	□
GIS Service	✓ Running	▷	■
GIS_ GeoServer client	✖ Error	▶	■
Log Service	✓ Running	▷	□
Multitenant	⊖ Not running	▷	□
Scheduler Service (core)	✓ Running	▷	■
Scheduler_ Jobs	⊖ Not running	▶	□

31. System configuration

The CMDBuild Administration Module includes some final pages that the administrator can use to set some configuration parameters of the system.

31.1. General options

This function allows you to set some general configuration parameters of CMDBuild.



Instance name: CMDBuild demo

Default page: Desktop

Relation Limit: 20

Reference combobox limit: 500

Session timeout: 3600

AJAX timeout: 60

Inline notes: ☐

Inline notes default closed: ☐

Date format: yyyy-mm-dd

Time format: 24-hour format

First day of week: Sunday

Decimals separator: Period

Thousands separator: Comma

CSV separator: Semicolon

CSV encoding: UTF-8 (unicode-1-1-utf-8 UTF8)

Preferred Office suite: Default

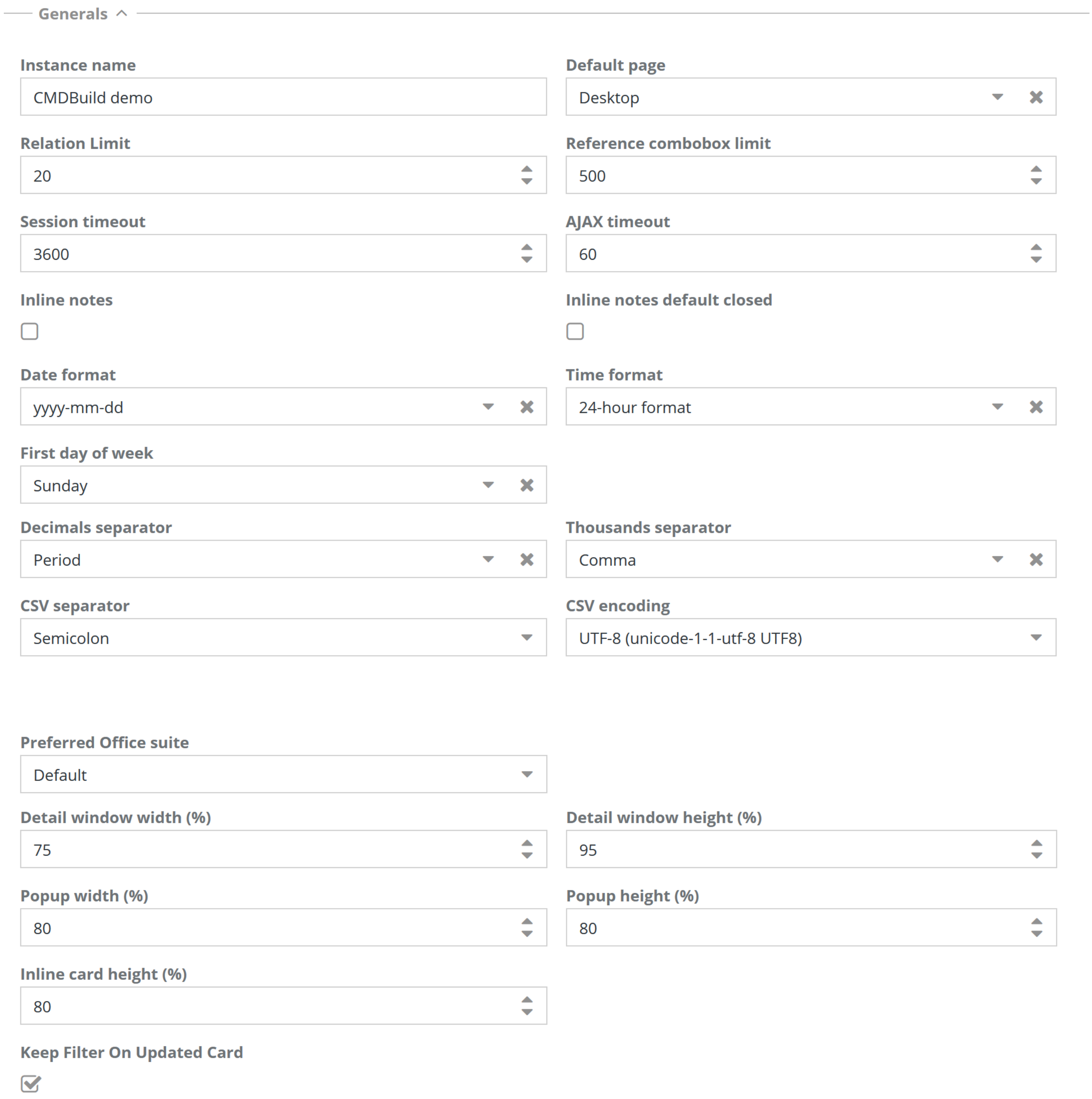
Parameters are divided into the two sections described below.

31.1.1. General properties

The following information is required in order to be used as default settings (some information can be reset in the user preferences) for the current instance:

- Instance name: the string is displayed up on the left near the application logo
- Starting page: default opening page
- Relation limit: it indicates the maximum number of relations shown separately; if exceeded, the relations are grouped together

- References and lookup combobox limit: it allows you to specify the maximum number of options shown in a combobox; if exceeded, the options are displayed in a popup window with pagination and filters
- Sessions duration: it allows you to specify the maximum inactivity time before the session expires (in seconds)
- AJAX timeout: it allows you to specify the timeout for the requests done from the user interface to the server through REST webservice (in seconds)
- Inline notes: it states that the management tab of notes has to be removed and managed as additional "fieldset" on the main page of each card (unless otherwise indicated on the single classes)
- Closed inline notes: it states that in case the previous options "fieldset" has been selected, it has to appear as closed by default when the page opens
- Date format: among the available ones
- Hour format: among the available ones
- First day of the week
- Decimal separator in the numeric fields: among the available ones (customizable by each user)
- Thousands separator in the numeric fields: among the available ones (customizable by each user)
- CSV separator: “;”, “|” or TAB
- CSV encoding: among the available ones
- Used Office software: used in the Export function and in the reports in CSV format for an optimal formatting of dates and numbers, the default one is Microsoft Office (customizable by each user
- Detail window width (%): it allows you to specify in percent the width of the editing window of the data cards of the available amount
- Detail window height (%): it allows you to specify in percent the height of the editing window of the data cards of the available amount
- Popup width (%): it allows you to specify in percent the width of the popup windows of the available amount
- Popup height (%): it allows you to specify in percent the height of the popup windows of the available amount
- Inline card height (%): it allows you to specify in percent the height of the space reserved to the inline display of a card within the grid of a class / process
- Keep FIlter On Updated Card: it allows you to specify if, after you applied a filter that excludes the current data card, the filter is disabled or the current card is hidden



Generals

Instance name: CMDBuild demo

Default page: Desktop

Relation Limit: 20

Reference combobox limit: 500

Session timeout: 3600

AJAX timeout: 60

Inline notes: □

Inline notes default closed: □

Date format: yyyy-mm-dd

Time format: 24-hour format

First day of week: Sunday

Decimals separator: Period

Thousands separator: Comma

CSV separator: Semicolon

CSV encoding: UTF-8 (unicode-1-1-utf-8 UTF8)

Preferred Office suite: Default

Detail window width (%): 75

Detail window height (%): 95

Popup width (%): 80

Popup height (%): 80

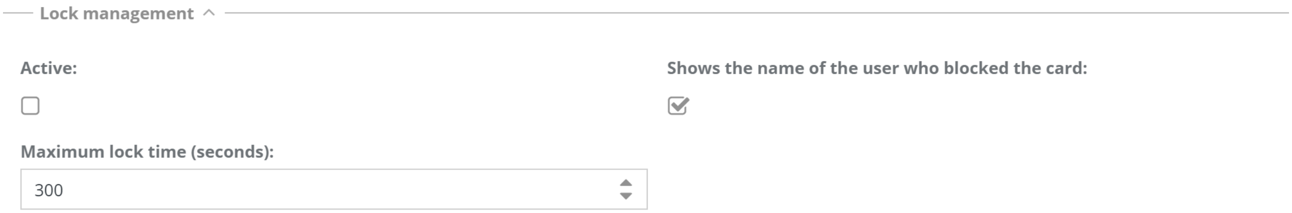
Inline card height (%): 80

Keep Filter On Updated Card: ☑

31.1.2. Lock management

The following information is required:

- Active: it allows you to enable (default) or disabled the lock management of those data cards and processes in edit by another user
- Show the name of the user who locked the card: it allows you to exclude such information (possible privacy needs) from the message shown by the system when asking for the editing of a card reserved for that purpose by another operator
- Lock time-limit: it allows you to specify when the card can be edited again by an operator



Lock management

Active:

□

Shows the name of the user who blocked the card:

☑

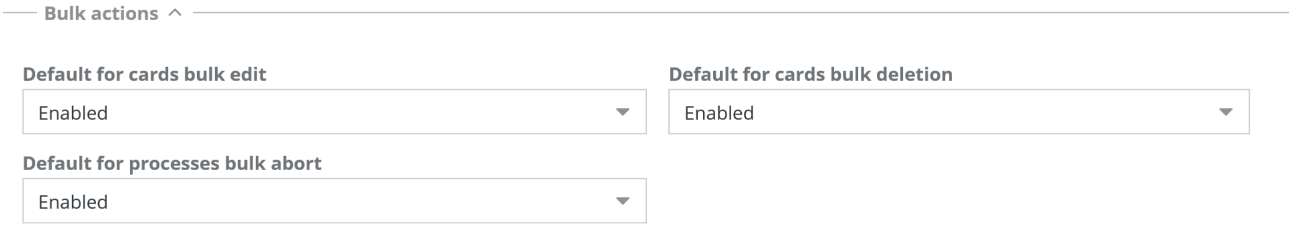
Maximum lock time (seconds):

300

31.1.3. Massive actions

The following information is required:

- Massive modification default: allows to enable or disable by default the action for all classes (except for specific configurations on some of them)
- Massive delete default: allows to enable or disable by default the action for all classes (except for specific configurations on some of them)
- Massive abort process default: allows to enable or disable by default the action for all classes (except for specific configurations on some of them)



Bulk actions

Default for cards bulk edit: Enabled

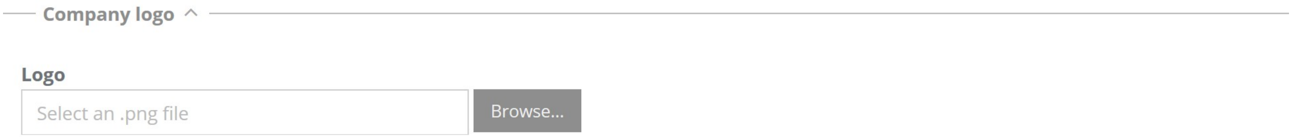
Default for cards bulk deletion: Enabled

Default for processes bulk abort: Enabled

31.1.4. Company logo

The following information is required:

- Logo: it allows you to upload the company logo. It will then appear in the header row of the user interface, after the CMDBuild logo



Company logo

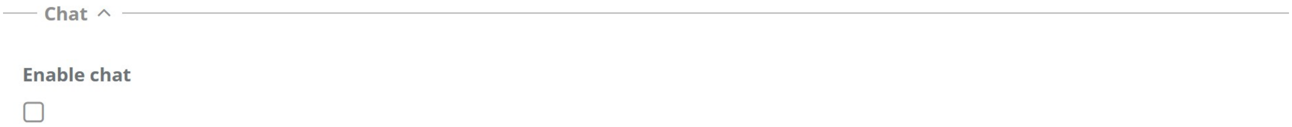
Logo

Select an .png file

31.1.5. Chat

The following information is required:

- It enables the CMDBuild chat



Chat

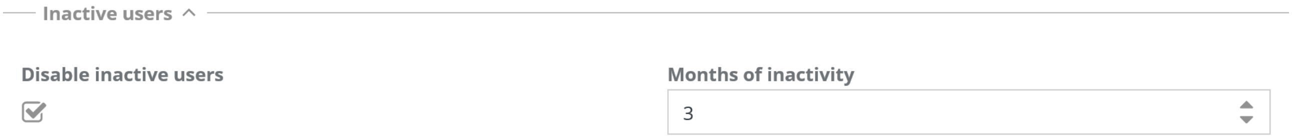
Enable chat

□

31.1.6. Inactive users

The following information is required:

- Disable inactive users: allows to set automatically as inactive all the users who didn't connect to the system for a number of months
- Inactivity months: parameter linked to the previous explanation



Inactive users

Disable inactive users

☑

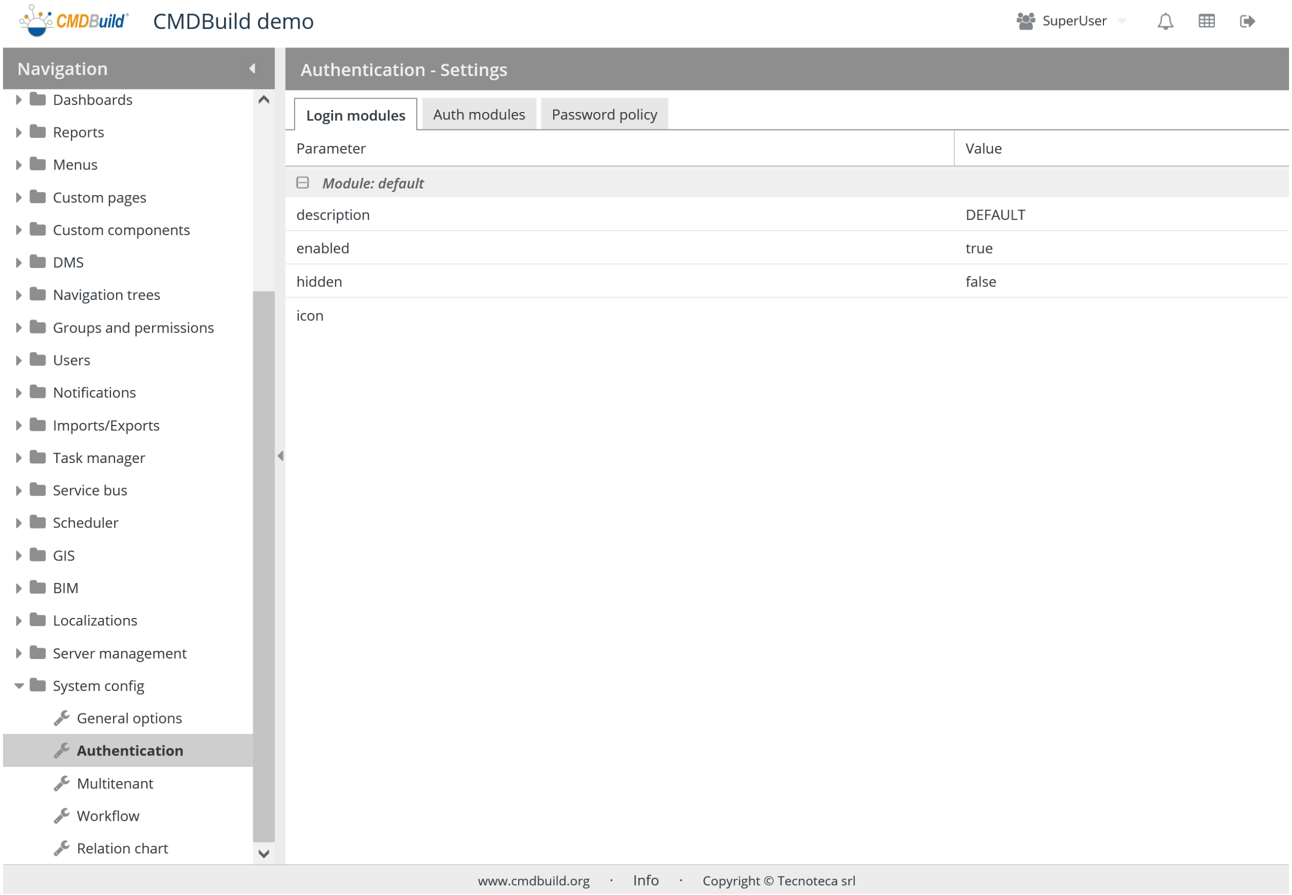
Months of inactivity

3

31.2. Authentication

31.2.1. Login modules

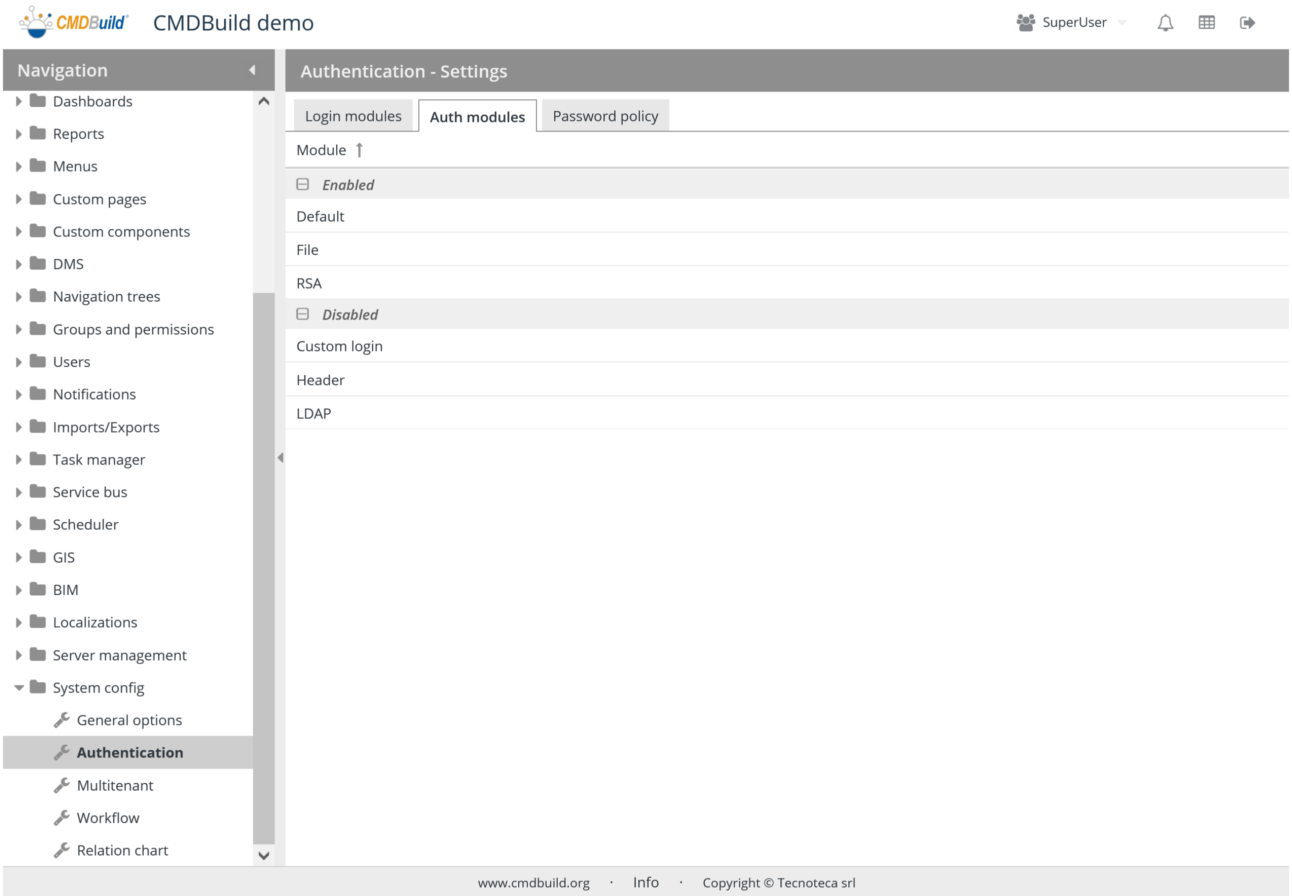
Active login modules are displayed with the related parameter value.



Parameter	Value
Module: default	
description	DEFAULT
enabled	true
hidden	false
icon	

31.2.2. Authentication modules

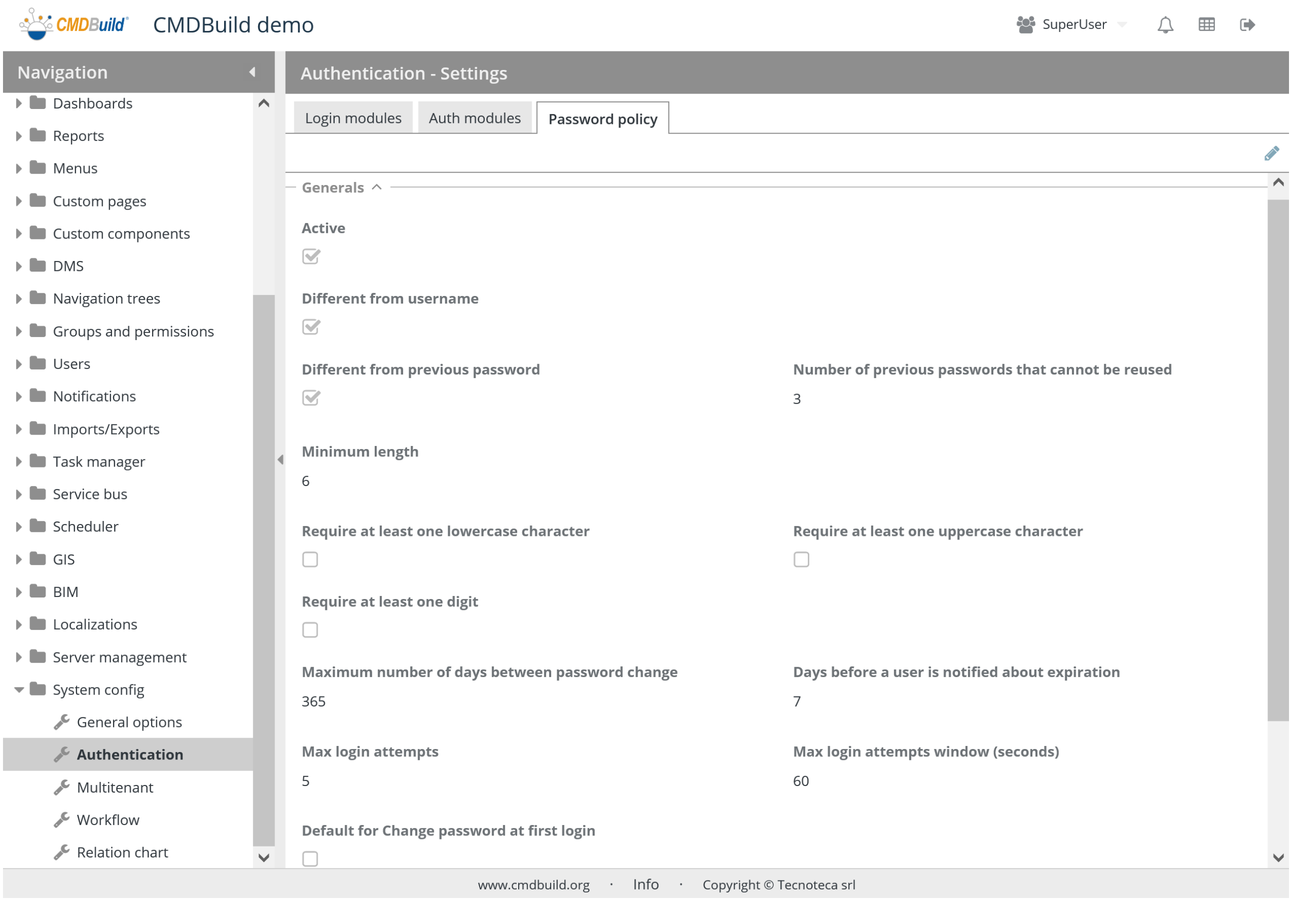
Active authentication modules are displayed.



Module
Enabled
Default
File
RSA
Disabled
Custom login
Header
LDAP

31.2.3. Password policy

This function allows you to set the criteria used by users to choose their password, according to the company's security policies.



Active: ☑

Different from username: ☑

Different from previous password: ☑

Number of previous passwords that cannot be reused: 3

Minimum length: 6

Require at least one lowercase character: □

Require at least one uppercase character: □

Require at least one digit: □

Maximum number of days between password change: 365

Days before a user is notified about expiration: 7

Max login attempts: 5

Max login attempts window (seconds): 60

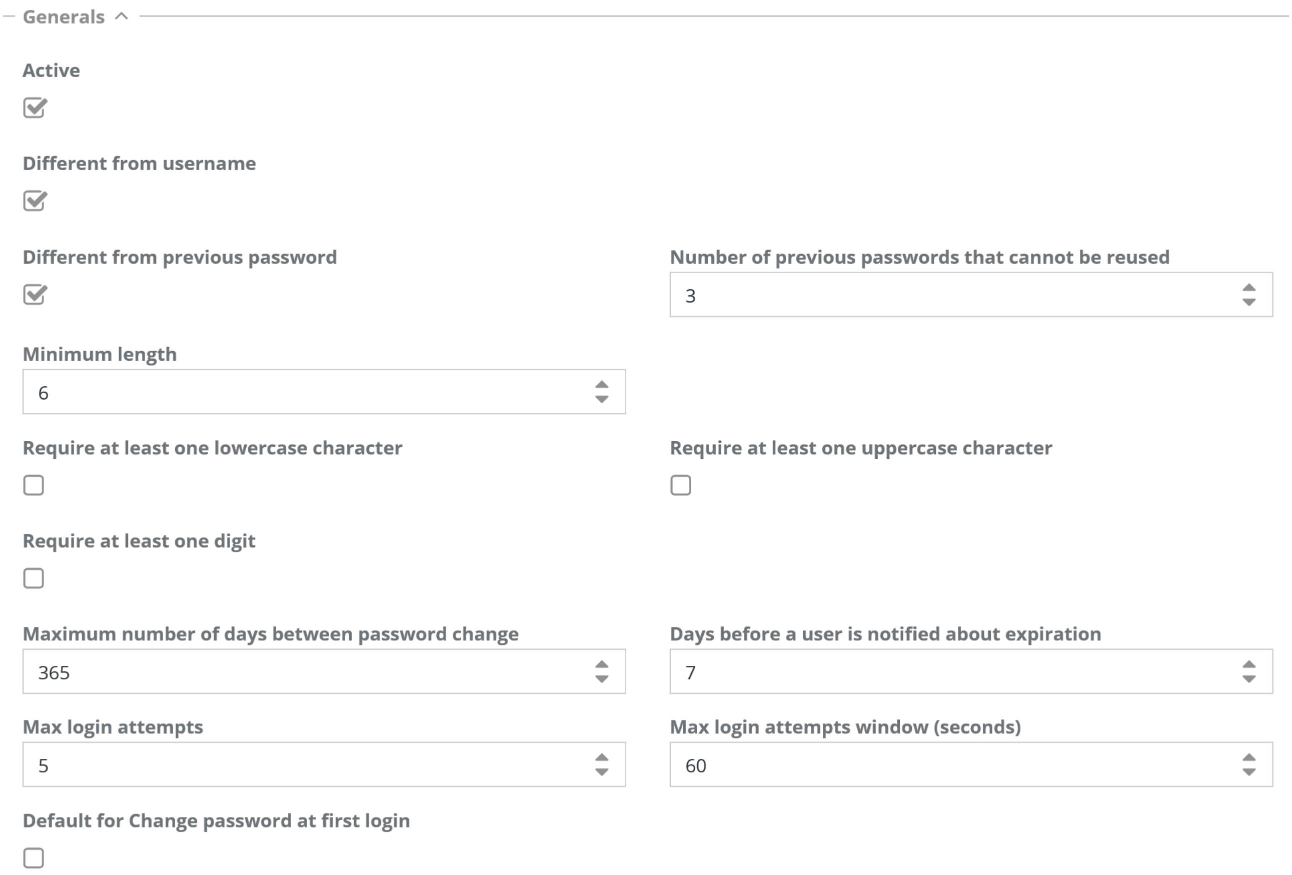
Default for Change password at first login: □

31.2.4. General properties

The following information is required:

- Active: it allows you to activate the rules for password choice
- Different from username: it prevents you from choosing a password which is identical to the username
- Different from previous password: it prevents you from choosing a new password which is identical to the previous one(s). The number of old passwords is shown below
- Number of previous passwords that cannot be reused
- Minimum length: minimum amount of characters
- Require at least one lowercase character
- Require at least one uppercase character
- Require at least one digit
- Minimum number of days between password change
- Days before a user is notified about expiration
- Maximum access attempts: after the maximum number of failed logins the user is blocked for 10 minutes
- Time window of maximum access attempts: related to the previous parameter

- First access password change default: you can assign the first access password change as mandatory



Generals

Active
☑

Different from username
☑

Different from previous password
☑

Number of previous passwords that cannot be reused
3

Minimum length
6

Require at least one lowercase character
□

Require at least one uppercase character
□

Require at least one digit
□

Maximum number of days between password change
365

Days before a user is notified about expiration
7

Max login attempts
5

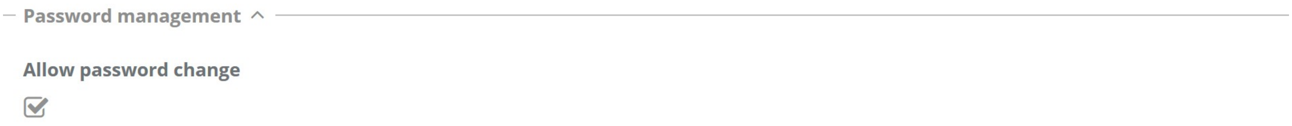
Max login attempts window (seconds)
60

Default for Change password at first login
□

31.2.5. Password management

The following information is required:

- Allow password change: enabled by default, it allows the users to modify their passwords on CMDBuild (usually this is not allowed in case of authentication on external systems)

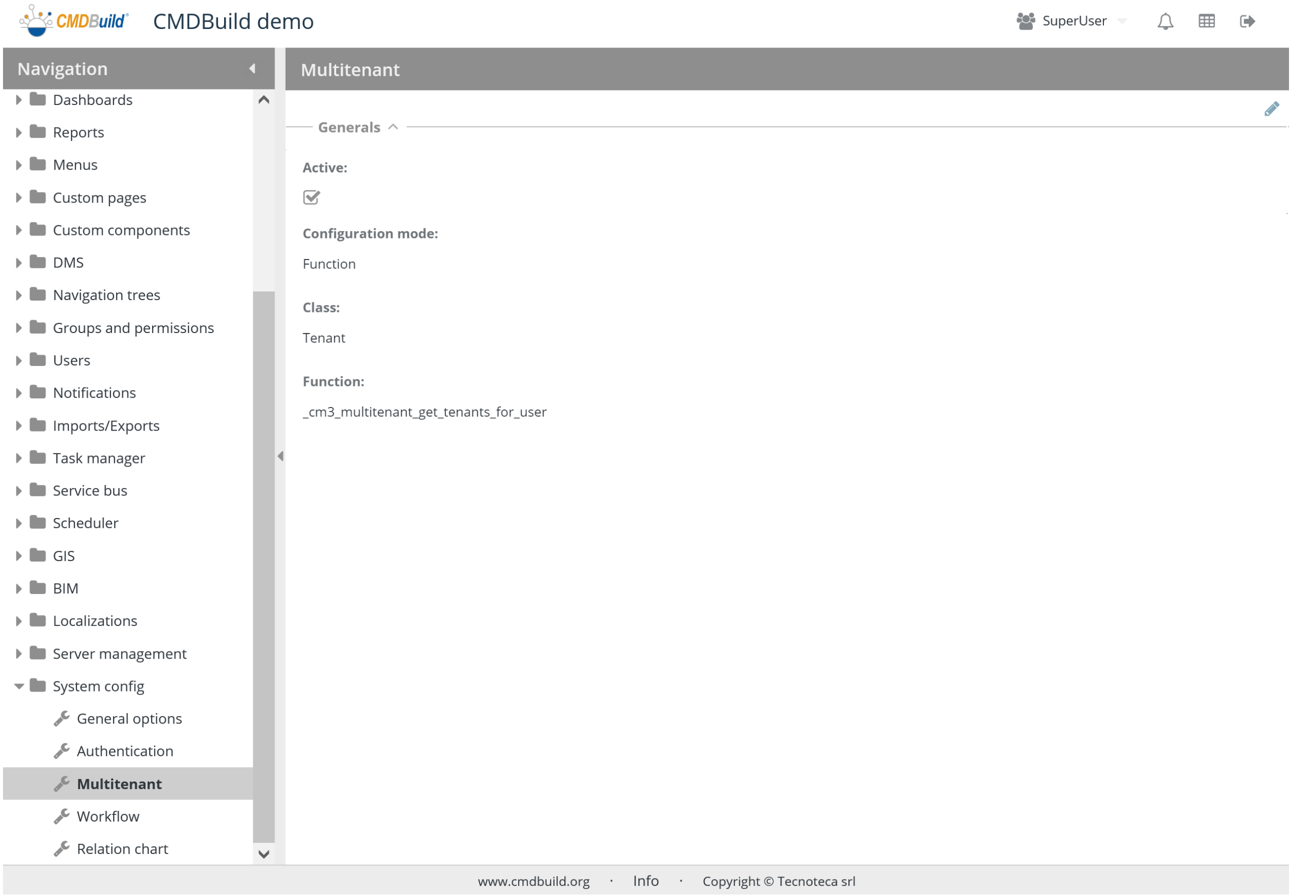


Password management

Allow password change
☑

31.3. Multitenant

This function allows you to set some configuration parameters of the CMDBuild multitenant system.



Active:

☑

Configuration mode:

Function

Class:

Tenant

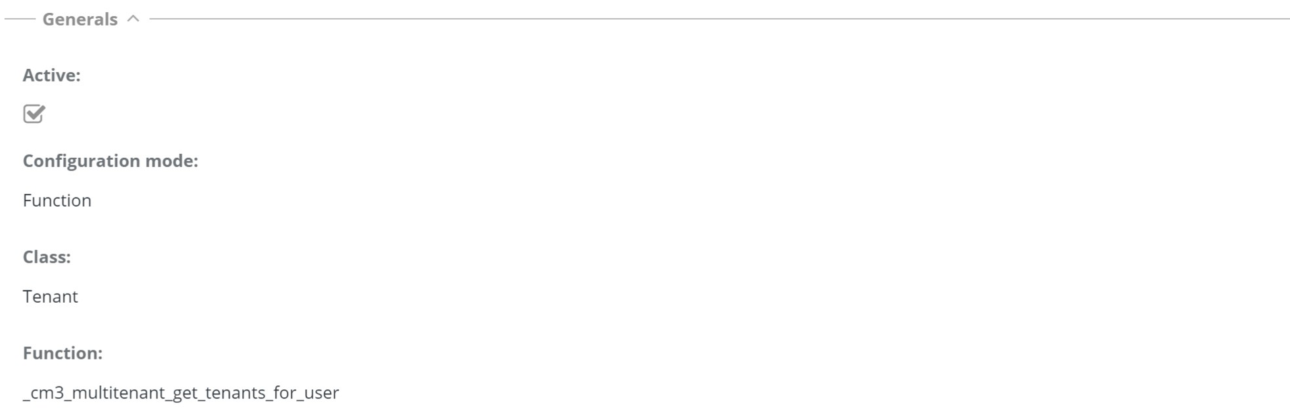
Function:

_cm3_multitenant_get_tenants_for_user

31.3.1. General properties

The following information is required:

- Active: it allows you to activate the multitenant mode in CMDBuild (see below)
- Configuration modes:
  - Class: the cards of an application class are used as a list of tenants (Company, Site, Organizational Unit, etc.)
  - Function: the values returned by a PostgreSQL database function are used as tenant list
- Class: it allows you to select the tenant root class, in case you choose the first option in the previous parameter
- Function: it allows you to select the root database function of tenants, in case you choose the second option in the previous parameter



Active:

☑

Configuration mode:

Function

Class:

Tenant

Function:

_cm3_multitenant_get_tenants_for_user

Important:

The multitenant configuration can be activated and set only once.

Since each card is marked with the proprietary tenant, to edit the configuration of the multitenant management implicates that you should work on all rows of all CMDBuild tables, and this cannot be done automatically from the user interface of the Administration Module.

Therefore it is important to configure only in case you are sure that you have identified the correct configuration criteria.

Example:

Here's a sample of configuration of a PostgreSQL function, which can be used in the configuration of the Multitenant.

How to configure

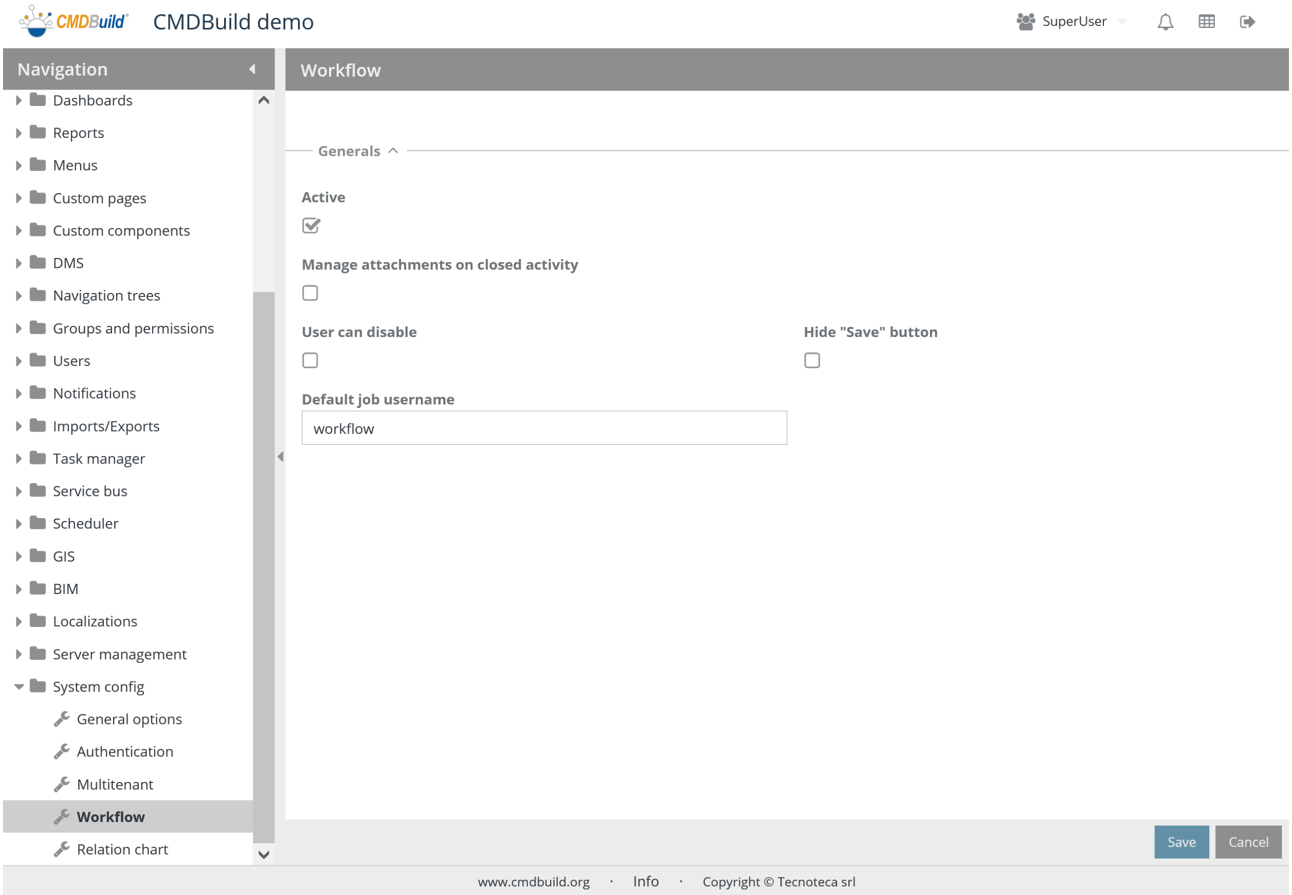
This is an example of _cm3_multitenant_get function; it takes a single bigint param, which can be:

    * -1 : in this case the function should return all available tenant ids;
    * any other value : in this case the function should return all available tenants for supplied user id param;

```
CREATE
OR REPLACE FUNCTION _cm3_multitenant_get (_user_id bigint) RETURNS SETOF bigint AS $$
BEGIN
  IF _user_id = - 1
THEN
  RETURN QUERY
  SELECT
    "Tenant"."Id"
  FROM
    "Tenant"
  WHERE
    "Tenant"."Status" = 'A';
ELSE
  RETURN QUERY
  SELECT
    "Tenant"."Id"
  FROM
    "Tenant"
    JOIN
      "Map_UserTenant"
      ON "Tenant"."Id" = "Map_UserTenant"."IdObj2"
  WHERE
    "Map_UserTenant"."IdObj1" = _user_id
    AND "Tenant"."Status" = 'A'
    AND "Map_UserTenant"."Status" = 'A';
END
IF;
END
$$ LANGUAGE PLPGSQL;
```

31.4. Workflow

This function allows you to set some configuration parameters of the CMDBuild workflow engine.



Workflow

Generals

Active

☑

Manage attachments on closed activity

□

User can disable

□

Hide "Save" button

□

Default job username

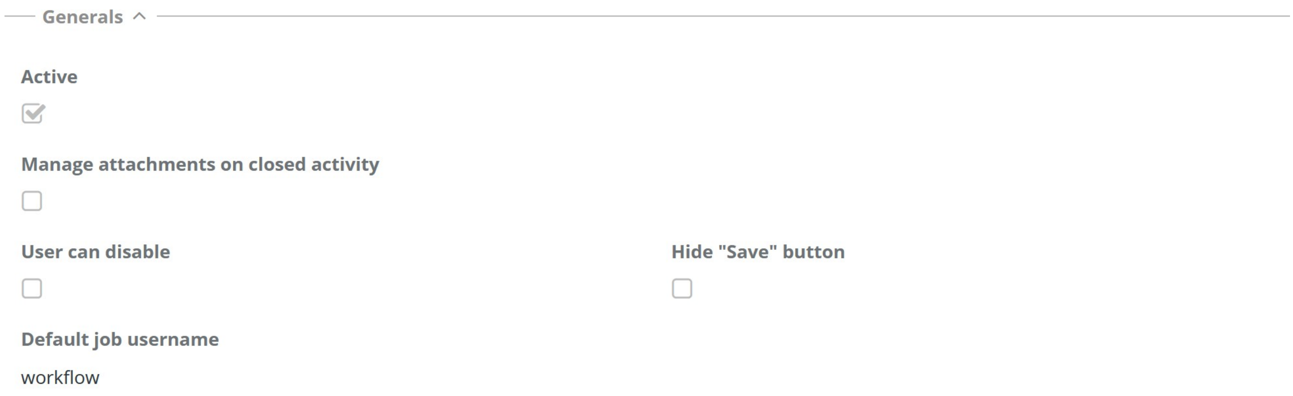
workflow

The required parameters are as follows:

31.4.1. General properties

The following information is required:

- Active: it allows you to activate the CMDBuild workflow engine
- Enable 'Add attachment' when the activities are closed: it allows you to attach documents to closed processes
- Stoppable by user: if enabled, it makes available an additional button that interrupts the process (independently of its status)
- Hide "Save" button: you do not see the button to save the current status of the process before proceeding to the following status
- Default job username: default setting used in the Tasks where not defined. It is related to the username to be used to start the workflow (from which permissions derive, according to the predefined group)



Generals

Active

☑

Manage attachments on closed activity

□

User can disable

□

Hide "Save" button

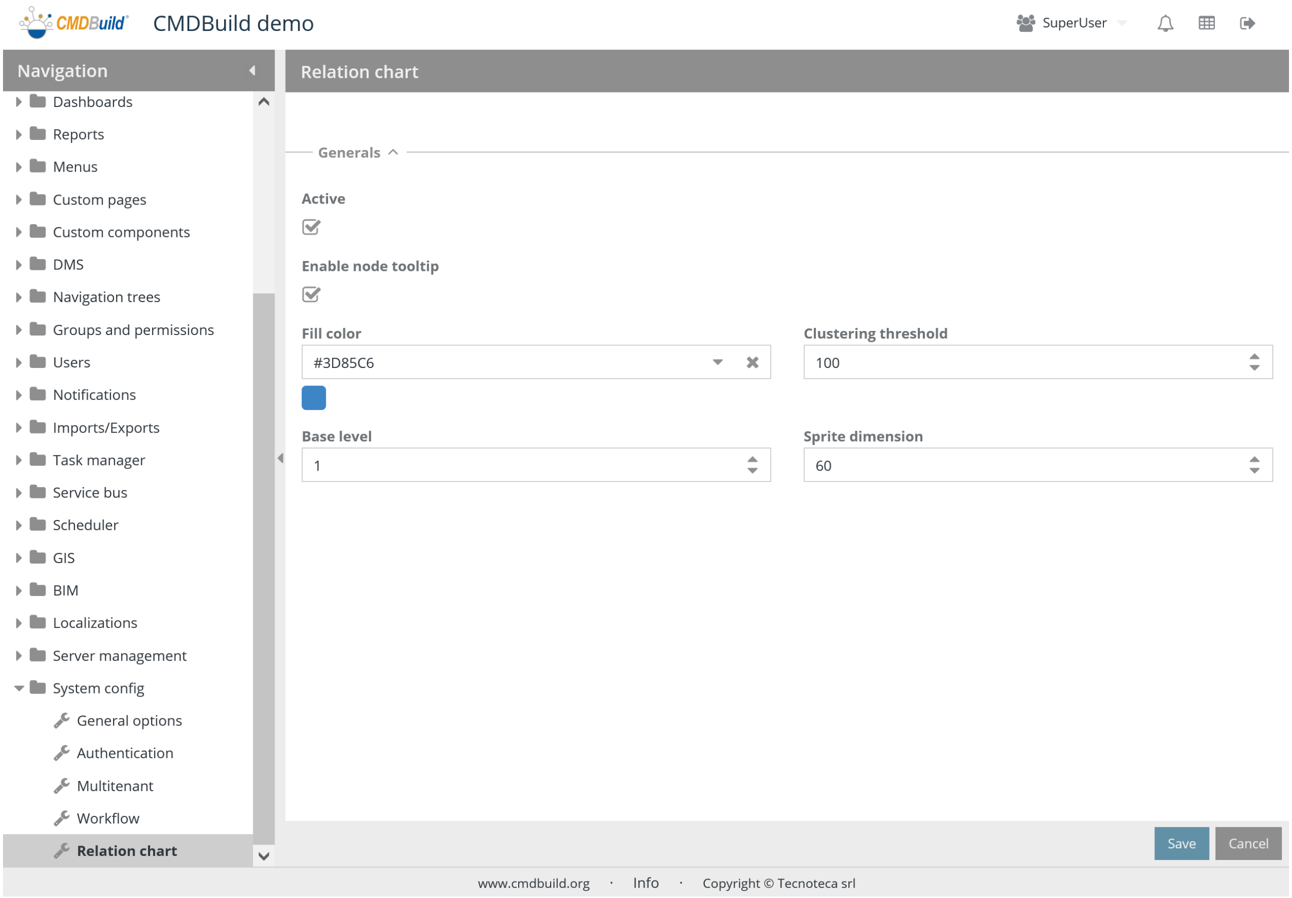
□

Default job username

workflow

31.5. Relation graph

The page contains the relation graph settings.



Relation chart

Generals

Active
☑

Enable node tooltip
☑

Fill color: #3D85C6

Clustering threshold: 100

Base level: 1

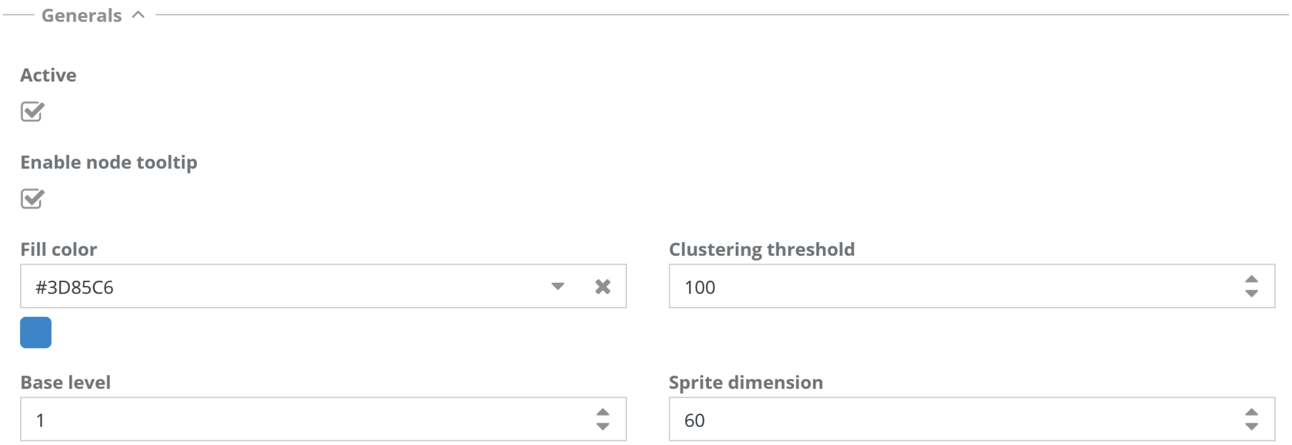
Sprite dimension: 60

The required parameters are as follows:

31.5.1. General properties

The following information is required:

- Active: allows to enable or disable the use of the graph
- Enable tooltip nodes
- Fill-in colour: it allows to set a personalized color for the graph arcs
- "Threshold for clustering nodes" - the maximum number of distinct nodes which can be shown in relation to another node (beside this limit, they are shown as a single different node)
- "Default level" - the default number of levels automatically expanded
- Sprite dimensions: allows to state the dimension of the icons representing different kinds of nodes in the chart



Generals ^

Active

☑

Enable node tooltip

☑

Fill color

#3D85C6

🟦

Clustering threshold

100

Base level

1

Sprite dimension

60

32. Appendix: Glossary

ATTACHMENT

An attachment is a file associated to a card.

In order to manage the attachments, CMDBuild uses in embedded mode any document system which is compatible with the standard protocol CMIS.

The management of the attachments supports the versioning of those files that have been uploaded a few times, with automatic numbering.

See also: Card

WORKFLOW STEP

Activity: workflow step.

An activity can be an interaction with the operator (interactive) or a script that processes operations via API (automatic).

A process instance is a single process that has been activated automatically by the application or manually by an operator.

See also: Process

ATTRIBUTE

The term refers to an attribute of a CMDBuild class (for example in "supplier" class the attributes can be: name, address, phone number, etc.).

CMDBuild allows you to create new attributes (in classes and domains) or edit existing ones.

In the database, every attribute is related to a column in the table which implements the associated class and corresponds, in the Data Management Module, to a data entry field of the specific card for the class management.

See also: Class, Domain, Report, Superclass, Attribute Type

BIM

Method with the aim to support the whole life cycle of a building: from its construction, use and maintenance, to its demolition, if any.

The BIM method (Building Information Modeling) is supported by several IT programs that can interact through an open format for data exchange, called IFC (Industry Foundation Classes).

CMDBuild includes a connector to sync some CI information (technical or maintenance records) and an interactive viewer for the 3D model of the building represented by the IFC file.

See also: CI, GIS

CI

We define CI (Configuration Item) each item that provides IT service to users and has a sufficient detail level for its technical management.

In CMDBuild, the term is applied to a generic context of Asset Management extending the concept usually used in the management of IT infrastructure.

CI examples include: server, workstation, software, plant, electric panel, fire extinguisher, furniture, etc.

See also: Configuration, ITIL

CLASS

A Class is a complex data type having a set of attributes that describe that kind of data.

A Class models an object that has to be managed in the CMDB, such as a company, a building, an asset, a service, etc.

CMDBuild allows the administrator - with the Schema Module - to define new classes or delete / edit existing ones.

A class is represented in the database with a table automatically generated when defining the class and corresponds - in the Data Management Module - to a card for the consultation and update of the cards expected in the model.

See also: Card, Attribute

CMDB

ITIL best practice (Information Technology Infrastructure Library), which has become a "standard de facto" and a non-proprietary system for services management, has introduced the term CMDB referred to the Configuration Item database.

CMDBuild extends the concept of CMDB applying it to a generic Asset Management context.

See also: Database, ITIL

CONFIGURATION

The configuration management process is designed to keep updated and available to other processes the IT items (Configuration ItemsI) information, their relations and their history.

Even if it is known as one of the main processes within the ITIL Best Practice, it is used in CMDBuild for generic contexts of Asset Management.

See also: CI, ITIL

DASHBOARD

In CMDBuild, a dashboard corresponds to a webpage including one or more different graphic representations, in this way you can immediately hold in evidence some key parameters (KPI) related to management aspects of the Asset Management service.

See also: Report

DATABASE

The term refers to a structured collection of information hosted on a server, as well as utility software that handle this information for tasks such as initialization, allocation, optimization, backup, etc.

CMDBuild relies on PostgreSQL, the most powerful, reliable, professional and open source database, and uses its advanced features and object-oriented structure.

The Asset Management database, implemented on the basis of the CMDBuild logics and philosophy, is also indicated as CMDB (Configuration Management Data Base).

DOMAIN

A domain is a relation between two classes.

A domain has a name, two descriptions (direct and inverse), classes codes, cardinality and attributes.

The system administrator, using the Administration Module, is able to define new domains or delete / edit existing ones.

It is possible to define custom attributes for each domain.

See also: Class, Relation

DATA FILTER

A data filter is a restriction of the list of those elements contained in a class, obtained by specifying boolean conditions (equal, not equal, contains, begins with, etc.) on those possible values that can be accepted by every class attribute.

Data filters can be defined and used exceptionally, otherwise they can be stored by the operator and then recalled, or configured by the Administrator and made available by operators.

See also: Class, View

GIS

A GIS is a system able to produce, manage and analyze spatial data by associating geographic elements to one or more alphanumeric descriptions.

GIS functionalities in CMDBuild allow you to create geometric attributes (in addition to standard attributes) that represent, on plans / maps, markers position (assets), polylines (transmission lines) and polygons (floors, rooms, etc.).

See also: BIM

GUI FRAMEWORK

It is a framework provided by CMDBuild to simplify the implementation of custom user interfaces and to grant a simplified access to non-technical users. They can be issued onto any webportals and can be used with CMDBuild through the standard REST webservice.

The CMDBuild GUI Framework is based on javascript JQuery libraries.

See also: Mobile, Webservice

ITIL

"Best practices" system that established a "standard de facto"; it is a non-proprietary system for the management of IT services, following a process-oriented schema (Information Technology Infrastructure Library).

ITIL processes include: Service Support, Change Management and the Configuration Management.

For each process, ITIL handles description, basic components, criteria and tools for quality management, roles and responsibilities of the resources involved, integration points with other processes (to avoid duplications and inefficiencies).

CMDBuild uses some ITIL concepts and applies them to a generic context of Asset Management.

See also: Configuration

LOOKUP

The term "Lookup" refers to a pair of values (Code, Description) set by the administrator in the Administration Module.

These values are used to bind the user's choice (at the form filling time) to one of the preset values (also called multiple choice or picklist).

With the Administration Module it is possible to define new "LookUp" tables according to organization needs.

See also: Attribute type

MOBILE

It is a user interface for mobile tools (smartphones and tablets).

It is implemented as multi-platform app (iOS, Android) and can be used with the CMDB through the REST webservice.

See also: Webservice

PROCESS

The term process (or workflow) refers to a sequence of steps that realize an action.

For each process (type of process) a new process instance will be started when users have to carry out an action on assets according to a procedure implemented as workflow.

A process is activated by starting a new process (filling related form) and ends when the last workflow step is executed.

The workflows managed in CMDBuild are described in the standard markup language XPDL (XML Process Definition Language), ruled by the WFMC (WorkFlow Management Coalition).

See also: Workflow step

RELATION

A relation is a link between two CMDBuild cards or, in other words, an instance of a given domain.

A relation is defined by a pair of unique card identifiers, a domain and attributes (if any).

CMDBuild allows users, through the Management Module, to define new relations among the cards stored in the CMDB.

See also: Class, Domain

REPORT

The term refers to a document (PDF or CSV) containing information extracted from one or more classes and related domains.

Reports can be configured in the Administration Module importing in XML format the description of the layout designed with the visual editor provided by the open source project JasperReports.

CMDBuild users can print reports using the Management Module, which will result as printouts, charts, documents, labels, etc.

See also: Class, Domain, Database

CARDS

The term "card" refers to an element stored in a class (corresponding to the record of a table in the database).

A card is defined by a set of values, i.e. the attributes defined for its class.

CMDBuild users, through the Management Module, are able to store new cards and update / delete existing ones.

Card information is stored in the database and, more exactly, in the table/columns created for that class (Administration Module).

See also: Class, Attribute

SUPERCLASS

A superclass is an abstract class used as template to define attributes shared between subclasses. From the abstract class, or from abstract class hierarchies, you can derive real classes that contain data and include both shared attributes (specified in the superclass) and specific subclass attributes, besides the relations on the superclass domains and on specific domains.

For example, you can define the superclass "Company" with some basic attributes (VAT number, Business name, Phone number, etc.) and the derived subclasses "Customers" and "Suppliers", each one with both generic attributes of the superclass and its own attributes and relations.

See also: Class, Attribute

TENANT

A "tenant", in CMDBuild, is a part of the CMDB reserved to users belonging to a suborganization of the CMDBuild instance (a Group Society, a Seat, a Division, etc.).

Working in "multitenant" mode, every user works only on data of his/her suborganization and, in case, on common data: "tenants".

The list of usable Tenants can be defined from an applicable class of CMDBuild (seats, companies, customers, etc.) or from a database custom function, where you can implement complex visibility rules.

ATTRIBUTE TYPE

Each attribute has a data type that represents attribute information and management.

The type of attribute and its management modes are defined in the Administration Module.

CMDBuild manages the following attribute types: “Boolean”, “Date”, “Decimal”, “Double”, “Inet” (IP address), “Integer”, “LookUp” (lists set in "Settings" / "LookUp"), “Reference” (foreign key), “String”, “Text”, “TimeStamp”.

See also: Attribute

VIEW

A view includes cards defined with logic criteria of filters applied to one or more CMDB classes.

In particular, a view can be defined in CMDBuild by applying a filter to a class (so it will contain a reduced set of the same rows) or specifying an SQL function which extracts attributes from one or more related classes.

The first view type maintains all functionalities available for a class, the second one allows the sole display and search with fast filter.

See also: Class, Filter

WEBSERVICE

A webservice is an interface that describes a collection of methods, available over a network and working using XML messages.

With webservices, an application allows other applications to interact with its methods.

CMDBuild includes a SOAP and a REST webservice, which are provided to external applications to read or write data on CMDB or process operations.

WIDGET

A widget is a component of a GUI that improves user interaction with the application.

CMDBuild uses widgets (presented as "buttons") that can be placed on cards or processes. The buttons open popup windows that allow you to consult or insert data or process other operations.

CMDBuild includes some standards widgets to process the most common operations, but it also supplies the specifications to implement other custom widgets.