Version

2.5

CMDBuild®

» Administrator Manual

November 2017

Author Tecnoteca srl

www.tecnoteca.com

ENG

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

CMDBuild ® is a project of Tecnoteca Srl. Tecnoteca is responsible of software design and development, it's the official maintainer and has registered the CMDBuild logo.

In the project also the Municipality of Udine was involved as the initial customer.

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

Contents

Introduction

CMDBuild is an Open Source web application designed to model and manage assets and services controlled by the ICT Department, therefore it handles the related workflow operations, if necessary according to ITIL best practices.

The management of a Configuration Database (CMDB) means keeping up-to-date, and available to other processes, the database related to the components in use, their relations and their changes over time.

With CMDBuild, the system administrator can build and extend its own CMDB (hence the project name), modeling the CMDB according to the company needs; the administration module 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.

CMDBuild provides complete support for ITIL best practices, which have become a "standard de facto" by now, a non-proprietary system for services management with process-oriented criteria.

Thanks to the integrated workflow engine, you can create new workflow processes with external visual editors, and import / execute them inside the CMDBuild application according to the configured automatisms.

A task manager integrated in the user interface of the Administration Module is also available. It allows to manage different operations (process starts, e-mail receiving and sending, connector executions) and data controls on the CMDB (synchronous and asynchronous events). Based on their findings, it sends notifications, starts workflows and executes scripts.

CMDBuild includes also JasperReports, an open source report engine that allows you to create reports; you can design (with an external editor), import and run custom reports inside CMDBuild.

Then it is possible to define some dashboards made up of charts which immediately show the situation of some indicators in the current system (KPI).

CMDBuild integrates Alfresco, the popular open source document management system. You can attach documents, pictures and other files.

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

The system includes also a SOAP and a REST webservice, to implement interoperability solutions with SOA.

CMDBuild includes two frameworks called Basic Connector and Advanced Connector, which are able - through the SOAP webservice - to sync the information recorded in the CMDB with external data sources, for example through automatic inventory systems (such as the open source OCS Inventory) or through virtualization or monitoring systems.

Through the REST webservice, CMDBuild GUI Framework allows to issue custom webpages on external portals able to interact with the CMDB.

A user interface for mobile tools (smartphones and tablets) is also available. It is implemented as multi-platform app (iOS, Android) and linked to the CMDB through the REST webservice.

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.

Available documentation

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 (“User Manual”)
- installation and system management ("Technical Manual")
- workflow configuration ("Workflow Manual")
- webservice details and configuration (“Webservice Manual”)
- connectors to sync data through external systems (“ConnectorsManual”)

Getting Started

CMDBuild philosophy

A CMDB is a storage and consultation system that handles the information assets of a company.

It is the official central repository and provides a consistent view of IT services.

It is a dynamic system that represents the current situation and knowledge of information technology assets and related entities: hardware (computers, peripherals, networking, telephone equipment), software (basic, environment, application software), documents (projects, contracts, manuals) and other resources, internal and external.

It is a monitoring system for the processes executed, described and managed through the workflow functions.

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 to manage, attributes and relations.

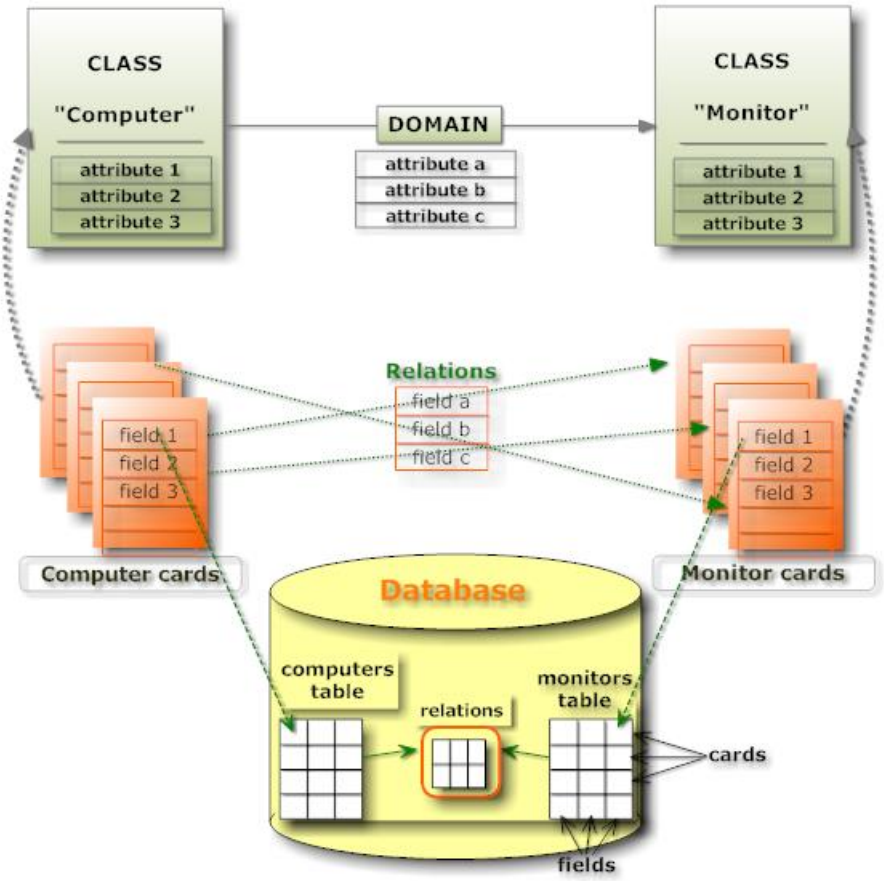
Since there are not two organizations that operate exactly on the same set of objects (assets) and, for each object, on the same information, we decided to set, as the primary CMDBuild feature, the system flexibility, developing features to configure the whole system: data model, processes, reports, external systems connectors, etc.

In particular configuration mechanisms implemented in CMDBuild and managed with the Administration Module allow to:

- add on your own new "classes", i.e. new items or processes typologies (both corresponding to DB charts)
- add / edit the "attributes" of a class, also geometric (DB column)
- define and import custom workflows both for IT process management and data management wizards
- add / edit the widgets to be placed on the card management of a class, through which application functions - useful for that type of object - can be performed
- add "domains", or "relations" between classes (N:M association tables)
- add / edit the "attributes" of a "domain"
- create predefined filters used during the data reference on the various classes
- create "views", both through filters and through SQL queries, which can be used in a way that is similar to the classes
- define navigation tree (subsets of the "domain" graph)
- create tablelists (lookup) to manage attributes with predefined values
- print data model details
- set up dashboards
- define and import custom reports
- configure custom menus for different user groups
- define roles and authorizations for various "classes" categories
- configure the various operations executed by the Task Manager in the background

- define e-mail notification templates
- configure georeferenced GIS and BIM classes
- localization of CMDBuild into other languages
- define options and system parameters

Here is a diagram that explains terms and concepts introduced above in relation to the configuration of the data model.



Design criteria

At first, it's important to:

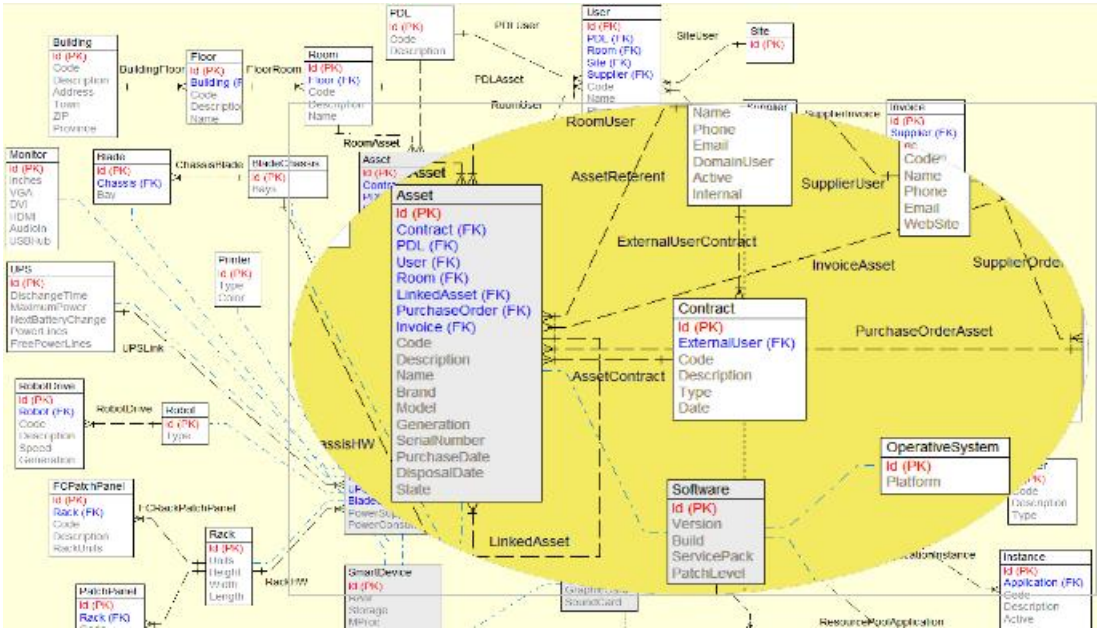
- choose a detail level proportional to the available organization needs, human resources, financial, information and technology
- identify and involve the staff who will set up (Administrator) and update (Operator) information in the system; an outdated system produces costs and no results
- introduce the new application in an organizational system based on procedures, roles and responsibilities that help the IT structure to correctly manage the information
- consider ITIL "best practice" model, a model that has become, in a short time, a "standard de facto" for services management

A successful CMDB project must consider impact and changes introduced by the system and must

acquire the explicit approval of organization's managers.

Working methodology

Enabling 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.



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:

- types of objects ("classes"): hardware (computers, printers, monitors, etc.), software (operating systems, business applications, productivity softwares), services (help desk, licenses, etc.), entities (suppliers, locations, etc.)
- "attributes": useful to define each class (code, description, monitor type, hard disk size, etc.) - each attribute has a specific data type (string, long text, integer, decimal, floating point, date, "lookup" list, reference) and might be a geographical attribute
- relations between classes
- "attributes" are useful to describe each "domain" (dependency between two software applications, number of a switch port connected to a server, relations between a system administrator and servers, etc.) - each attribute has a specific data type (string, long text, integer, decimal, floating point, date, "lookup")
- organizational processes that the system has to manage, described in terms of activities, transition rules and "attributes" involved
- user accounts

Another thing to consider is the hierarchy of classes since, in CMDBuild, it's possible to define generic classes ("superclass", for example "Computer") and then derive subclasses (for example,

"Desktop", "Laptop", "Server"). These subclasses will share some common information (attributes of the superclass) and define some specific attributes that the superclass doesn't have.

It's important to identify a hierarchy that meets the 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 tools
- use the Management Module to insert, update and display cards

Technical Information

For more technical information about the mapping between CMDBuild logical data model and database physical model, see the "Technical Manual".

The same manual also describes how to install the application.

User Interface

The CMDBuild user interface uses Ajax technology.

This solution, part of the new Web 2.0 paradigm, provides a more intuitive application, improves interaction and gets faster responses from the system.

General design criteria

The user interface includes the following main elements:

- left side menu - "accordion" style - to access menus of Management and Administration modules
- data area in the top right corner, containing:
    - add new card button
    - sort options and column list (limited to the Management Module)
    - Page functions, basic and advanced filters, print and export grid data (only for the Management Module)
- data area in the bottom right corner, containing:
    - dedicated TAB to access specific card sections
    - buttons to manage the selected item in the grid
    - complete card - information and labels
    - buttons to confirm or cancel the current operation
    - links to related cards with the ability to insert, modify and delete (logical delete) data
    - buttons on the right side corresponding to the widgets configured for the card

On top of the page, in the header:

- CMDBuild logo (registered trademark Tecnoteca Ltd) on the left
- a clickable central panel with the current user and group; the panel shows the link to the Administration Module (only to system administrators) and Management Module
- button to logout (center panel)
- application name on the right

A footer, containing:

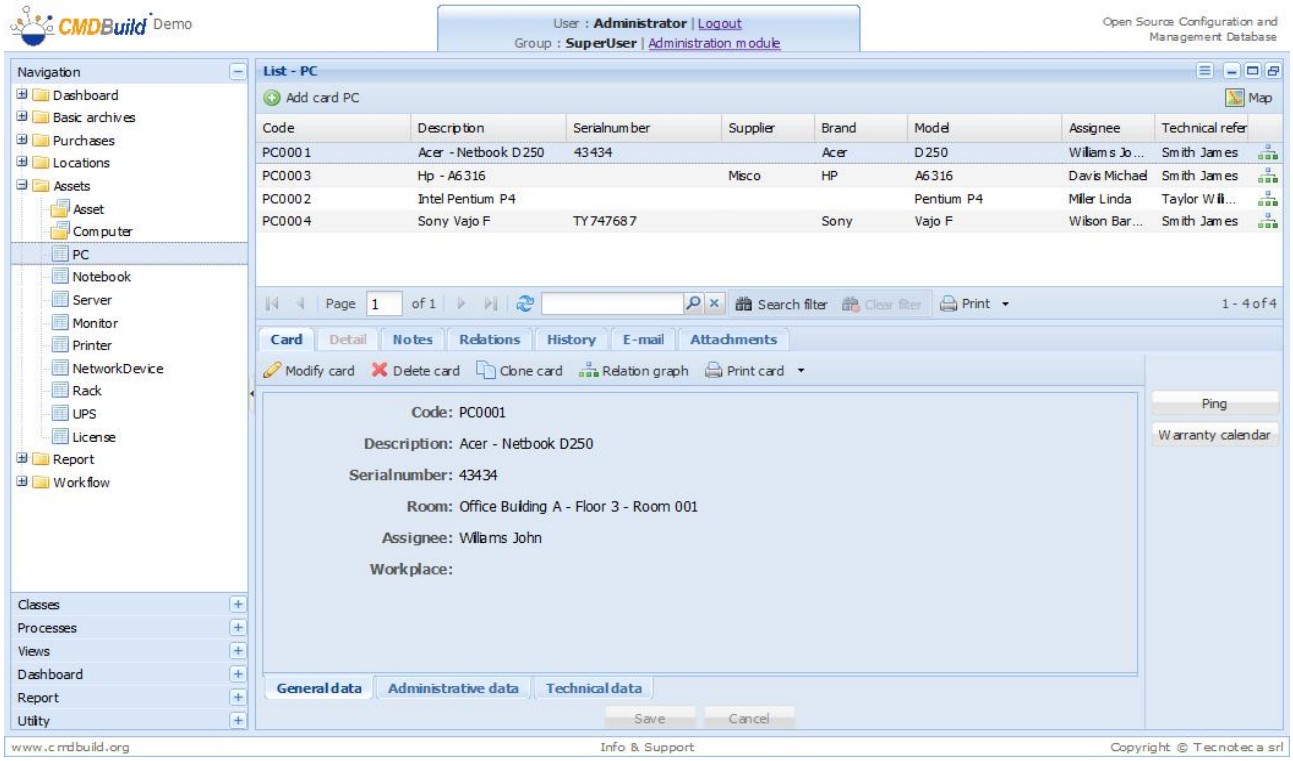
- Official website URL
- application credits
- the copyright notice

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

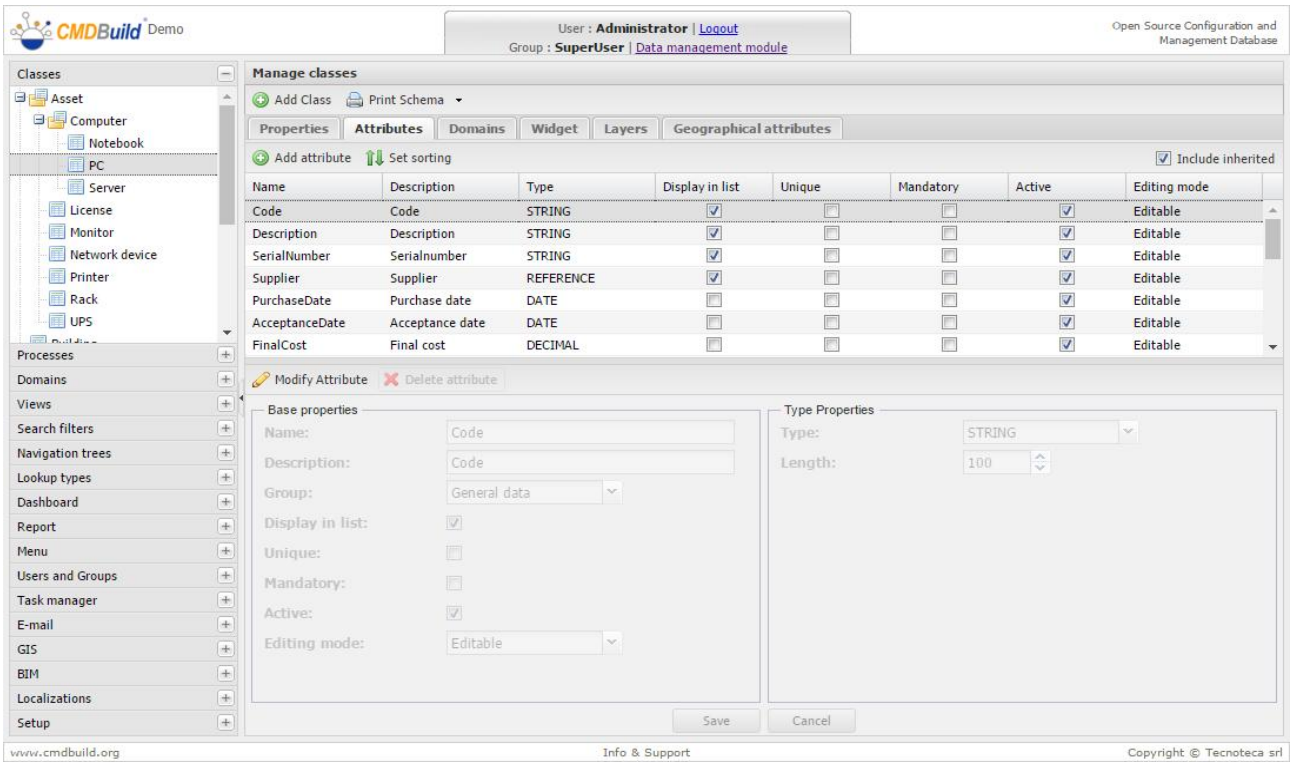
We present now the details of the Management Module since the Administration Module has a dedicated manual.

Here are two screenshots of the Management Module (blue shades) and Administration Module (shades of gray).

Management Module:



Administration Module:

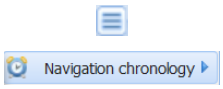


Control elements

Navigation chronology

Top right there is an icon to access the chronology of those actions performed in the system starting from the last login and to enter a card you have already visited:

- navigation chronology



This list shows the sequence of actions and the related type, function and detail. It allows even to return to the same position of the chosen row.

Resizing the main areas

The three main areas described above can be resized by clicking and dragging the layout borders.

On pages with a standard layout (divided into upper and lower area) you can (only in Management Module) act more quickly using buttons located on the top right to:

- expand to full-height card details area
- expand to full-height the data grid
- restore standard layout

“Accordion” menu

The menu on the left side is accordion-style and allows the user to open / close each item of the first level with a simple click.

Opening / closing a menu entry toggles the sub-elements of the selected entry.

Grid management

The standard data grid management (Management Module only) permits to:

- sort on a specific column with a mouse click
- see additional columns in the grid (among available ones)

In some cases you can call special features associated with a row of the grid simply by double clicking on the item (open relations, open document in the attachments tab, etc.).

There is also a button to print (PDF or CSV) data (rows and columns) that's currently displayed in the grid.

Finally, using the "Map" button it is possible to switch between textual and geographical modes.

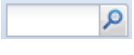
Quick action on grid elements

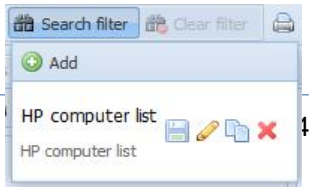
In some cases you can call special features associated with a row of the grid simply by double clicking on the item (open relations, open document in the attachments tab, etc.).

Selection filter

The selection filter, if defined, is available in two modes:

- quick search on all card attributes (not just those shown on the grid)
- advanced search filter, which provides advanced filtering (included





the storing and the filter reuse) and will be described in individual use cases

Interactive help

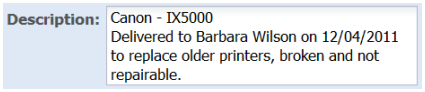
In some cards (in particular in the process management) there is a button in the lower right corner that displays a panel with some compilation tips.

Filling in forms for data editing

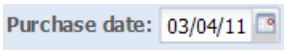
Data cards compilation requires different fields:

- numeric or string type with simple content

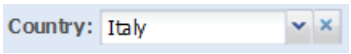
- multiline text



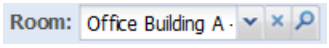
- date type, with interactive calendar



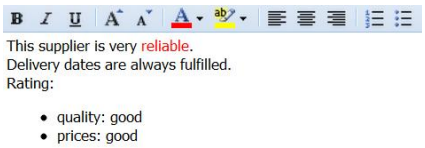
- simple list select



- advanced list select (with filtering options)



- formatted text (with editor)



you can then use the widgets (buttons) configured for the card.

Every insert operation includes confirmation and cancellation buttons

Menu

Both Administration and Management modules work on the same objects, the first to set configuration options (data structures), the second to manage the information stored in these objects (cards).

Both modules have an "Accordion" style menu and include the same entries, however for each entry there are different options in the two modules:

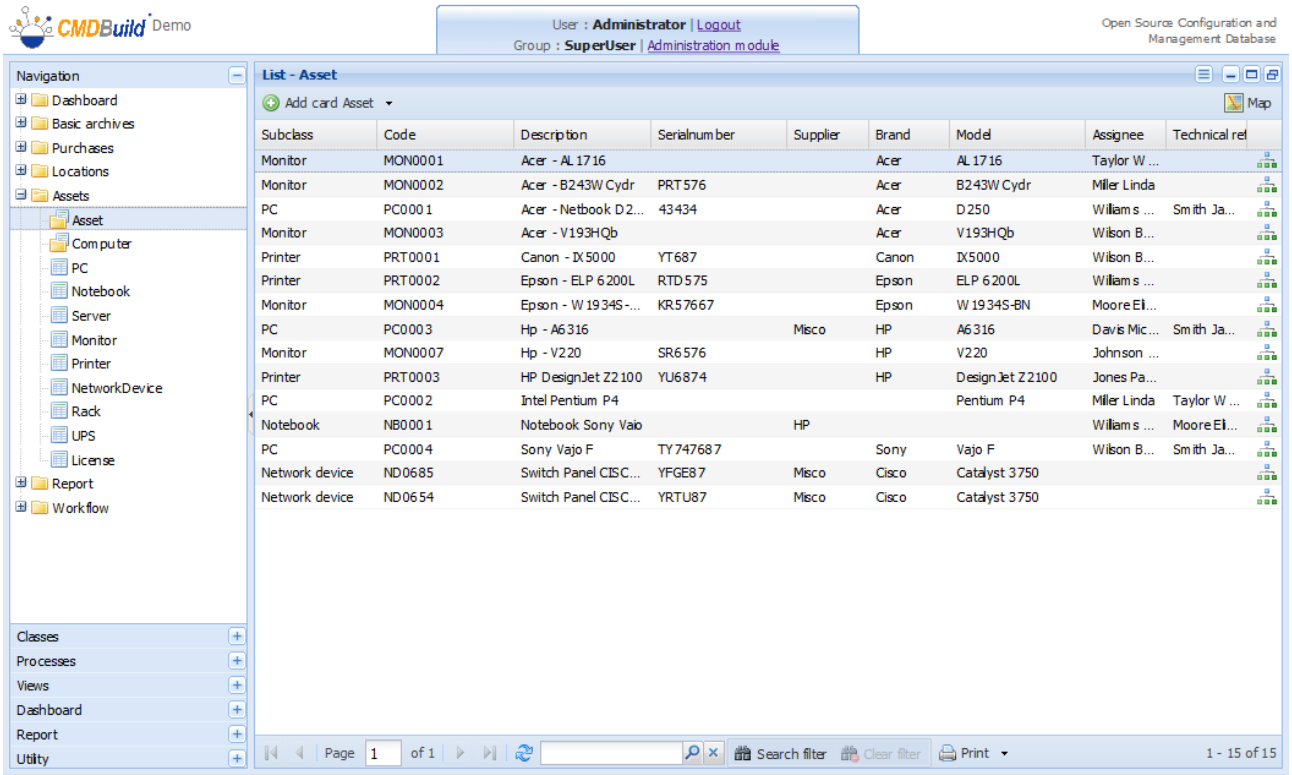
- lookup tables, definition of filters and views, users and groups, definition of menu, GIS configuration, configuration of base parameters in the Administration Module
- navigation menu (list of items available to the current user, ordered in folders in a coherent and organized way) views, dashboards and utilities in the Management Module

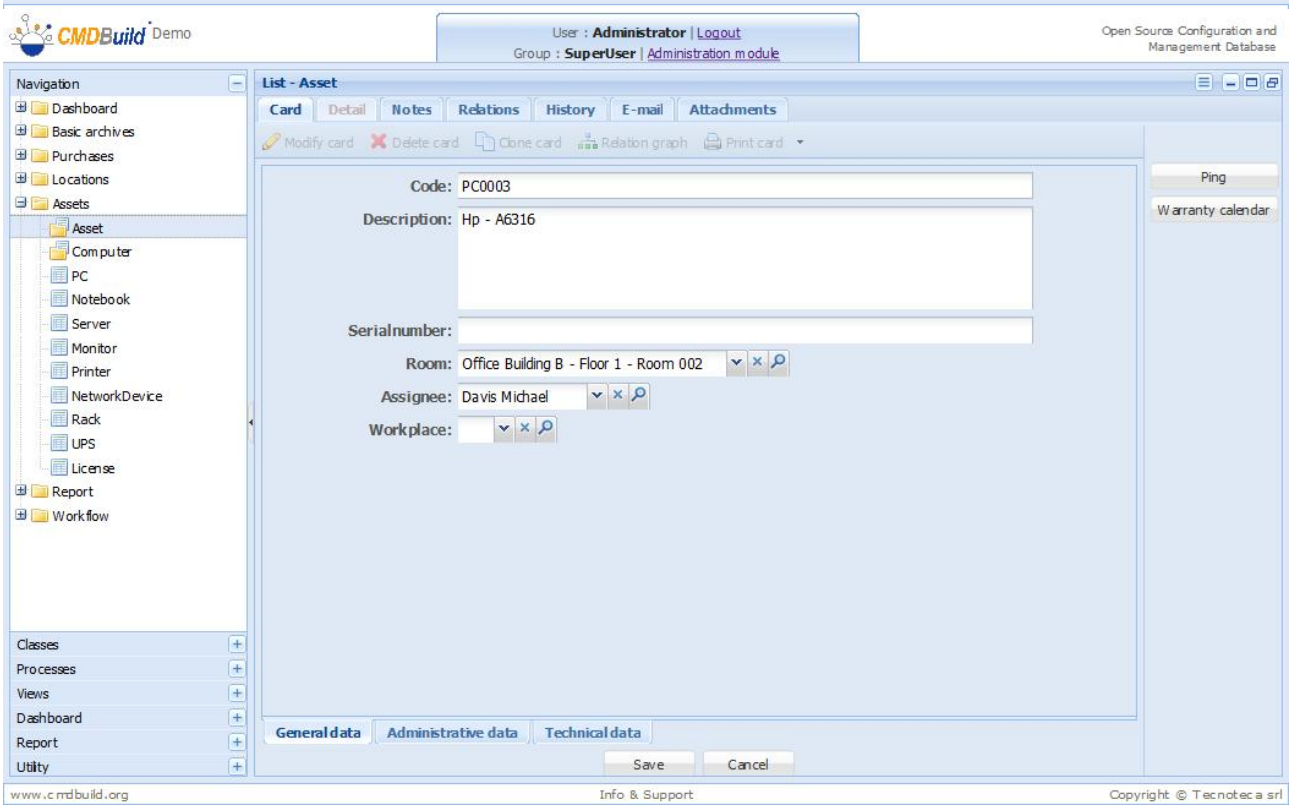
Simplified user interface

Through proper configuration parameters set in the Administration Module (Groups Management – Configuration User Interface), you can define a simplified interface for those users' Groups that demand it.

In particular, you can:

- hide one by one the headings of the accordion Menu placed in the left side of the page
- hide one by one all TABs of the cards and of the processes
- hide the accordion Menu itself when opening the page
- set an alternative display mode of the cards / processes list and of the insertion or update form of a card / process





Accessing the application

The Administration Module is available only to the administrator and can be used to perform operations on CMDBuild model and to make changes and extensions to the existing structure.

The changes made will be immediately available within the Management Module.

PC Requirements

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

The system user has to arrange on his/her processor only an updated web browser (Firefox up to version 43, Chrome up to version 48, Microsoft Explorer 8 or more recent up to version 10).

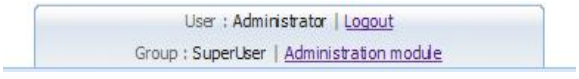
The web architecture ensures complete usability to any IT organization that operates in multiple locations (ie collaborative workflow); any entrusted client can connect and interact with the system using a standard web browser.

Authentication

You must login to use the application.

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

Once the user has logged in, the system starts with the Management Module and only the users that have the required role can switch to Administration Module by clicking on the link on top of the page (visible by clicking on the information panel located in the center)

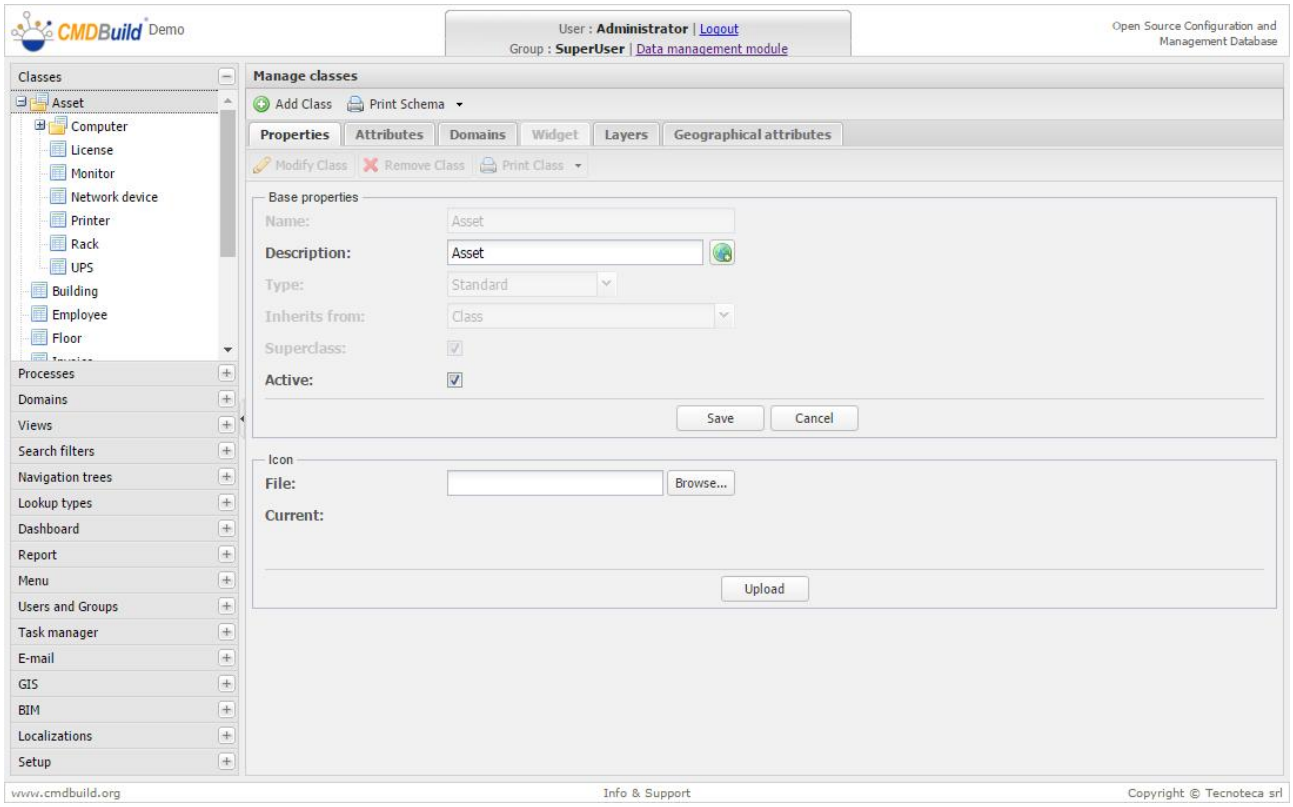


Classes management

Properties tab

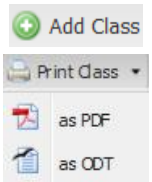
The first tab, "Properties", allows you to create new classes / superclasses and edit some options.

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

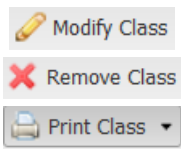


There is the possibility of performing the following operations:

- add a new class
- print data model schema (PDF or ODT)



- edit an existing class by selecting it from the menu
- delete an existing class (logical deletion)
- print current class details



To create a new class you need to specify the following information:

- "Name" - the name of the table in the database (ie "Contract")
- "Description" - the class name used in the application
- "Type" - can be "Standard" (normal CMDBuild class), "Simple" (class without inheritance, history, relations, with "foreign key" attributes but not "reference" attributes - see note below)
- "Inherit from" - name of the superclass (by default all classes inherit from the superclass "Class", anyway you can select a specific superclass, for example "Monitor" is derived from the superclass "Item")
- "Superclass", this flag indicates whether the class will contain data (superclass = no) or will define some common attributes for subclasses (superclass = yes)
- "Active", indicates whether the class is active or has been deleted (logical deletion)

Then, you can associate an icon to the class used for the relations graph.

Note:

The "Simple" class type can be used to store data streams coming from other systems, useful for example to store electricity consumption data, alert notifications on services availability, server logs, etc.

Localization:

In CMDBuild there are two localization typologies:

- first-level localization, related to the only base CMDBuild interface (menu texts and standard buttons, headers, footers, etc.)
- second-level localization, related to all application elements (class names, attributes, domains, lookups, etc.)

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

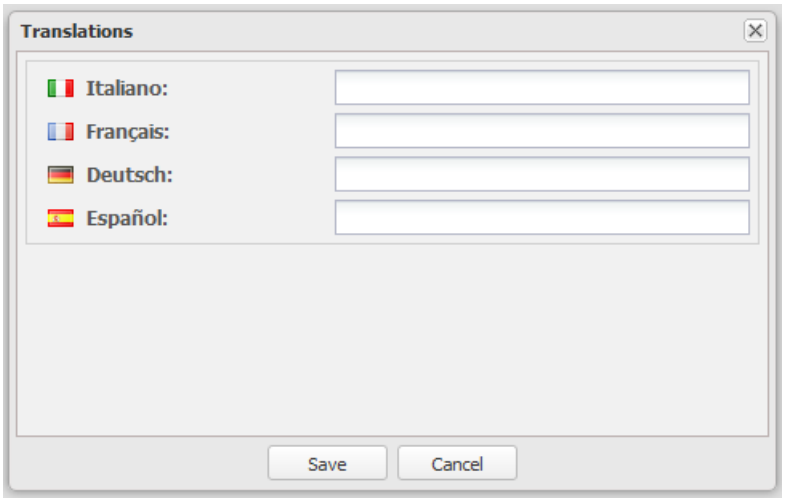
The second-level localization is:

- managed through popup windows through a proper button set in every form requiring it
- filed in a new system table (“_Translation”) in the CMDBuild database

In case of class management, near the "Description" field there is the following icon:

- Translation into other languages 

The icon allows to access the following pop-up window where you can specify the translation of the class name into other languages, which can be selected among those supported by CMDBuild:



By selecting a language the name will be translated into the desired language.

The same mechanism occurs for those elements with the description of a possible translation:

- class description
- attribute description
- description, direct description, inverse domain description
- domain attribute description
- view description
- filter description
- lookup heading description
- dashboard description
- report description
- menu heading description

Otherwise, you can work on all elements, by accessing the menu dedicated to the localization functions (see the specific chapter)

In the manual you can refer to this description wherever you find the translation icon.

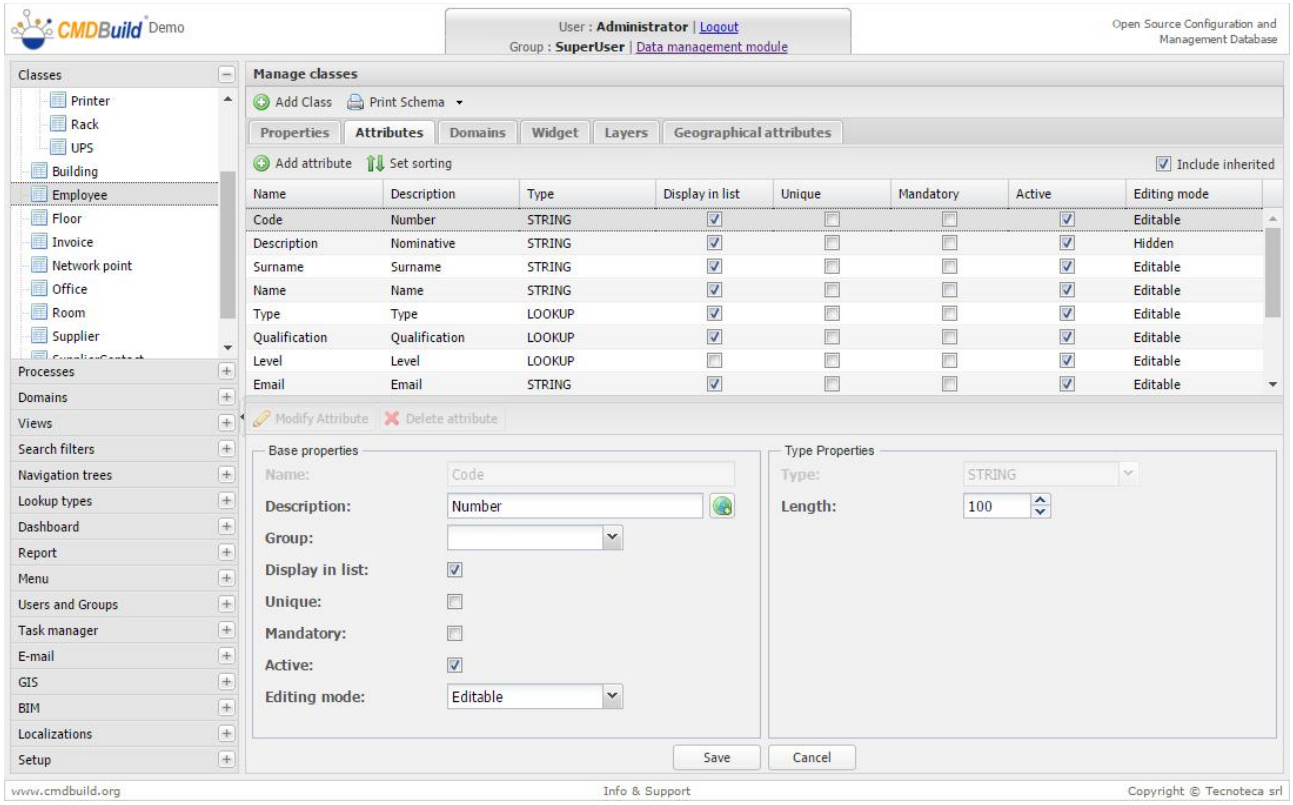
Attributes tab

The second tab, "Attributes", allows you to manage the class attributes.

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

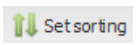
In the example the attributes of the "Monitor" class include:

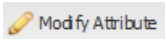
- the attributes inherited from the "Asset" superclass ("Monitor" class inherits from "Asset"); please note that it's not possible to modify inherited attributes
- some other specific attributes, for example, "Type" and "Size"

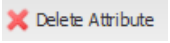


There is the possibility of performing the following operations:

- create a new attribute 

- select the attributes on which you'd like to sort cards 

- edit an existing attribute (edit available fields) 

- delete an existing attribute (if the class contains no data) 

The attributes order can be modified by moving the lines in the grid (drag and drop).

For each attribute you have to specify:

- "Name" - the column name in database
- "Description", - the attribute name in the application
- "Group" - used to group together attributes and organize them in tabs (for example, the

"Monitor" class has several attributes tab: "General", "Administrative data" and "Technical data")

- Data types available
  - “Boolean”,
  - “Char”
  - “Date”,
  - “Decimal”,
  - “Double” (double precision floating point)
  - "Foreign Key" (link to another class, only for "Simple" classes type)
  - “Inet” (IP V4 or IP V6 address, with possibility of specifying the mask value)
  - “Integer”
  - “LookUp” (predefined list of values - could be set in “Settings” / “LookUp”)
  - "Reference" (reference to another class through a "domain", only for "Normal" classes type)
  - “String”
  - “Text”
  - “Time”
  - “TimeStamp”
- “Display in list” - to select if the attribute has to appear in the defaul list of attributes used in the management grid
- "Unique" - to indicate that the field represents a unique key
- "Mandatory", to enable a mandatory field
- "Active", to reactivate attributes logically deleted
- "Editing mode", that can be:
  - Editable: normal editable attribute
  - Read Only: visible but not editable
  - Hidden: attribute not visible

Some attributes have more fields, here are the details.

“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

“String” attributes

For the "string" attributes, a further information is required:

- "Length", indicates the maximum amount of characters allowed

“Text” type attributes

For the "text" attributes a further information is requested:

- "Editor type", might be “Plain text” or “Html”

“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 the "Lookup Types" menu in the Administration Module (see specific chapter).

Hence, when you select a "Lookup" attribute you're asked to select the list of values you want to use.

Note: multilevel lookup lists are also available.

“Reference” attributes

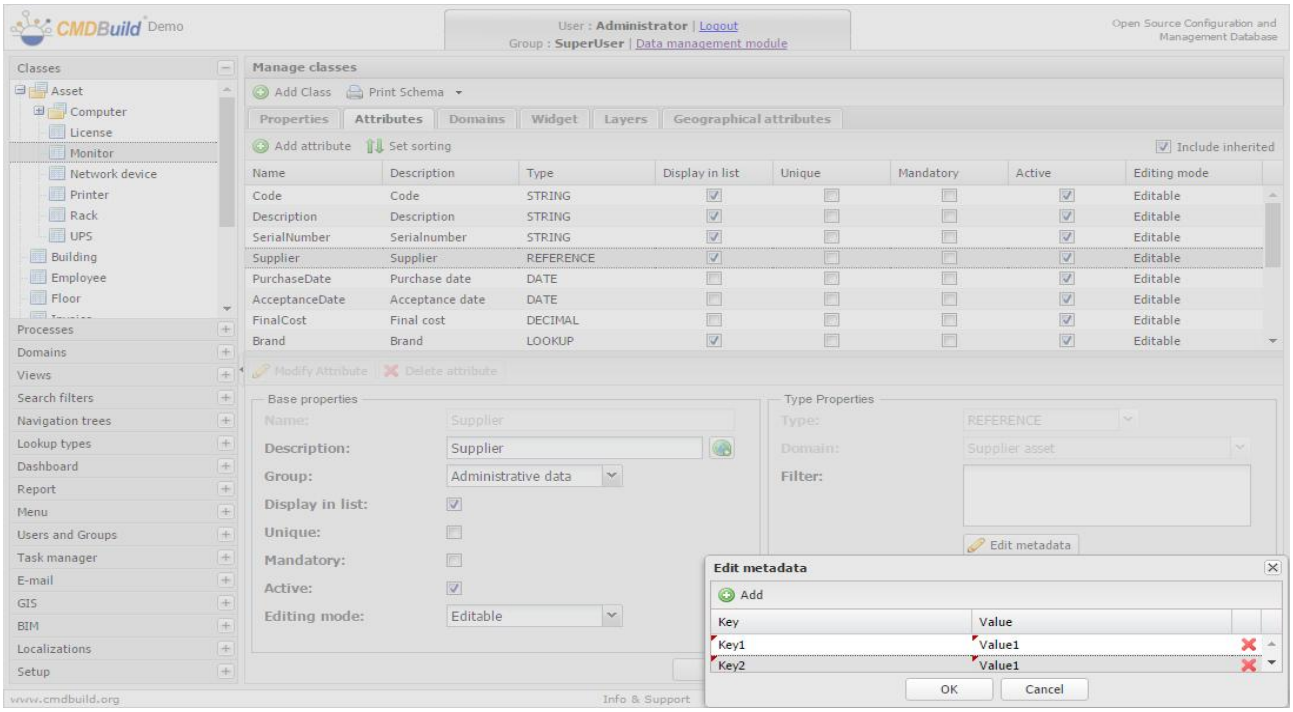
“Reference” attributes allow you to link the attribute to a card of another class via a 1:N domain (where the "N" side is the current class)

In the Management Module it will be possible to select values 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.

Therefore, when defining a "Reference" attribute, you're asked to choose the "domain" you want to apply.

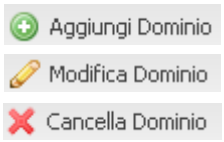
An advanced feature of reference attributes is the CQL (CMDBuild Query Language) filter, that permits to define a filter for the list. The filter could use variables set in the metadata popup window, visible by clicking the "Edit metadata" button (see the appendix).

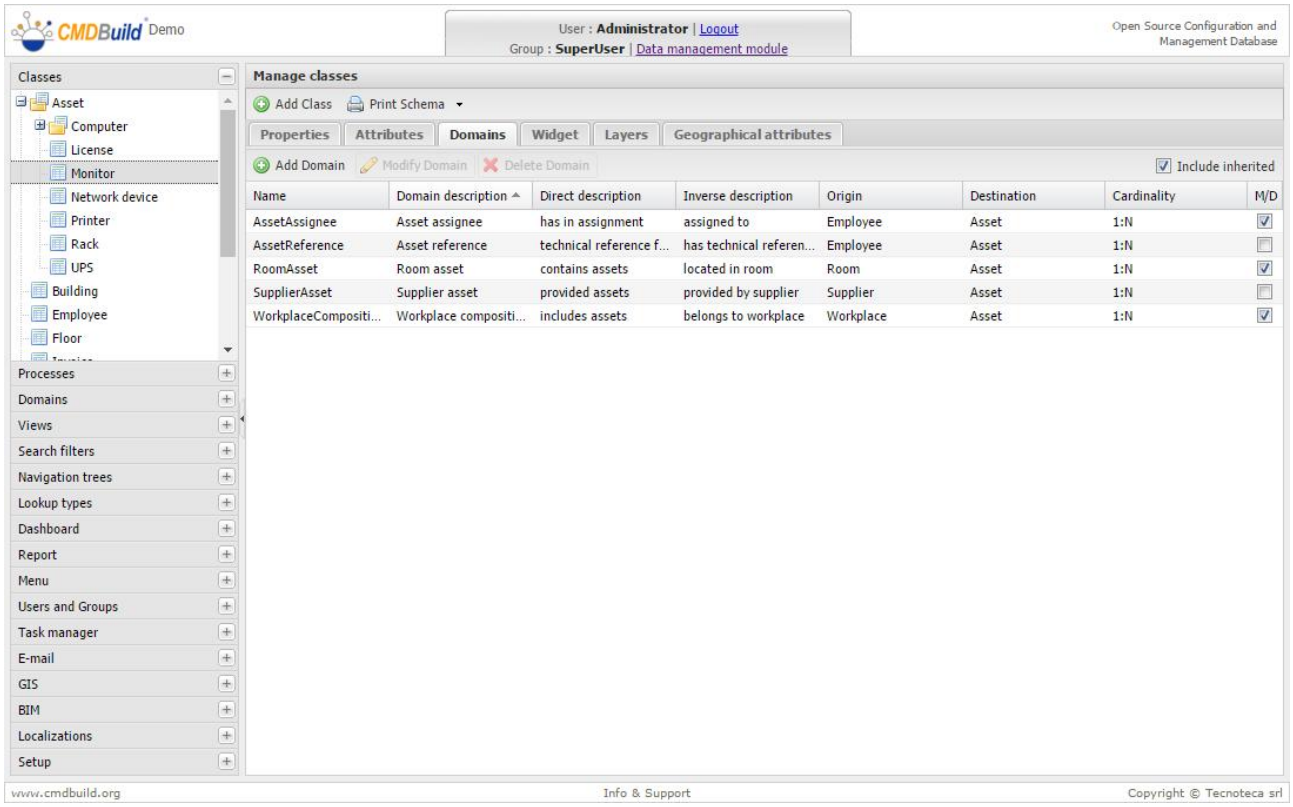


Domains tab

This feature allows you to define relations ("domains") between classes, and involves the following steps:

- overview of domains already configured
- creation of a new domain
- modification of an existing domain
- deletion of an existing domain





Just like for attributes, it's possible to display in the grid:

- only the domains defined on the current class
- the domains defined on the current class plus the domains defined fot the superclass

By selecting the editing functions (insert, modify, delete) the system will change the menu ("accordion" menu on the left) from classes management to domains management, therefore you'll be able to change domain information and domain attributes.

The description of the "Properties" and "Attributes" tabs is available in the Domains chapter.

Widget tab

This feature allows you to configure some default features that will be available as "buttons" in the Management Module.

Currently the following widgets are available:

- Create report: it allows to print a report
- Calendar: it shows the specified deadlines on a calendar
- 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)
- Ping: it performs a ping on the specified host
- Create or modify card: it allows to insert / edit a data card into a class that is different to the current one

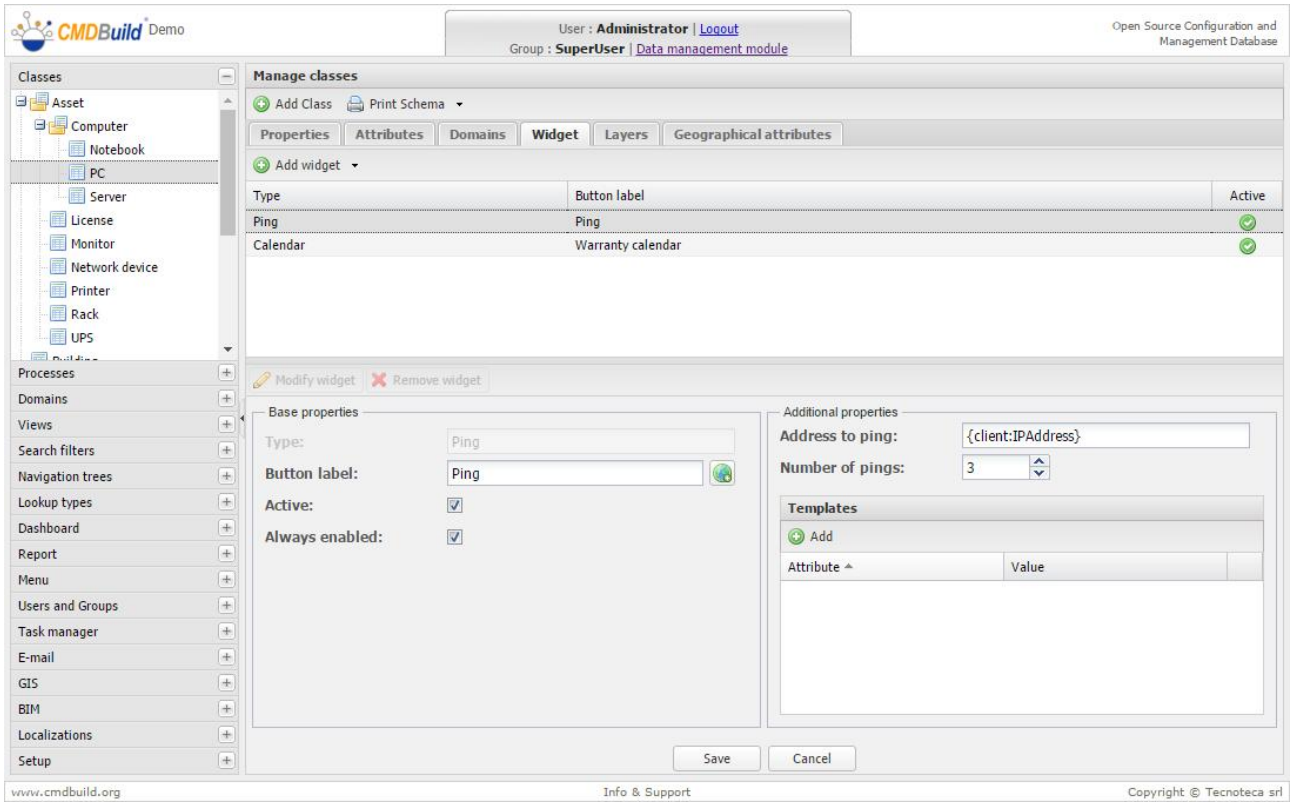
Further widgets can be used just within the workflows (see the Workflow Manual).

There is the possibility of performing the following operations:

- display widgets already configured
- configure a new widget 

- edit an existing widget 

- delete an existing widget 



The following parameters are common to all widgets (others are related to widget type):

- Button text

- Active: when a widget is deleted this i set to false
- Always on: this means that the widget is available also in view mode (no edit mode required)

Print report widget

The specific parameters for this widget are:

- Report: select a report from the list of available reports
- Force format: force output type (PDF or CSV)
- Parameters list: set input values for required report parameters

Calendar widget

The specific parameters for this widget are:

- Reference class: in this class you read the information which will be transfered on the calendar (this is optional if there is the CQL filter)
- Starting date: name of the attribute which contains the starting date of the event
- Ending date: name of the attribute which contains the ending date of the event (this is optional, if the event is omitted it will be considered punctual, i.e. it starts and ends at the same time)
- Default date: default calendar date
- Event's title: name of the attribute which contains the event's title
- CQL filter: selection filter of the charts which you have to treat (this is optional, if specified it has the right over "Reference class")

Workflow launch widget

The specific parameters for this widget are:

- Selection type (by name or through CQL query)
- Process (only in case of “by name”): name of the starting process
- Process attributes (only in case of “by name”): setting of the default value for each parameter which starts the process
- CQL filter (only in case of "through CQL query”): you can define a list of workflows in order to choose dynamically which one you will start, creating a faked application class with the workflow names and defining a selection CQL filter in this field

Widget to perform a “ping” on an IP address

The specific parameters for this widget are:

- IP address: the IP to check

- Number of ping packets
- Custom parameters concatenated to the ping command

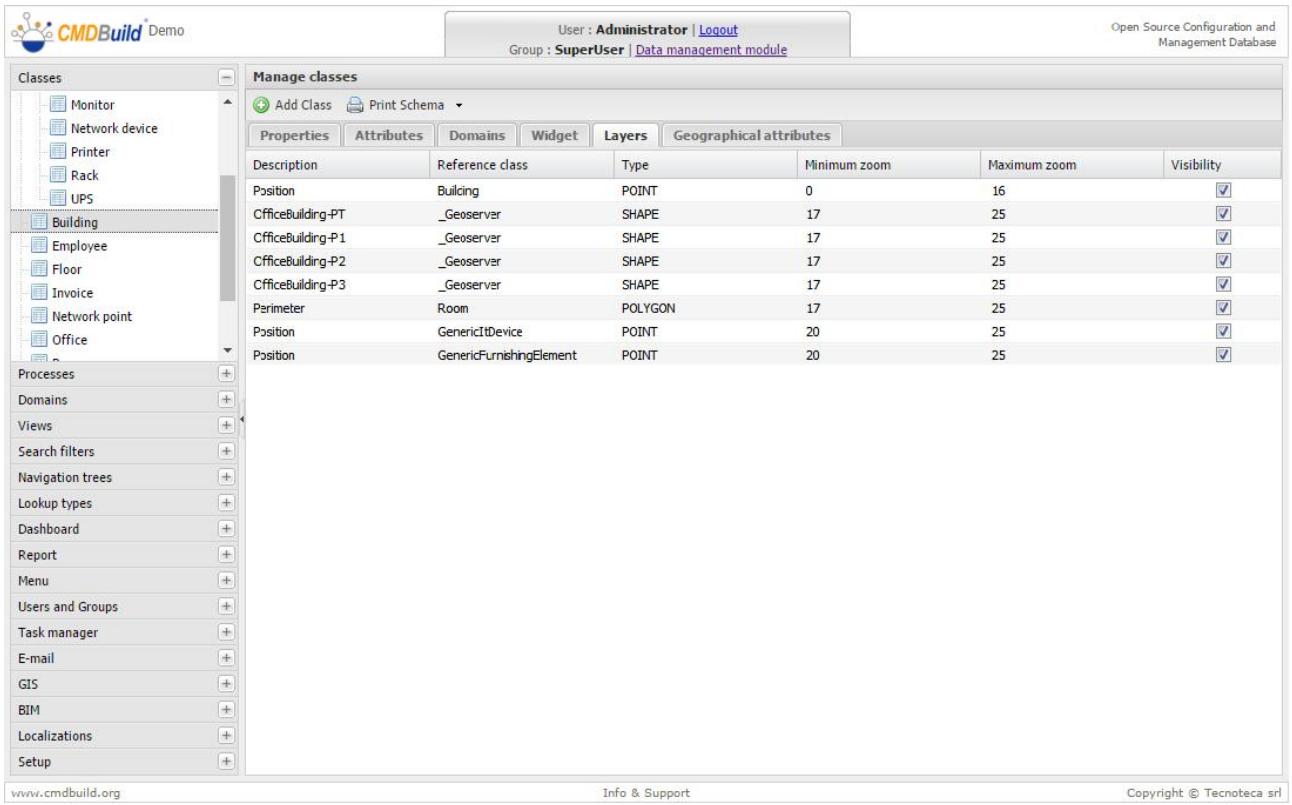
Widget entry / modify card

The specific parameters for this widget are:

- Reference class: this class belongs to the card which you have to work on
- CQL filter: selection filter of the card you have to manage, like the phrase “{client:nome_field_del_form}” (if the filter does not extract any cards, the widget will create a new one)

Layers tab

The "Layers" tab displays the list of geographical attributes available (each one on a separate layer) and allows you to select the ones to show as background for the current class.

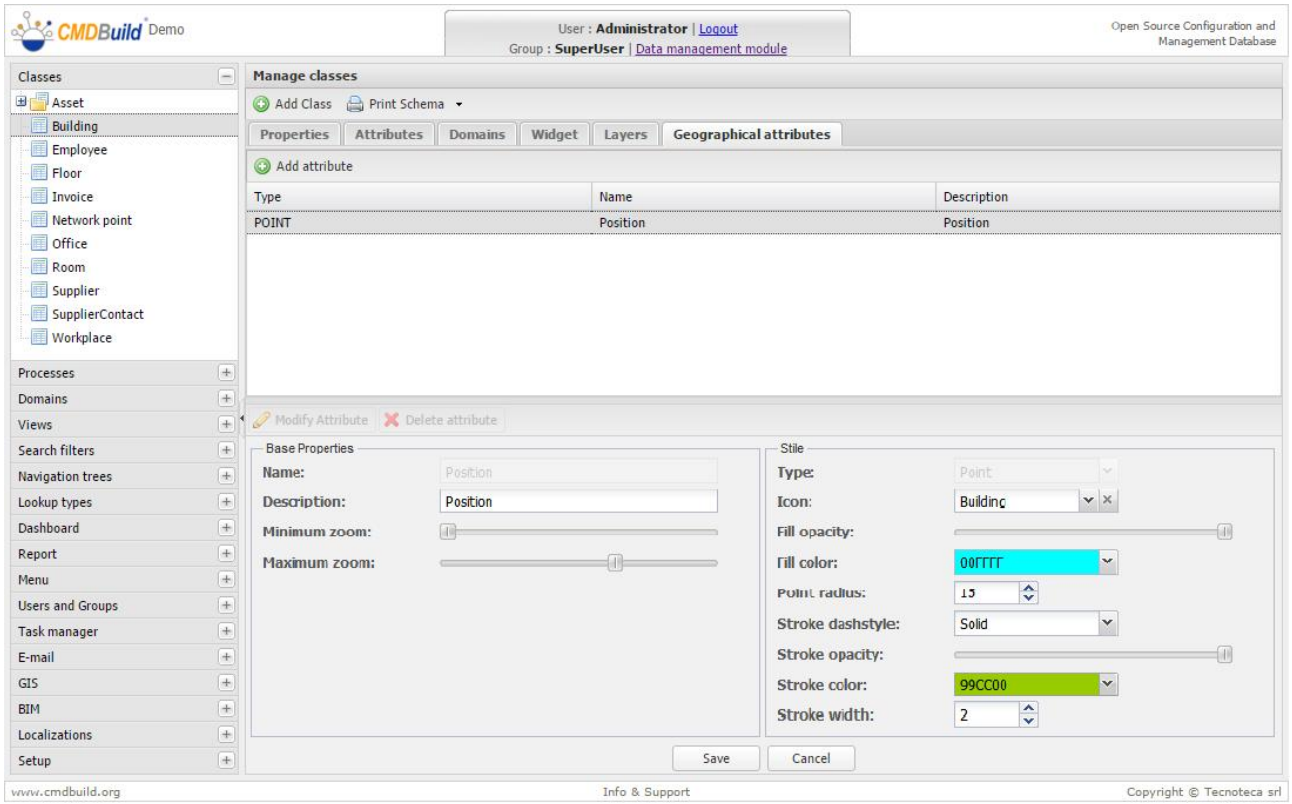


Geographic attributes tab

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

There is the possibility of performing the following operations:

- create a new geographic attribute
- edit an existing attribute (edit available fields)
- delete an existing geographic attribute (if the class contains no data)



The attributes order can be modified by moving the lines in the grid (drag and drop).

For each attribute you have to specify:

- "Name" - the column name in database
- "Description", - the attribute name in the application
- "Minimum zoom", the minimum zoom level required to display the attribute (for example, if we have the whole Italy in the viewport it makes no sense to show cities buildings since they will appear overlapped)
- "Maximum zoom", the maximum zoom level required to display the attribute

- Data types available
  - "Point",
  - "Line"
  - "Polygon"

Some geographical attributes have more fields, here are the details.

"Point" attributes

For "Point" attributes this additional information is required:

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

"Line" attributes

For "Line" attributes this additional information is required:

- "Stroke color"
- “Stroke opacity”
- "Stroke width", from 1 to 10
- "Stroke dashstyle" (dot, dash, dashdot, longdash, longdashdot, solid)

"Polygon" attributes

For "Polygon" attributes this additional information is required:

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

Process management

General Information

CMDBuild is a management system for the IT department and supports ITIL "best practices"; CMDBuild models and manages the configuration management database and supports related processes.

Given the amount of processes options, the organizational procedures and the flexibility pursued by the CMDBuild project, we chose not to implement a series of rigid and predefined processes, but a generic workflow engine to model processes case-by-case.

The workflow management system is an important feature of CMDBuild and provides:

- a standard interface for users
- a secure update of the CMDB
- a tool to monitor provided services
- a repository for activities data, useful to check SLA

Representation modes

In CMDBuild each workflow contains information about:

- activities sequence, with conditional flows
- information to be displayed or to be filled by user on each step
- tasks to be performed (start process, update DB, send mail etc.)
- roles authorized to perform each step of the workflow

Workflows are designed using the JPEd visual editor, an external open source tool that uses XPDL standard, and then imported into CMDBuild and executed with the Enhydra Shark engine.

The "glossary" used in CMDBuild includes the following terms:

- process (or workflow): sequence of steps ("activities") to perform a specific action in compliance of specific rules, using a wizard
- activity: a workflow step
- process instance: active process created executing the first step
- activity instance: creation of an activity, accomplished automatically or by an operator

The above terms are arranged into CMDBuild as follows:

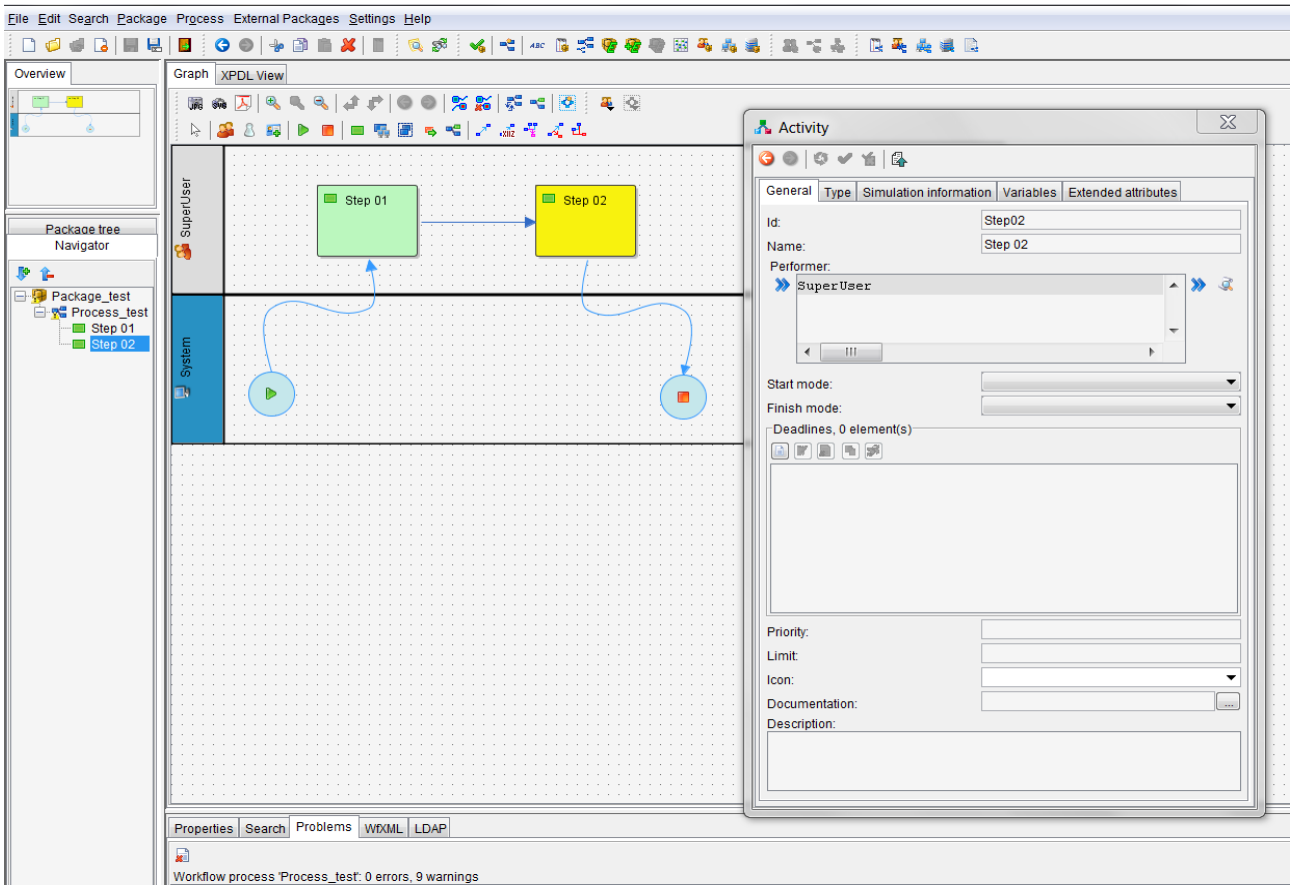
- each process is represented by a special class configured using the "Processes" menu in the Administration Module; the class includes all the attributes of scheduled activities
- each process instance corresponds to the current card, plus the list of its versions (ended activities)
- each activity instance corresponds to a card (current activity) or to a historicized version (ended activity)

Modeling a new process in CMDBuild requires the configuration of a new "process" class that includes all the attributes of scheduled activities.

For all the details about workflow configuration, check out the "Workflow Tutorial" manual.

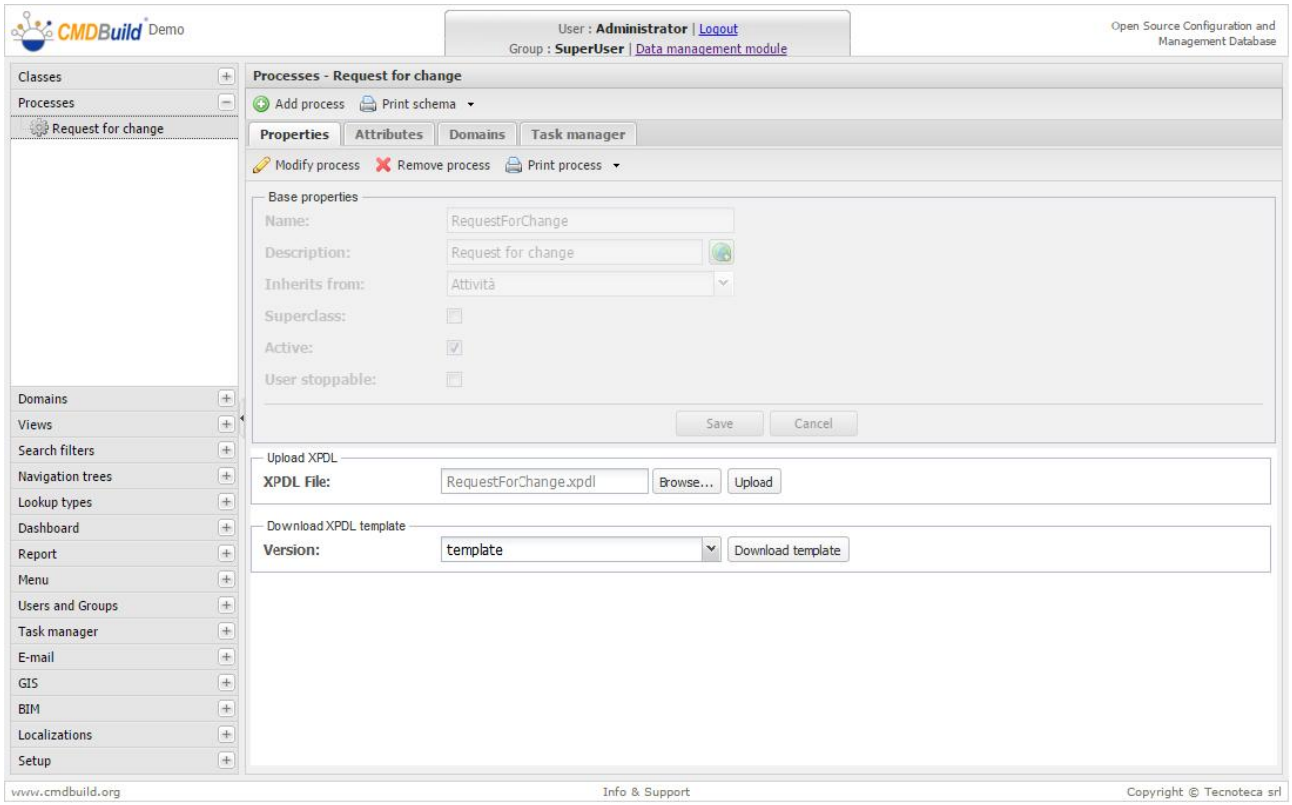
This manual only describes how to configure a "process" class, describing in particular how to use the "XPDL" and "Planning" tabs; these tabs represent the main difference between a standard class

and a process class.



Properties tab

The TAB Process Properties differs from the Class Properties one because it provides another data section that allows the upload/download of the XPDL file; you can use this tab to download the initial XPDL definition and, once designed the process, upload your file.



Upload XPDL

This function allows you to import the XPDL file that you've designed and configured using the external editor TWE (or JPEd).

The information managed by CMDBuild is:

- user stoppable process (enabled / disabled)
- XPDL file
- process sketch (image)

Once concluded the import operations described above, the new process is included in the Management Module, thus the process can be executed using the workflow engine "Enhydra Shark".

You can edit a process you've already

Download XPDL template

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

The exported file includes:

- process name
- the list of process attributes (corresponding to the list of attributes defined in the class)
- the list of methods available (for the CMDBuild application)
- the list of "actors" (users) that interact with the process (the "System" role is automatically created to identify system activities); the list corresponds to the user groups defined in CMDBuild
- data type description for custom Lookup and Reference attributes

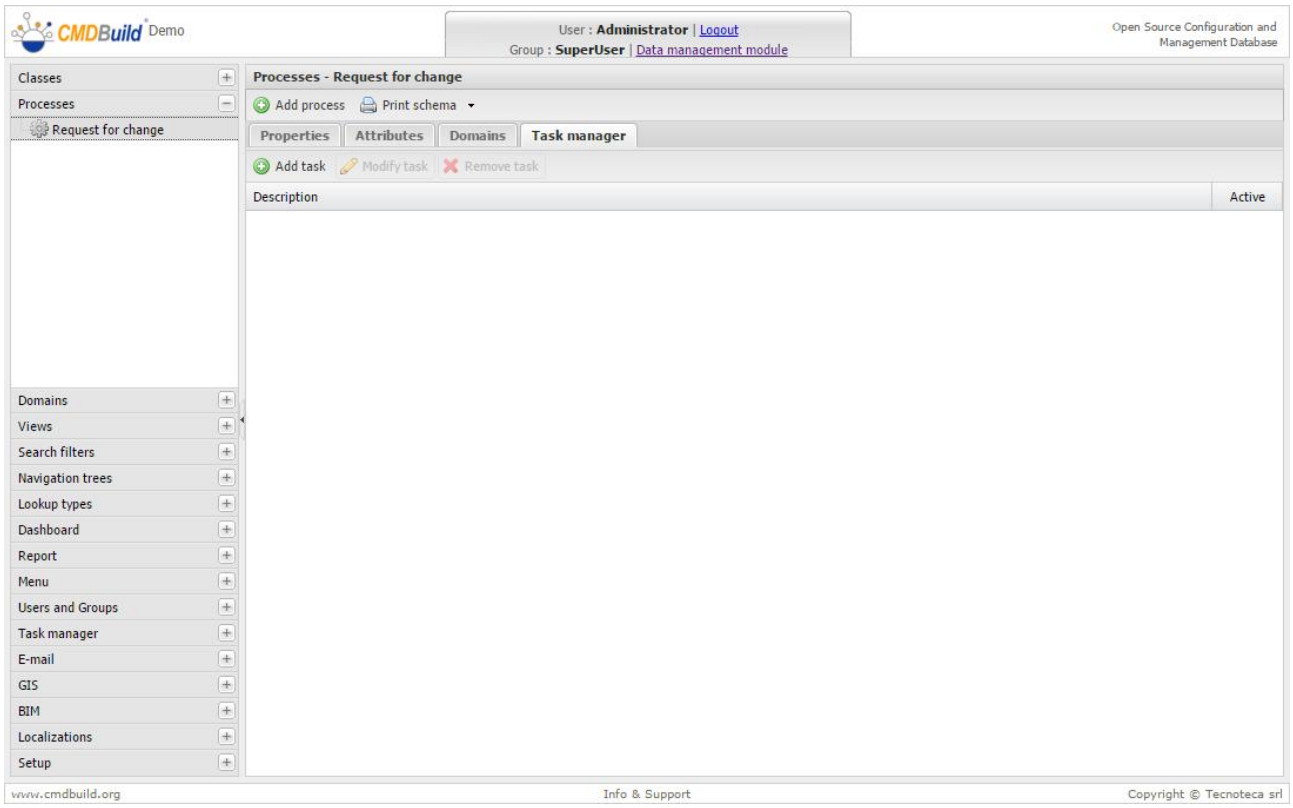
The information managed for this function are:

- process class name (read only mode)
- user stoppable process (enabled / disabled)

The same function can be used to download the current process and edit workflow steps using an external editor.

Task Manager tab

The “Task Manager” TAB allows you to access the functions of the Menu heading in order to schedule the execution of the current process.



By the order "Add process", you will shift to the Task Manager function with the wizard for the scheduling of the current started process.

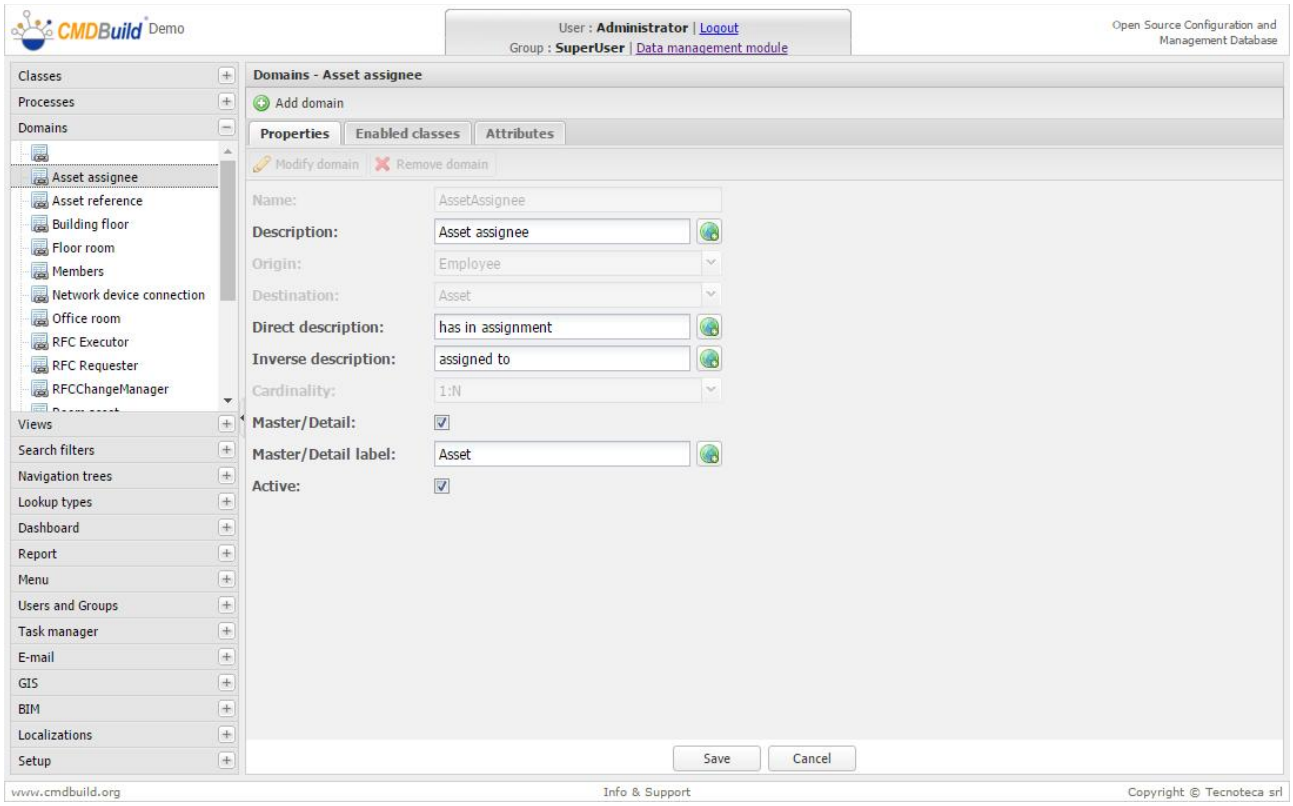
Please see the chapter about Task Manager for more details about it.

Manage Domains

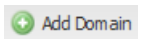
Properties tab

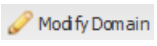
The first tab, "Properties", allows you to create new domains and edit domains' details.

A domain represents relations to be stored in the CMDB (supplier provides asset, user uses computer, computer has software, etc.). The domain is represented by a list of relations and related attributes using a table created CMDBuild automatically in the database to store the reports added.



This feature allows you to define relations ("domains") between classes, and involves the following steps:

- visualization of configured domains
- creation of a new domain 

- modification of an existing domain 

- deletion of an existing domain 

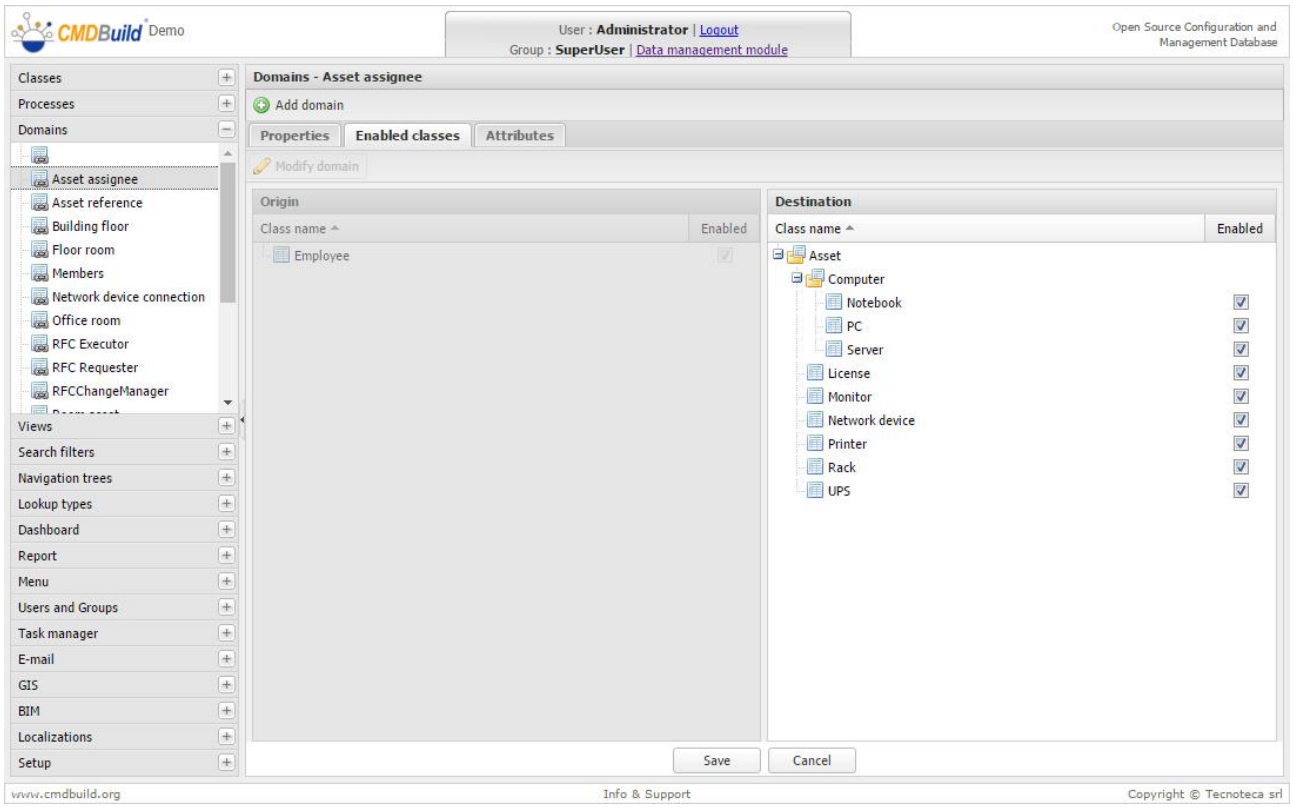
For each domain you must define:

- "Name" - the name of the table in the database
- “Description”- the mater-detail button (if applicable) in the Management Module

- "Origin" - such as "supplier"
- "Destination" - such as "assets"
- “Direct description”, for example, the supplier "has provided" the asset
- “Inverse description”, for example, the asset "has been provided by" a supplier
- "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 validate new relations
- "Master Detail" - activate master-detail feature, ie elements of the detail class ("N" side) presented as rows of the main class ("1" side)
- "Active", indicates whether the domain is active or has been deleted (logical deletion)

TAB Enabled classes

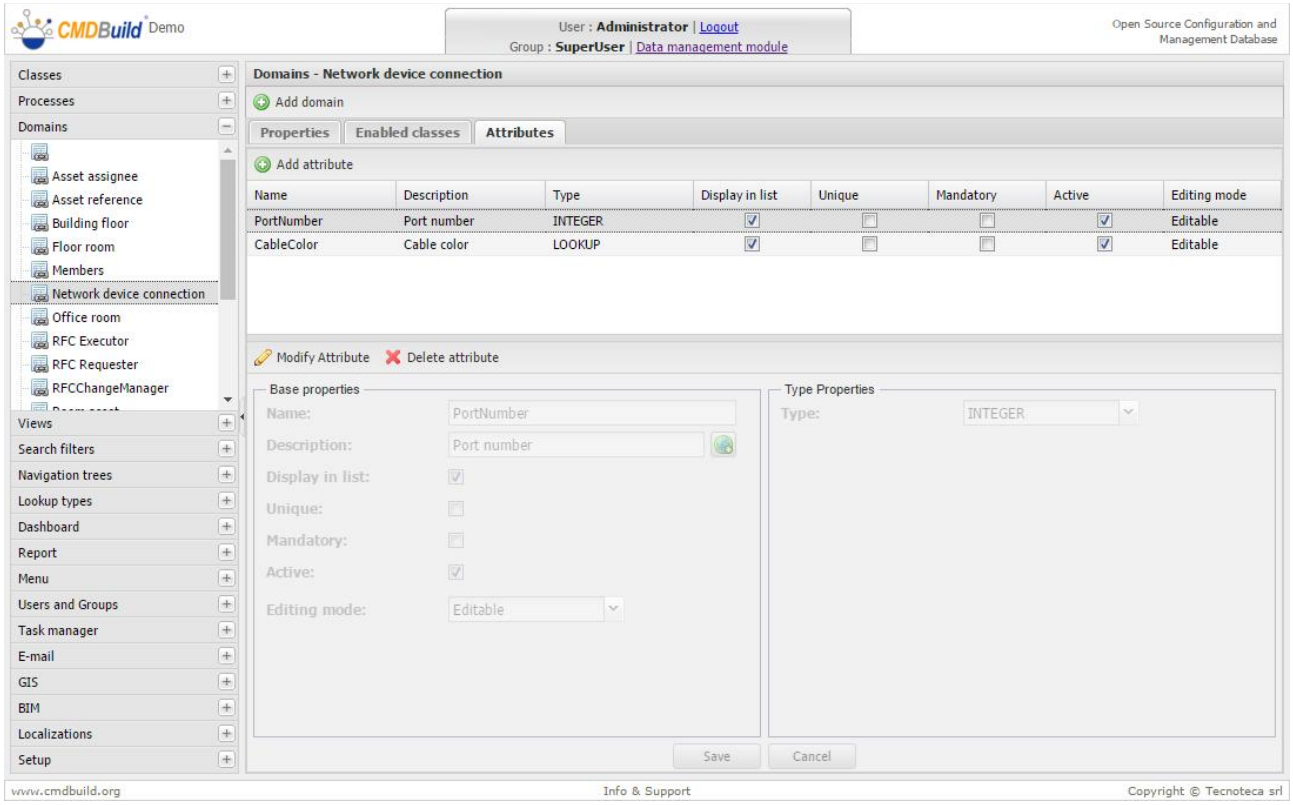
The second TAB “Enabled classes” allows you to disable a domain defined on a superclass on one or more subclasses.



Attributes tab

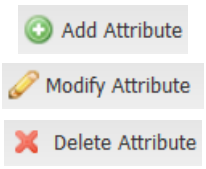
The third TAB “Attributes” allows you to manage the domain attributes.

This tab is used to add information to a new domain, or update an existing domain.



There is the possibility of performing the following operations:

- create a new attribute
- edit an existing attribute (edit available fields)
- delete an existing attribute (if the class contains no data)



The attributes order can be modified by moving the lines in the grid (drag and drop).

For each attribute you have to specify:

- "Name" - the column name in database
- "Description", - the attribute name in the application
- Data types available
  - “Boolean”,
  - “Char”
  - “Date”,
  - “Decimal”,

    - “Double” (double precision floating point)
    - "Inet" (IP address)
    - “Integer”
    - “LookUp” (predefined list of values - could be set in “Settings” / “LookUp”)
    - “String”
    - “Text”
    - “Time”
    - “TimeStamp”
- "Display in list", to show the attribute in reference's details, in the Management Module
- "Unique" - to indicate that the field represents a unique key
- "Mandatory", to enable a mandatory field
- "Active", to reactivate attributes logically deleted
- "Editing mode", that can be:
    - Editable: normal editable attribute
    - Read Only: visible but not editable
    - Hidden: attribute not visible

Each attribute might have more fields, according to the same rules described for class attributes.

Views

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

Views can be defined as subsets of cards of the same class (based on filters) or as groups of data cards made up by information filed in different classes (based on SQL queries).

In the first case they keep their common functionalities of the class management, in the second one they are Read Only.

This views mechanism recalls the restriction functionality of access permissions to a class keeping out some rows or columns.

While the restrictions in the access permissions keep out definitively the consultation of certain information, for a single user group there is the possibility to define a view that shows a certain data group and another view that show other data (additional or separated).

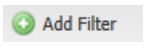
For instance, when using a "State" attribute or an "Assignee" one, you can create for the same group of operators an "Asset in use" view and an "Available assets" view from the "Assets" class.

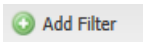
Views based on filters

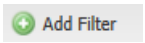
The system administrator can create a view based on filters:

- by reusing and cloning a filter defined by a user
- by defining a new search filter, i.e. by setting search criteria on attributes of the specified source class

It's possible to perform the following operations:

- initiating the creation of a new view 

- editing a pre-existing view 

- deleting a pre-existing view 

- opening a popup window where you can choose a user filter or create a new filter 
- deleting the current filter criteria 

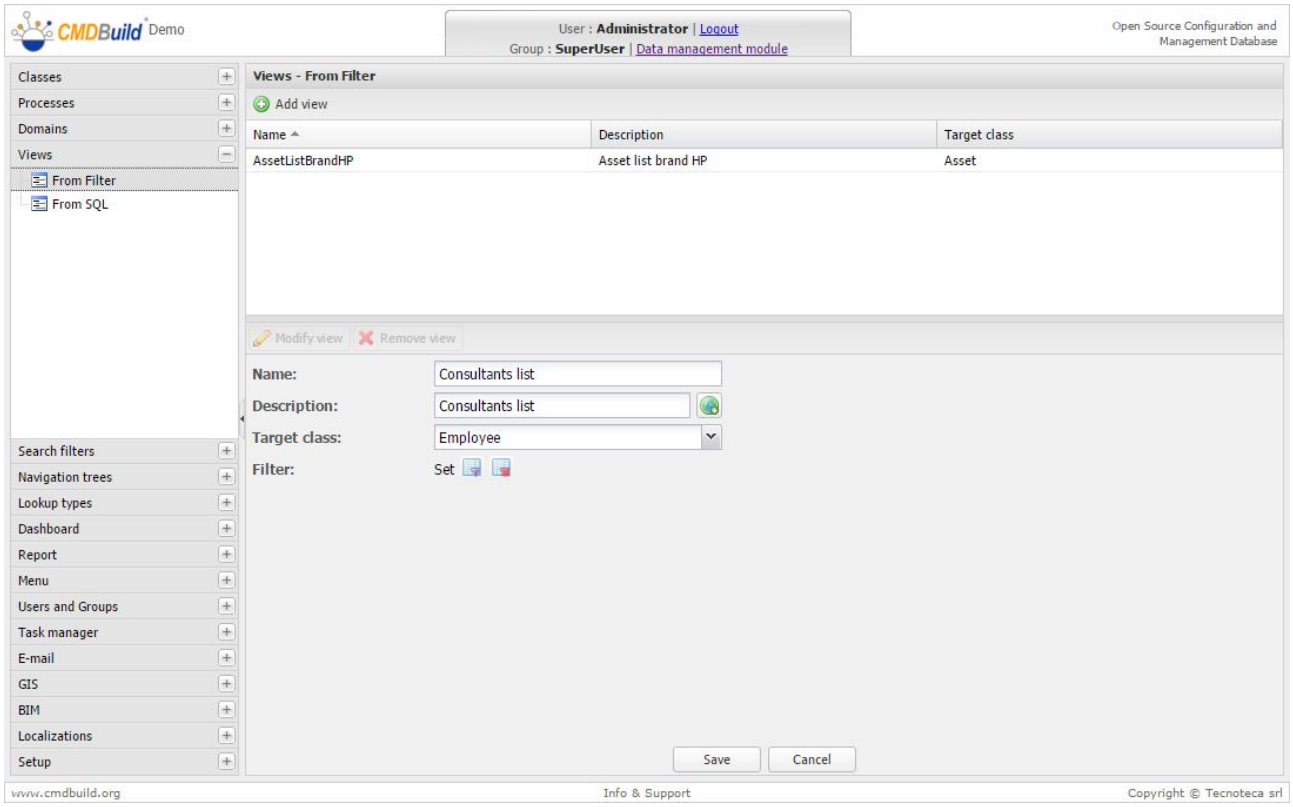
To create or modify a search filter, the following information is required:

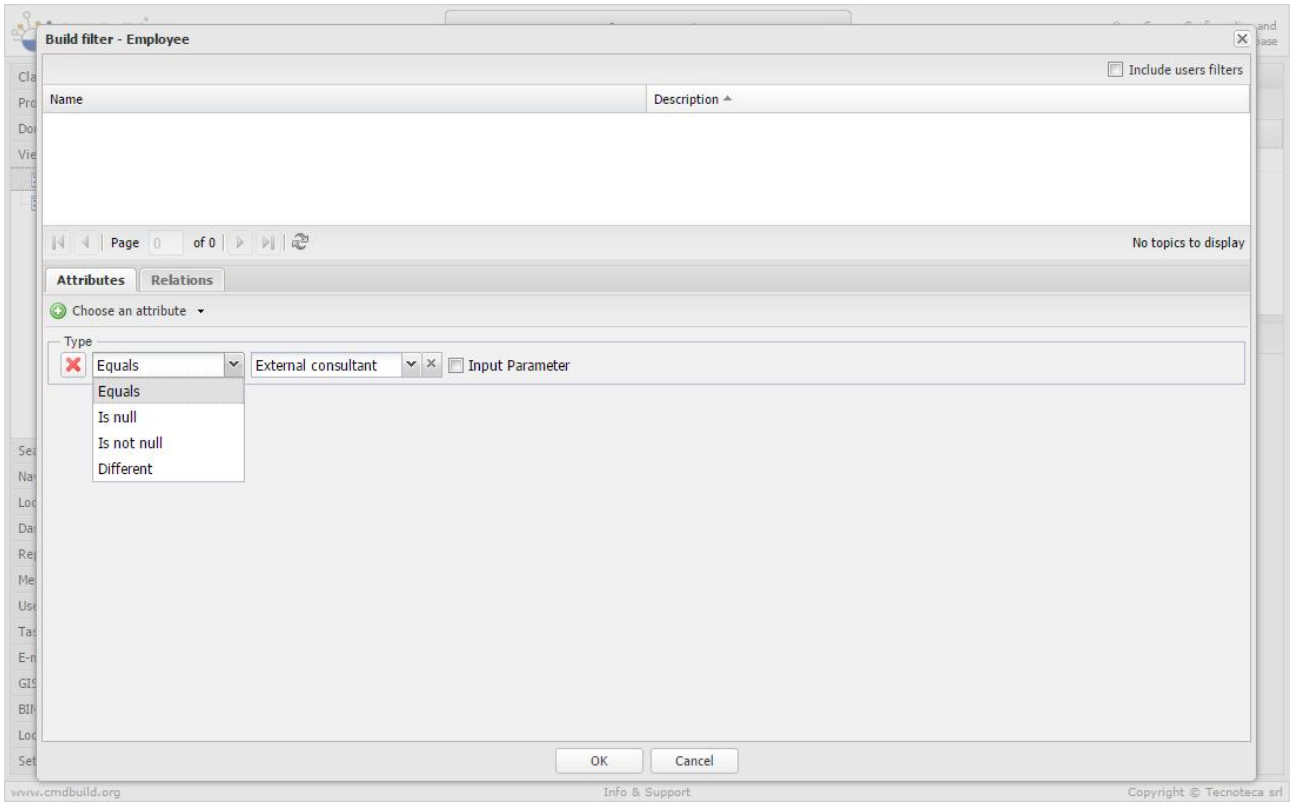
- “Name” - name of the view (e.g. “AssetInUse”)
- “Description” - description of the filter (e.g. “Asset in use”)
- “Source Class” - class which the filter can be applied to (e.g. Asset)

A new view 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” / “Views”.

Below you will find two screenshots that describe the user interface provided for the management of views based on a filter.





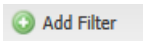
Views based on SQL

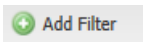
Operators can be provided with data cards obtained by collecting attributes present on different classes.

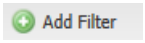
These views typologies can be Read Only, limited to the main card (but relations and history)

From a technical point of view, the SQL query should be encapsulated in a PostgreSQL function, defined within some criteria that will allow to CMDBuild its identification and use. For a description of these criteria, see the paragraph related to the definition of Dashboard charts (Definition of data source - PostgreSQL function).

It's possible to perform the following operations:

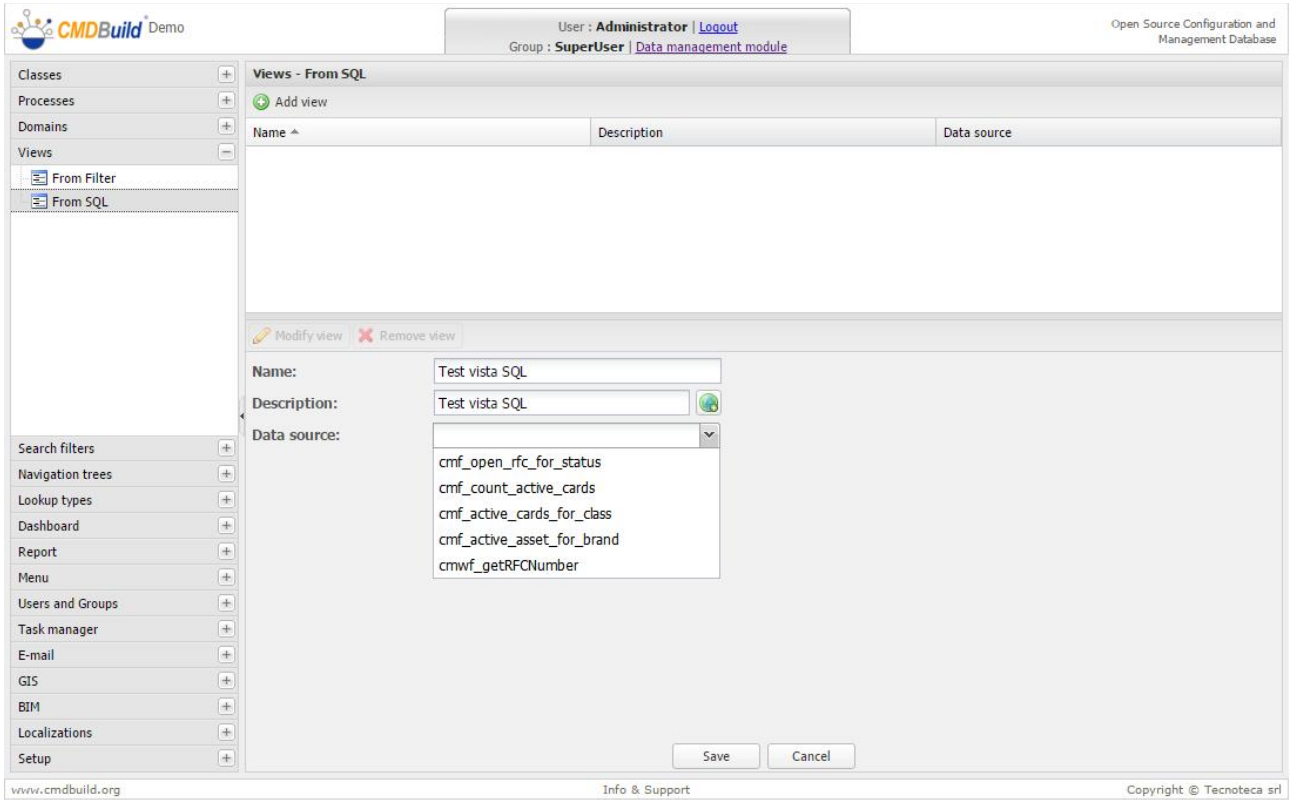
- initiating the creation of a new view 

- editing a pre-existing view 

- deleting a pre-existing view 

To create or modify an SQL view, the following information is required:

- “Name” - name of the view (e.g. “AssetInUse”)
- “Description” - description of the filter (e.g. “Asset in use”)
- “Data source” - PostgreSQL function which the SQL query is defined in



Please note that the PostgreSQL function should satisfy the limitations described in the chapter related to Dashboards, in particular in those pages that carry examples of SQL functions

Also in this case, a new view 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” / “Views”.

Search filters

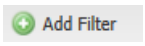
Search filters can be used in the Management Module in order to display only those data cards that comply with the defined criteria.

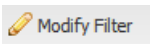
In the Management Module, every user can autonomously save those search filters used to refer to the data cards of a class and find them when accessing to the application.

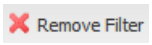
This function of the Administration Module lets the system administrator to create search filters which can be enabled to one or more user groups (they will be then found among those filters available in the Management Module) in one of the following modalities:

- by reusing and cloning a filter defined by a user
- by defining a new search filter, i.e. by setting search criteria on attributes of the specified source class

It's possible to perform the following operations:

- initiating the creation of a new filter 

- edit a pre-existing filter 

- delete a pre-existing filter 

- opening a popup window where you can choose a user filter or create a new filter 
- deleting the current filter criteria 

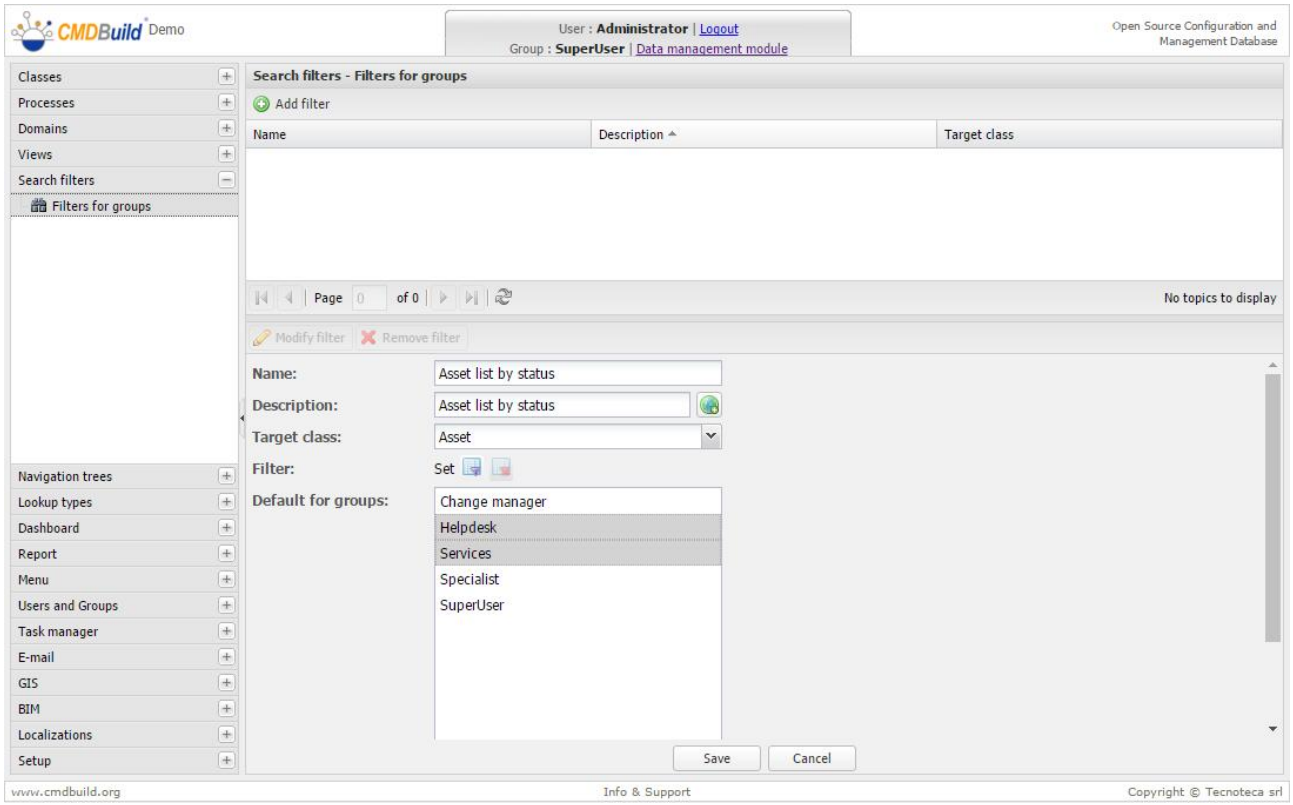
To create or modify a search filter, the following information is required:

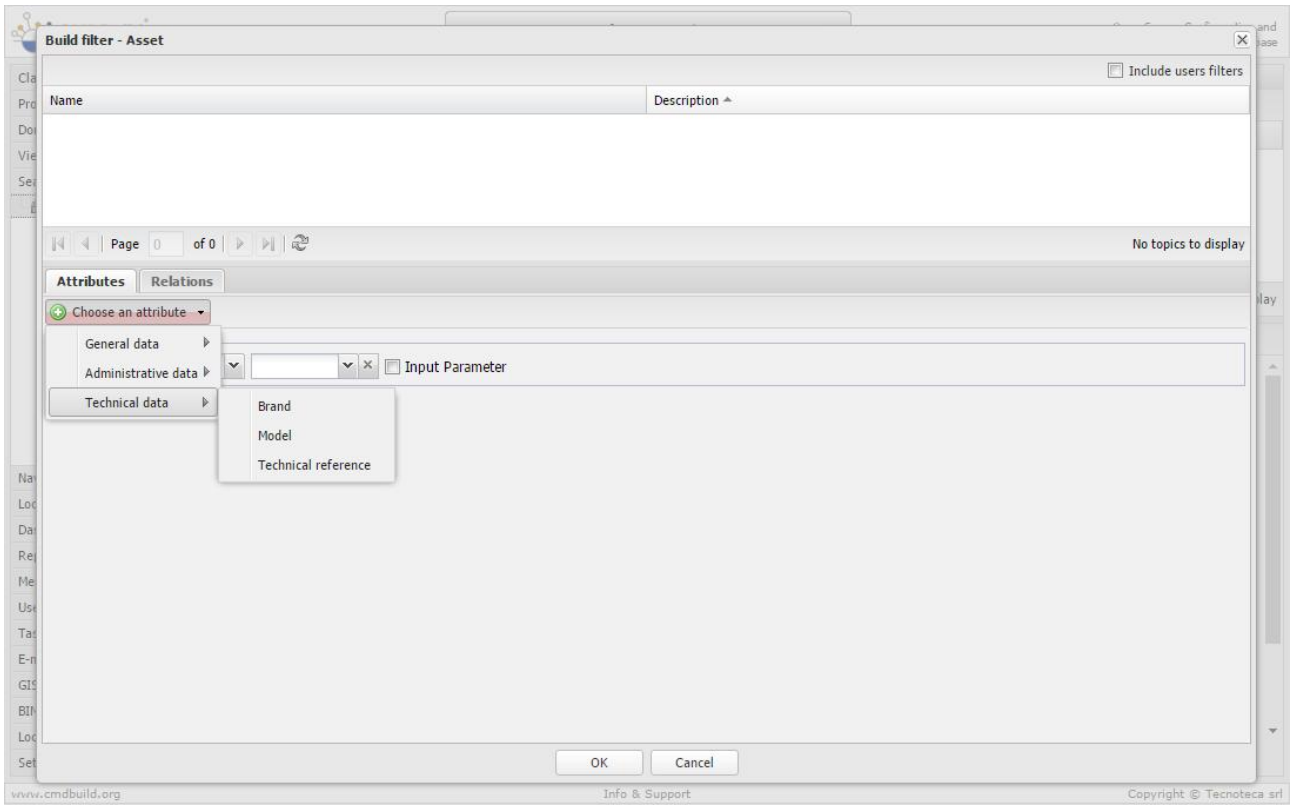
- “Name” - name of the filter (e.g. “PCBrandDell”)
- “Description” - description of the filter (e.g. “PC brand Dell”)
- “Source class” - class which the filter can be applied to (e.g. PC)
- “Default for groups”, field “multiselect” which allows you to specify that the current filter is applied as default filter onto the above mentioned class for the selected groups (see also the TAB "Default filters" of Users and Groups)

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”.

Below you will find two screenshots that describe the user interface provided for the management of search filters.


CMDBuild Demo
User : Administrator | Logout
Group : SuperUser | Data management module
Open Source Configuration and Management Database
Classes
Processes
Domains
Views
Search filters
Filters for groups
Navigation trees
Lookup types
Dashboard
Report
Menu
Users and Groups
Task manager
E-mail
GIS
BIM
Localizations
Setup
Search filters - Filters for groups
Add filter
Name
Description
Target class
Page 0 of 0
No topics to display
Modify filter
Remove filter
Name: Asset list by status
Description: Asset list by status
Target class: Asset
Filter: Set
Default for groups:
Change manager
Helpdesk
Services
Specialist
SuperUser
Save
Cancel
www.cmdbuild.org
Info & Support
Copyright © Tecnoteca srl


Build filter - Asset
Include users filters
Name
Description
Page 0 of 0
No topics to display
Attributes
Relations
Choose an attribute
General data
Administrative data
Technical data
Input Parameter
Brand
Model
Technical reference
OK
Cancel
www.cmdbuild.org
Info & Support
Copyright © Tecnoteca srl

Navigation trees

Properties tab

The navigation trees in CMDBuild are used to make available a hierarchical view of the information in certain management functions, among these:

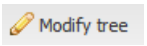
- the selection of an element through tree interface in the widget “Navigation tree”
- Relations Graph
- GIS navigation
- BIM navigation

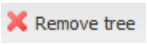
The navigation trees will be also used in the following versions of CMDBuild in order to extend the functions of the Relation Graph.

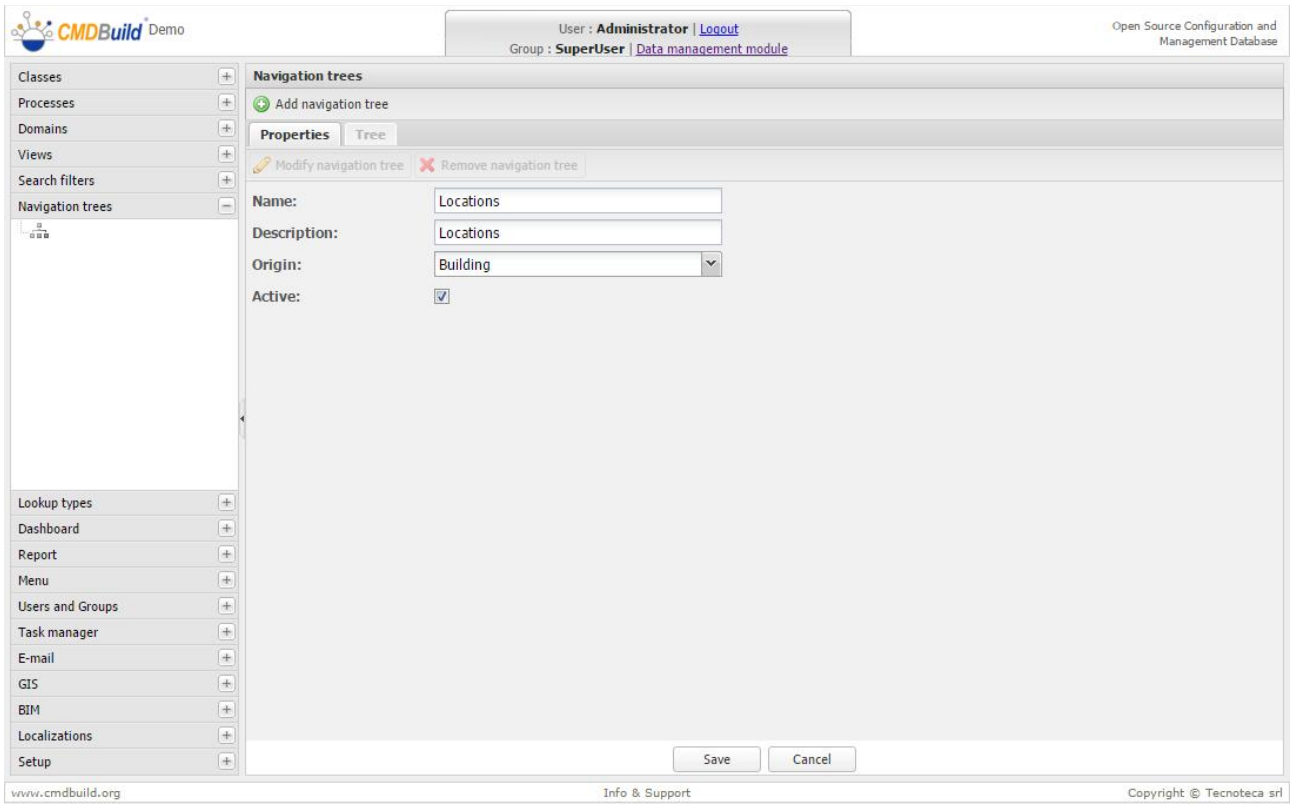
The Properties tab allows to create new navigation trees and modify the description of a preexisting tree.

It's possible to perform the following operations:

- initiating the creation of a new navigation tree 

- modifying a preexisting navigation tree 

- deleting a preexisting navigation tree 



In order to create a new navigation tree, the following information is required:

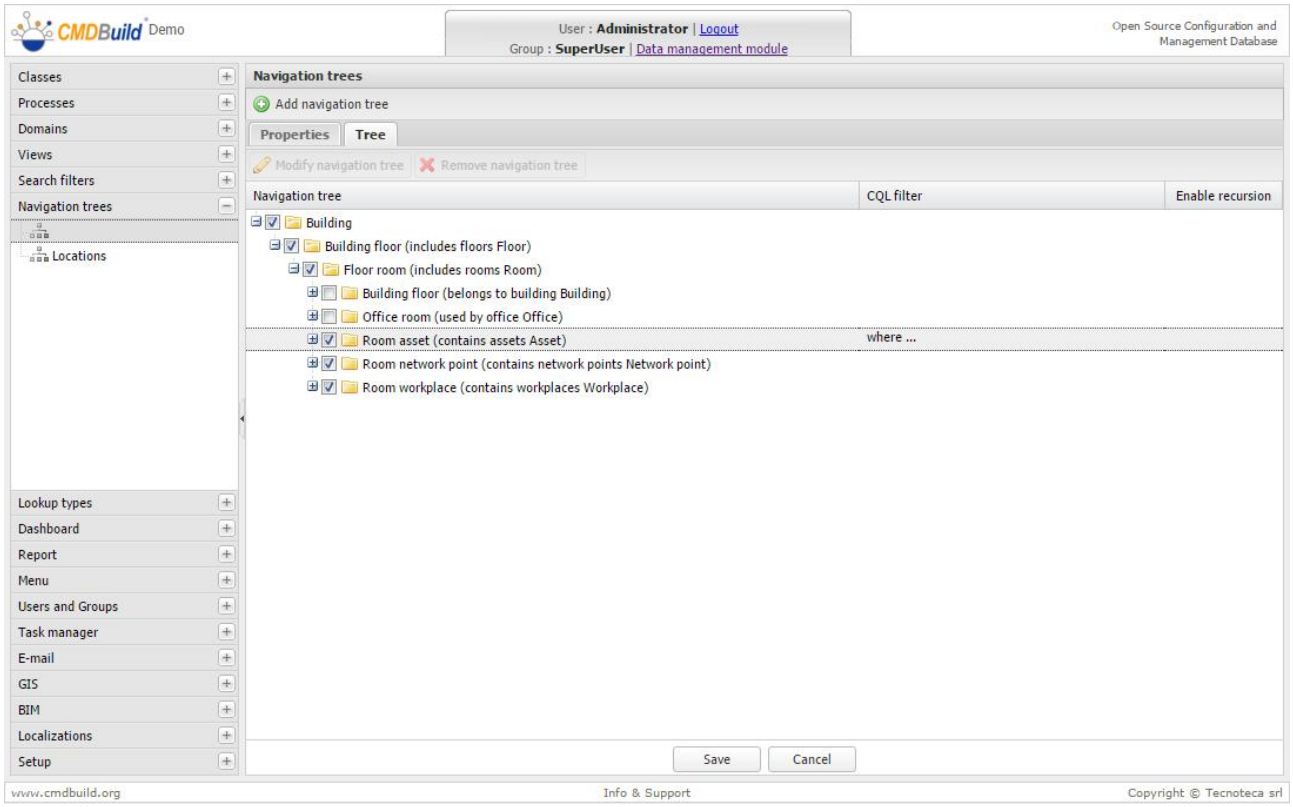
- “Name”, name of the navigation tree
- “Description”, description of the navigation tree
- “Origin”, root class of the tree
- “Active”, if the tree is active

Tree tab

Through the Tree tab you can visually build the current navigation tree.

For each node you can also:

- define a possible CQL filter in order to limit the number of selectable children
- specify that another domain on the same class must be followed with a recursive approach



Lookup

As previously described, one of the data types available in CMDBuild is the "lookup" type, ie an attribute that shows up as a select field with predefined options.

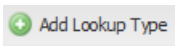
Therefore, the user must select an option from the list of options available (for example, monitor type could be "CRT", "LCD" or "Plasma" - monitor brand could be "HP", "Dell", "Philips" etc.)

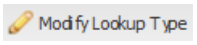
You can also create multi-level lookup attributes (for example: macrocategory => category)

Properties tab

Using the "Properties" tab you can manage the Lookup lists.

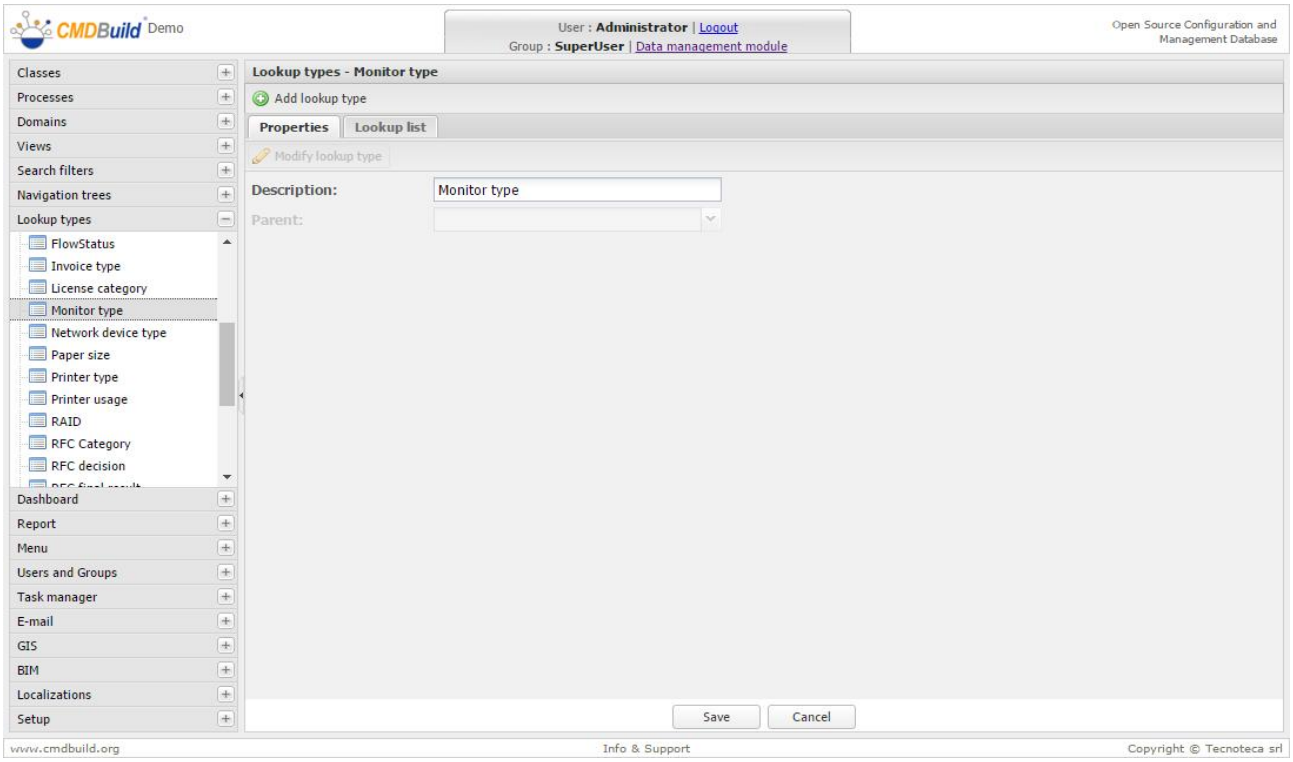
It's possible to perform the following operations:

- add a new lookup list 

- edit an existing lookup (description only) 

To create or modify a Lookup list, the following information is required:

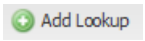
- "Description" - the name of the list (for example, "Monitor Type" or "Brand")
- “Parent” - the parent of the current list (can be set only at creation)

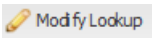


Lookup list

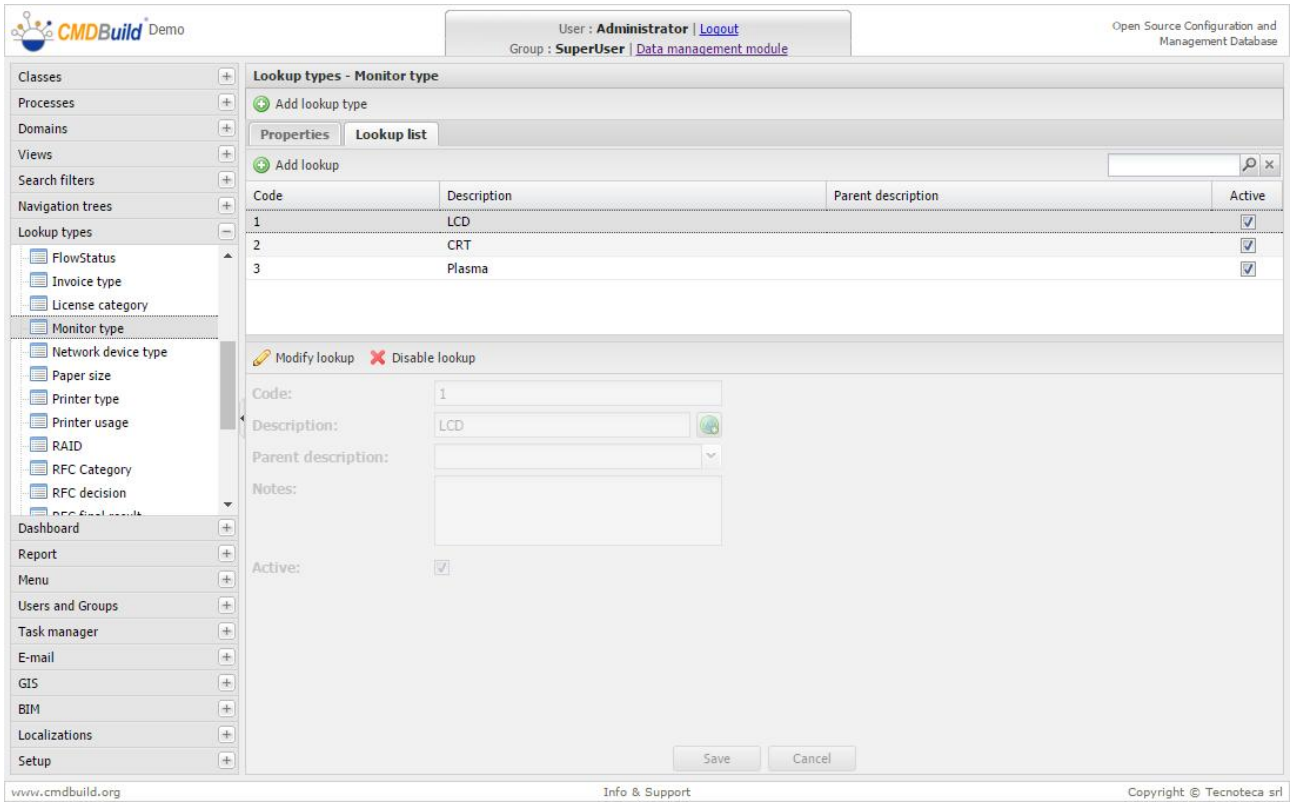
With the "Lookup list" tab you can create new entries in the list or edit existing ones.

It's possible to perform the following operations:

- create a new entry 

- edit an existing entry 

- delete an entry 



To define a new entry you need to specify the following information:

- "Code" of the new entry
- "Description" of the new entry
- "Parent description" - select an entry of the parent list, if available
- "Notes"- any additional information about the entry
- "Active" - indicates whether the item is active or has been deleted (logical deletion)

The entries order can be modified by moving the lines in the grid (drag and drop).

Dashboard

CMDBuild allows you to configure in the system "dashboards" pages, which singularly contain more charts of different typologies; then they will be consulted in the Mangaement Module (both in a new specific accordion menu and in the Navigation Menu).

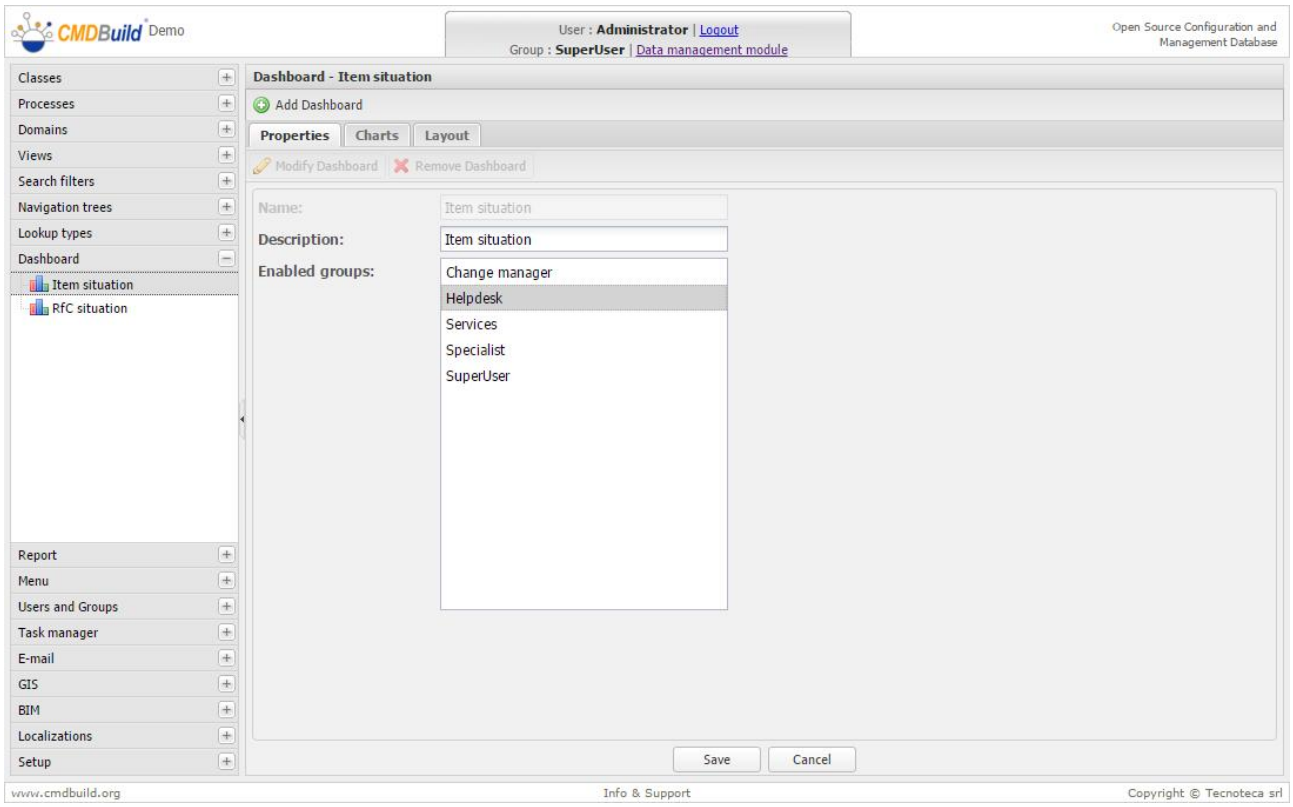
Every dashboard can be addressed to a different typology of aspects which should be controlled: asset situation, service desk performances, cost allocation, etc.

Every dashboard is made up of a number of charts which can adopt different typologies: pie, bar, line, gauge chart.

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.

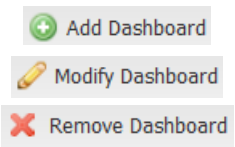
Properties tab

The "Properties" tab allows you to create a new dashboard showing names and groups able to display it.



There is the possibility of performing the following operations:

- create a new dashboard
- edit a pre-existing dashboard
- delete a pre-existing dashboard

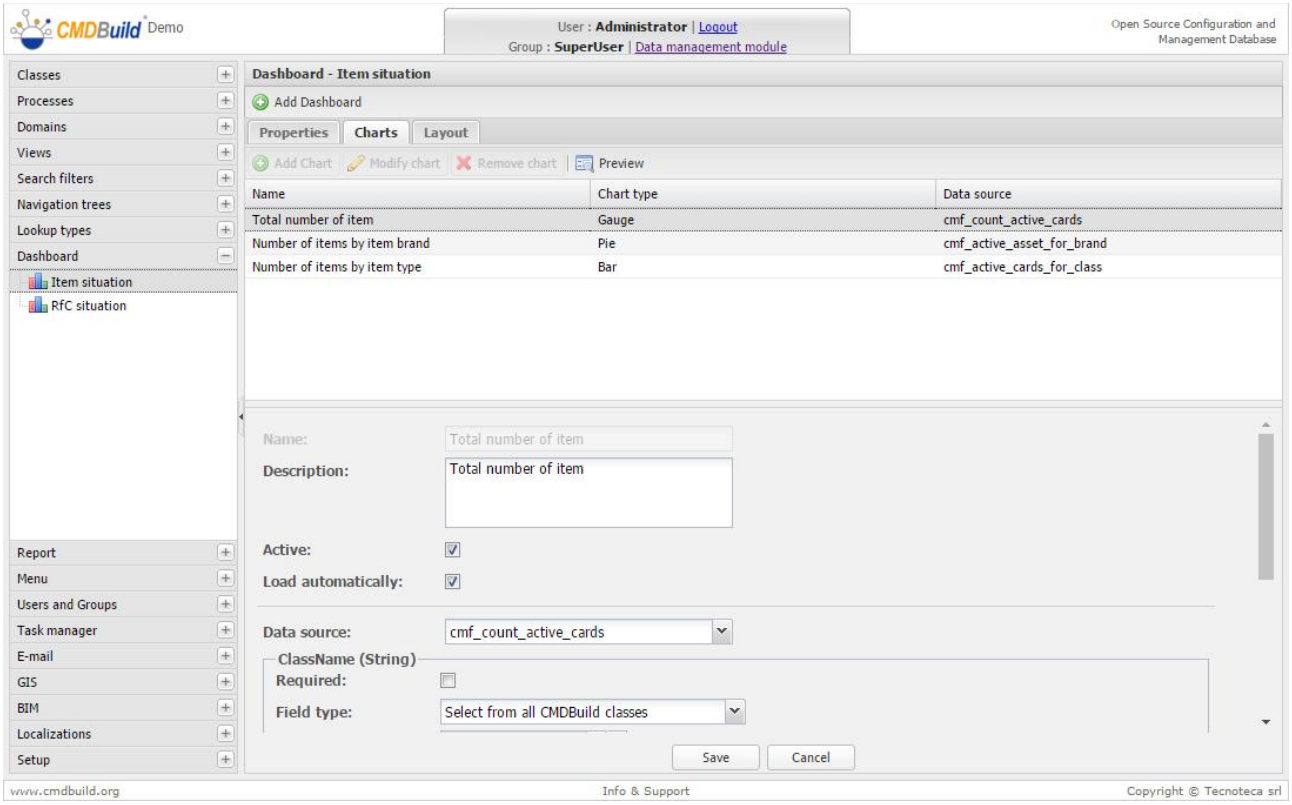


To create a new dashboard you need to specify the following information:

- “Name”
- "Description"
- "Enabled groups" - the groups (along with the system administrator) allowed to display the dashboard

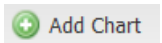
Chart TAB

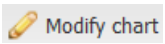
The "Charts" TAB allows you to set up the charts you want to display in the selected dashboard.

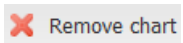


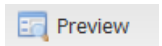
There is the possibility of performing the following operations:

- add a new chart
- edit a pre-existing chart
- delete a pre-existing chart
- show a preview of the chart





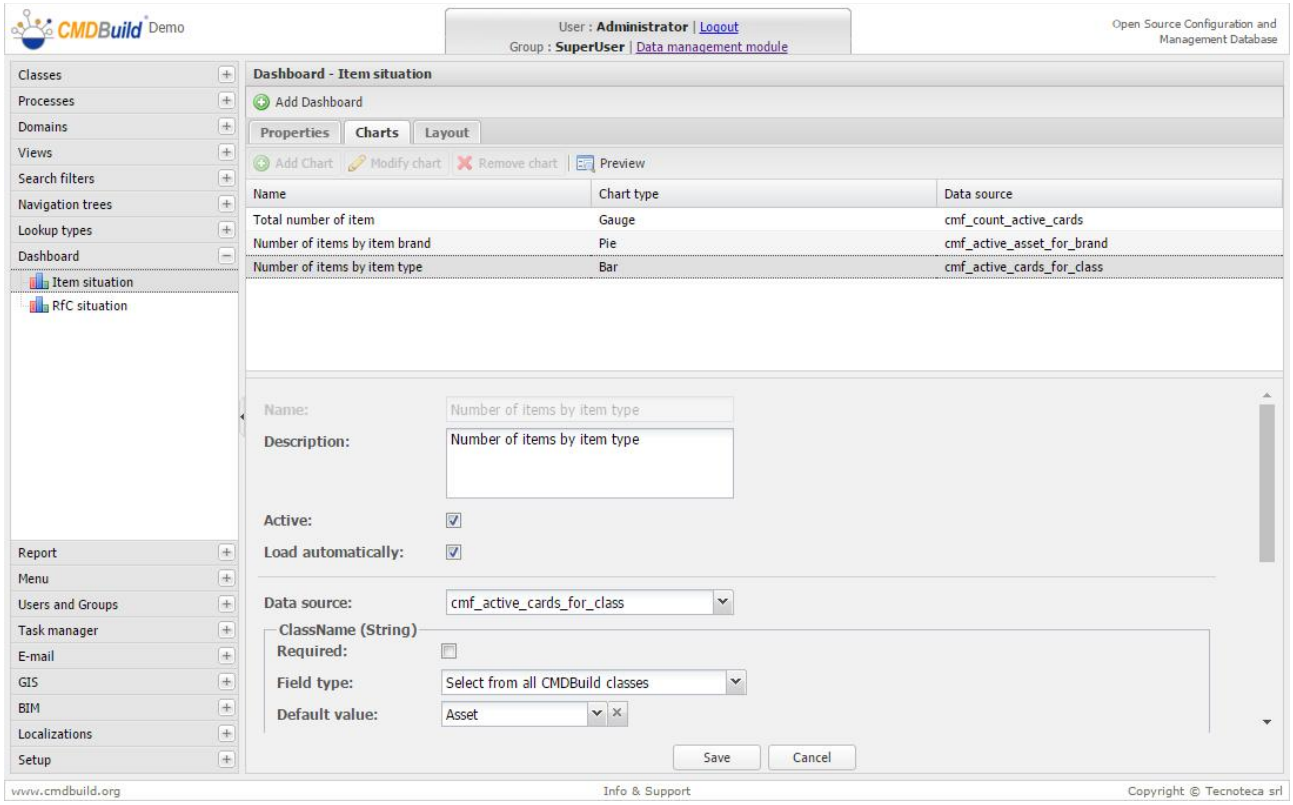


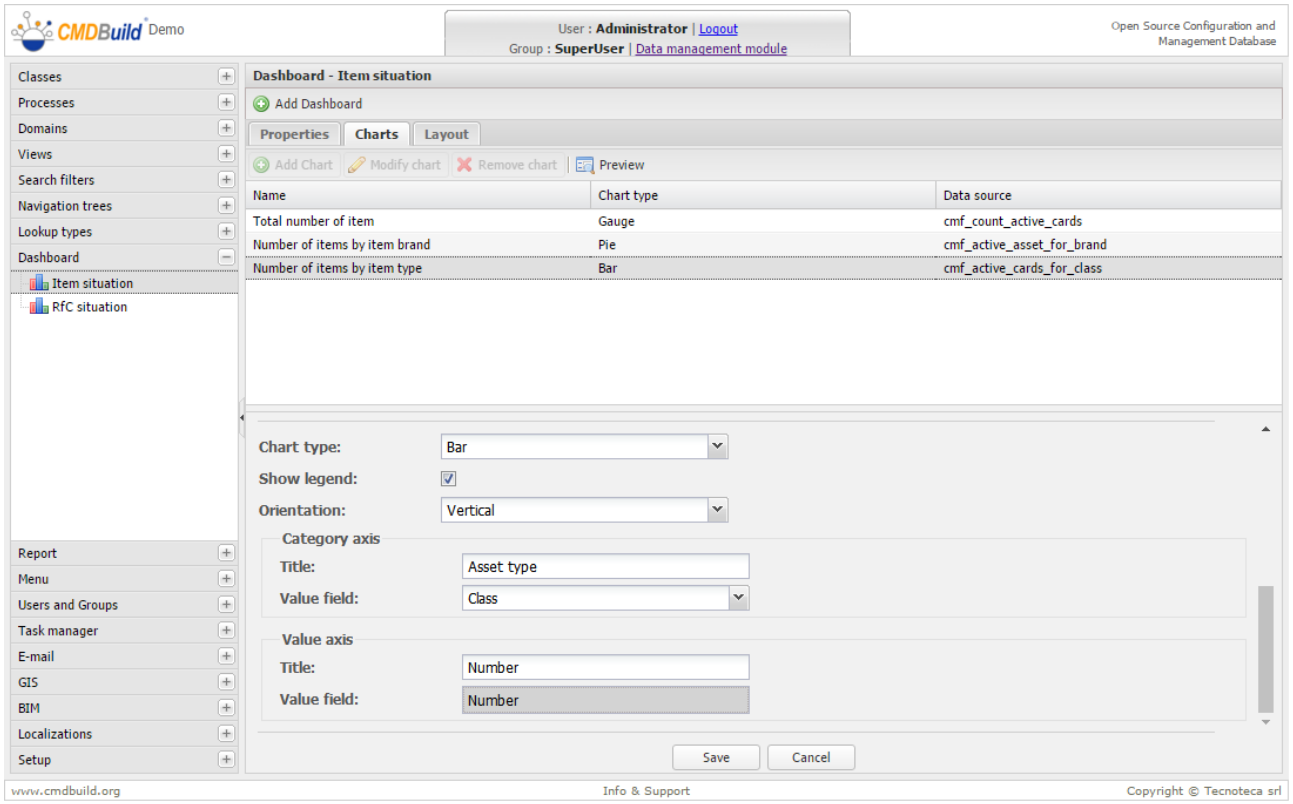


To set up each chart you have to:

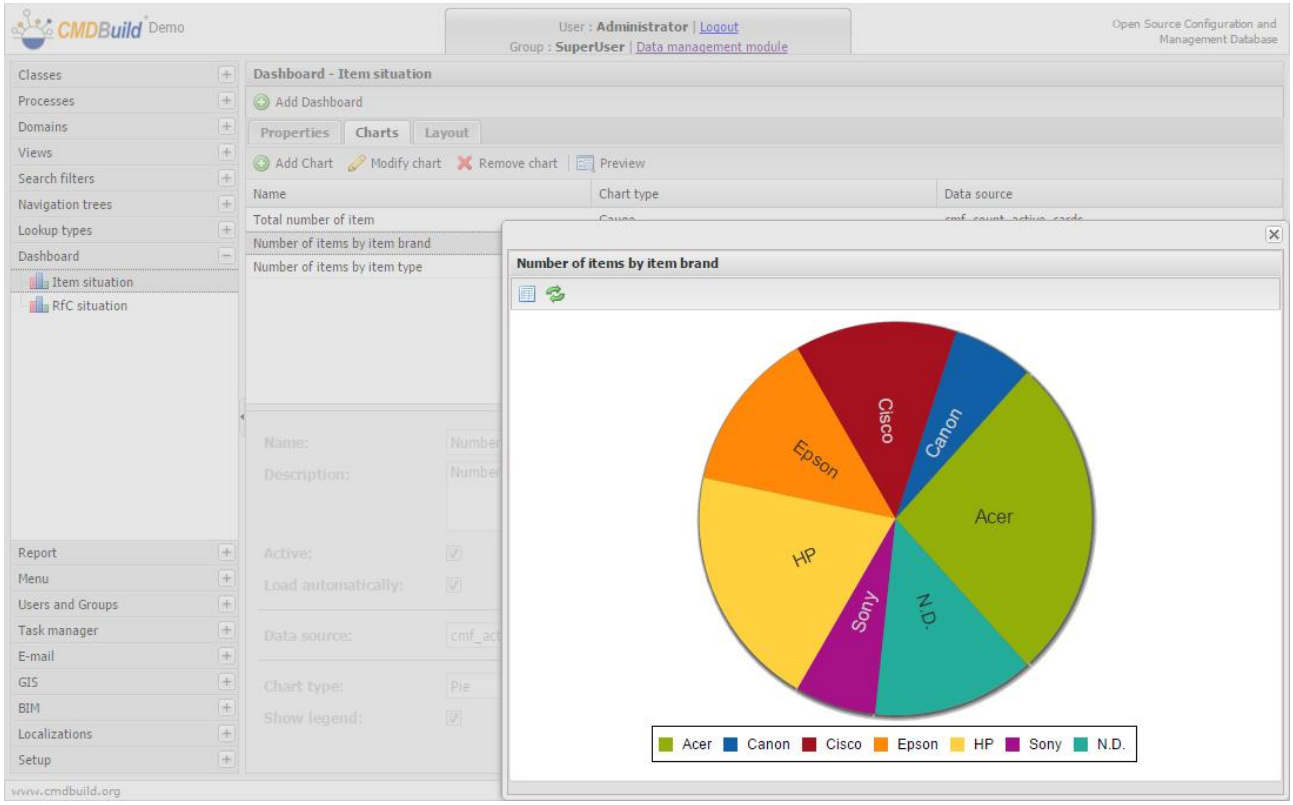
- show some basic information
- highlight the data source (a PostgreSQL function which must have already been created in the database with the instructions shown below), in which the system will read the data in order to populate the chart
- set up the mapping among the output parameters of the PostgreSQL function and the input parameters provided by the chart typology (showing also whether it can be input by keyboard and the possible widgets that should be used)
- specify the chart typology (pie, bar, line, gauge), which determines the request of further specific parameters

The following example (divided into two screenshot) defines an histogram chart:





Here's an output sample of the "Preview" function:



The detailed information required for the configuration of a chart include:

- “Name” of the new chart
- “Description” of the new chart
- “Active”, indicates whether the chart is active has been disabled
- “Load automatically”, indicates whether it can be immediately shown without the input of parameters through keyboard
- “Data source” shows the list of PostgreSQL functions available in the definition of the chart
- list of data source parameters, with the possibility for each of them to indicate:
  - if it is mandatory
  - type (string, integer, decimal, data)
  - in case of string parameters a subtype is requested among: free string, name of a CMDBuild class, current user, current group
  - in case of integer parameters a subtype is requested among: free integer, "Id" of a CMDBuild class, "Id" of a lookup with choise of the concerning lookup, "Id" of a card with choise of the concerning class
  - the default value (free value or value from the list, according to the type of the datum)
- “Chart type”, that can adopt one among the following values: pie, bar, line, gauge
- “Show legend” (if necessary it will be located under the chart)

- other parameters specific to each type of chart:
    - for pie charts:
        - value field (for the dimensioning of the areas)
        - label field (cited in the areas)
    - for bar charts:
        - 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 (for the bar dimensioning, perhaps with more overlapping series)
    - for line charts:
        - orientation (horizontal or vertical)
        - title of category board (label)
        - value of category board (information represented by each point on the line)
        - title of value board (label)
        - value / values value board (for the line height, perhaps with more overlapping series)
    - for gauge charts:
        - the maximum scale value
        - the minimum scale value (zero if not specified)
        - number of discretization intervals
        - foreground color
        - background color
        - value field (in order to define the indicator shown)

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”, “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”

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

Here below there are two samples of right PostgreSQL functions, used in the demo database:

```
CREATE OR REPLACE FUNCTION cmf_active_cards_for_class(IN "ClassName" character varying,
OUT "Class" character varying, OUT "Number" integer)
  RETURNS SETOF record AS
$BODY$
BEGIN
    RETURN QUERY EXECUTE
        'SELECT _cmf_class_description("IdClass") AS "ClassDescription", COUNT(*)::integer
AS "CardCount"' ||
        '    FROM ' || quote_ident($1) ||
        '    WHERE "Status" = ' || quote_literal('A') ||
        '        AND _cmf_is_displayable("IdClass")' ||
        '        AND "IdClass" not IN (SELECT _cm_subtables_and_itself(_cm_table_id(' ||
quote_literal('Activity') || ')))'
        '    GROUP BY "IdClass"' ||
        '    ORDER BY "ClassDescription"';
END
$BODY$
  LANGUAGE plpgsql VOLATILE
  COST 100
  ROWS 1000;
ALTER FUNCTION cmf_active_cards_for_class(character varying) OWNER TO postgres;
COMMENT ON FUNCTION cmf_active_cards_for_class(character varying) IS 'TYPE: function';


CREATE OR REPLACE FUNCTION cmf_count_active_cards(IN "ClassName" character varying, OUT
"Count" integer)
  RETURNS integer AS
$BODY$
BEGIN
    EXECUTE 'SELECT count(*) FROM ' || quote_ident("ClassName") || ' WHERE "Status" = ' ||
quote_literal('A') INTO "Count";
END
$BODY$
  LANGUAGE plpgsql VOLATILE
  COST 100;
ALTER FUNCTION cmf_count_active_cards(character varying) OWNER TO postgres;
COMMENT ON FUNCTION cmf_count_active_cards(character varying) IS 'TYPE: function';
```

Layout TAB

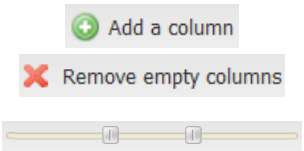
When the charts of the dashboard are defined, it will be possible to set their layout.

In particular you can choose whether to distribute them on one, two or three columns, and to move the present charts among the columns (using the “drag and drop” method).

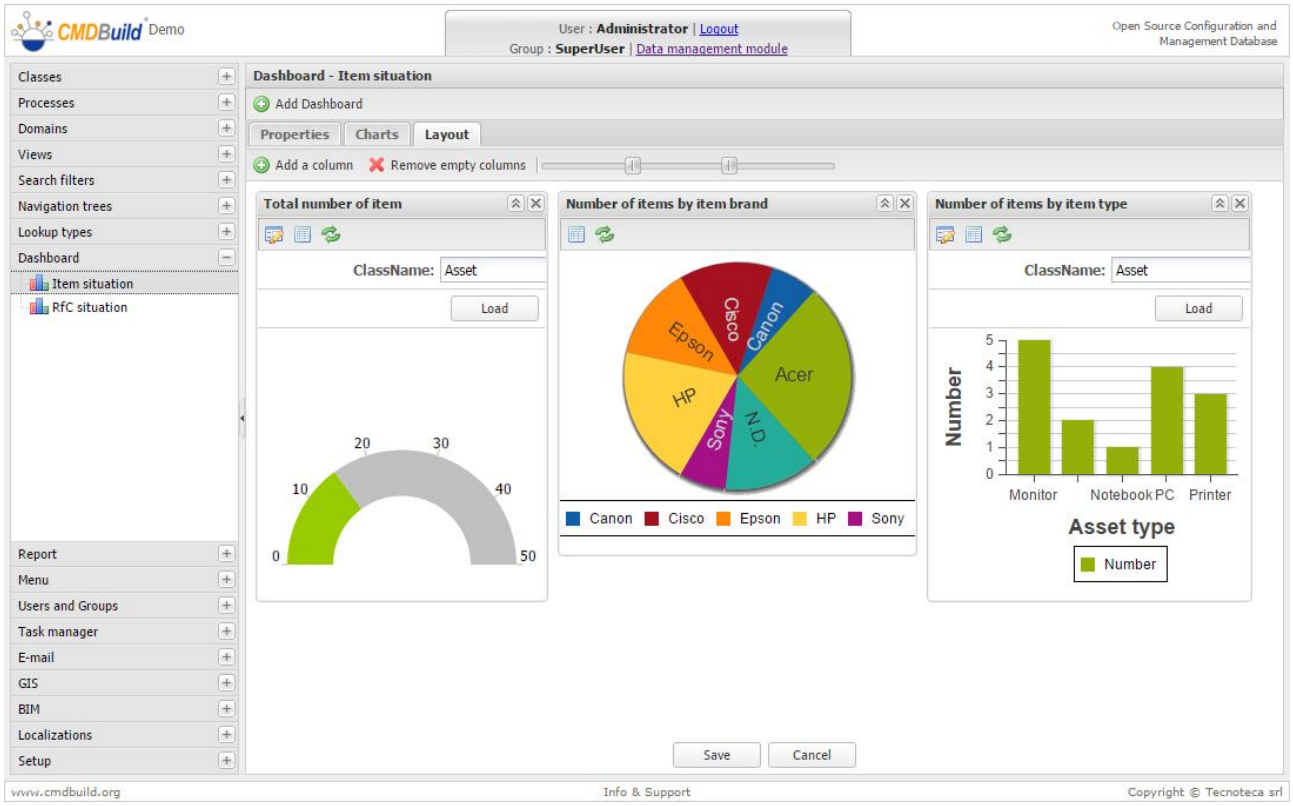
It is also possible to move a chart from a dashboard to another by selecting it and dragging it on the name of its new dashboard (in the accordion menu on the left).

The available functions are as follows:

- add a new column
- remove a column, if it is empty
- personalize the distribution of the room among the provided columns



Here's a sample of a 3-column layout:



Report

CMDBuild allows you to import custom reports designed with external tools, and then to include them in the Management Module.

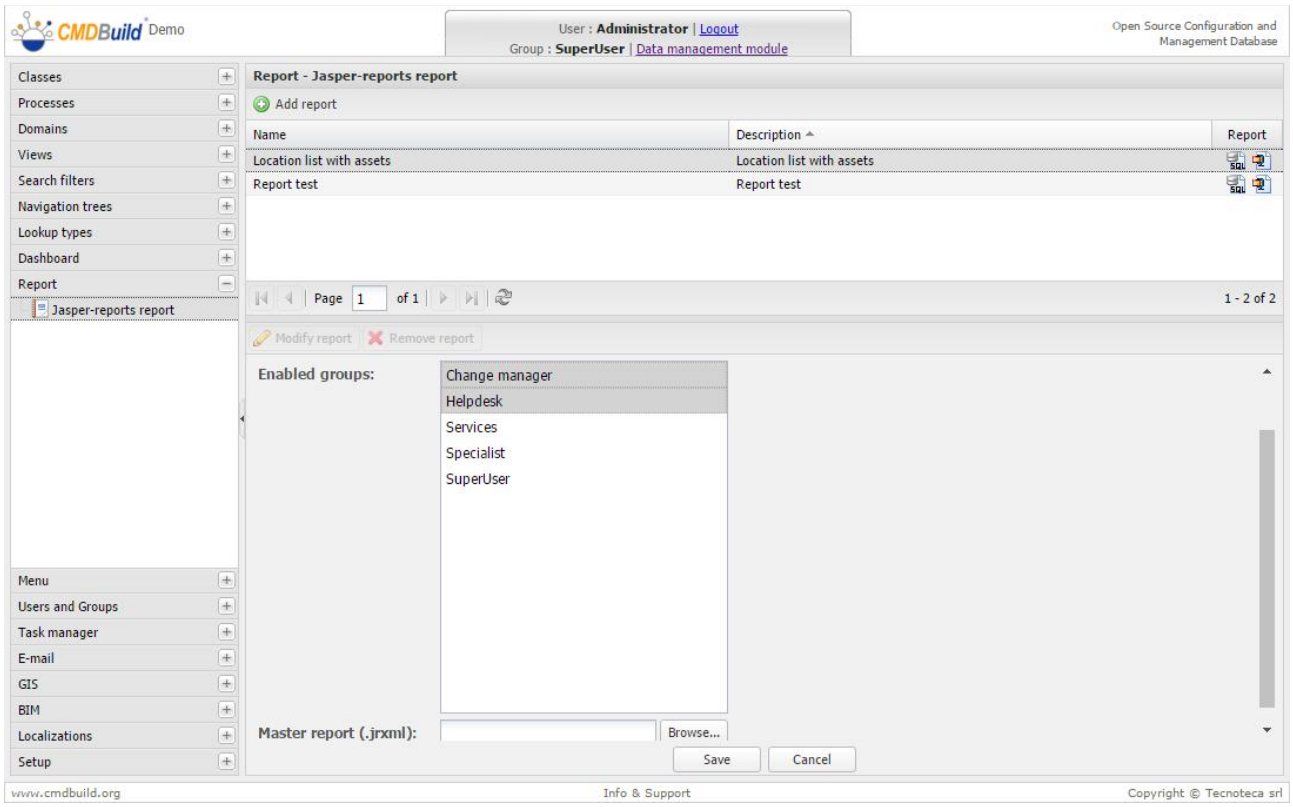
JasperReports

CMDBuild is able to import reports created with iReport (visual editor) which is part of the JasperReports open source project.

iReport features include:

- text formatting options (font, text alignment, space, colour, etc.)
- standard elements definition (header, footer, column headings, summary, etc.)
- data grouping
- expressions evaluation
- computed fields
- advanced subreports management
- images and graphics
- PDF, HTML, XLS and CSV formats

The import function handles also reports containing subreports and images, whose upload is managed using a wizard.

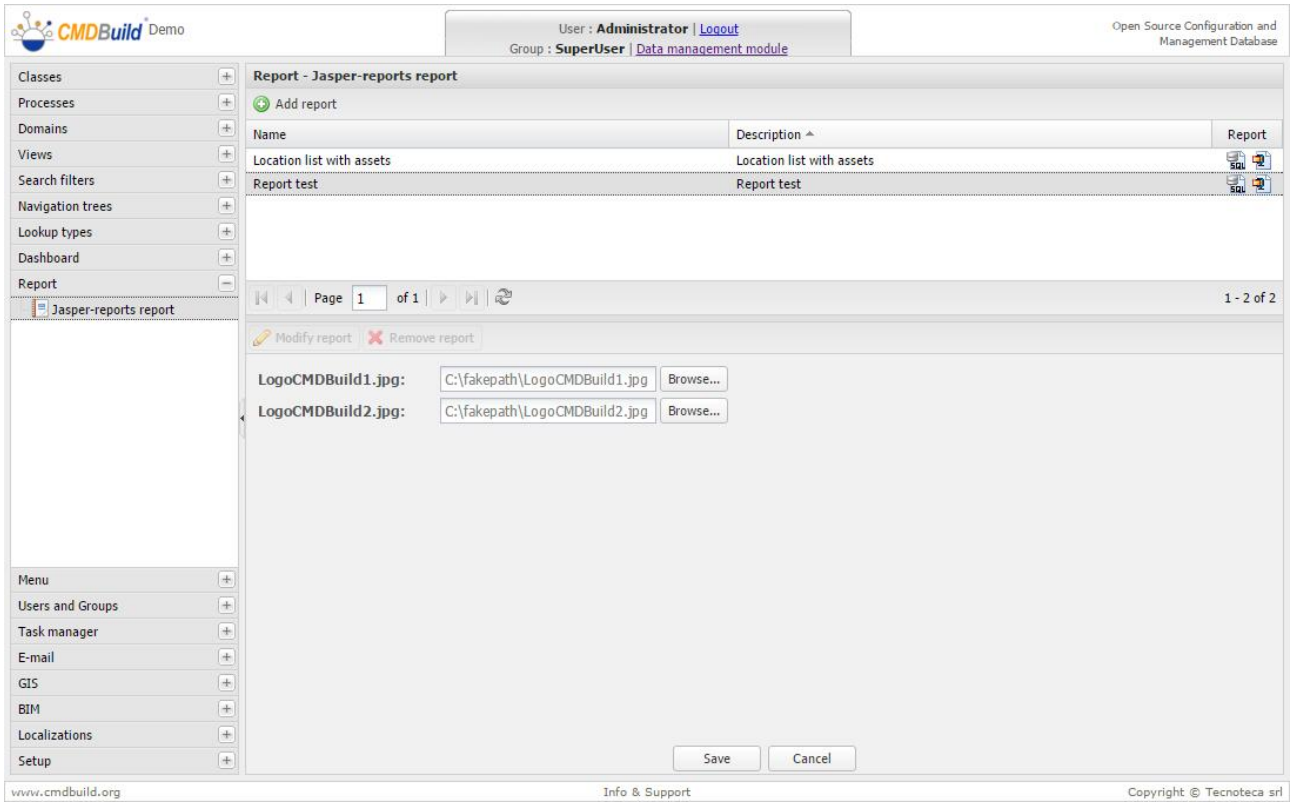


To create a new report you have to insert the following information:

- "Report name" - the name will appear in the list of available reports
- "Report description" - the description will appear in the list of available reports
- "Enabled groups" - the groups (along with the system administrator) allowed to execute the report
- "Master report" - the template file created with iReport

If the master report contains subreports or images (both managed by iReport as external files), the administrator will be asked to upload the required files.

The report you find in the demo instance includes two logos whose images are required:



Once completed the import, the report can be executed using the Management Module at the Report menu.

Each report can also be inserted into the navigation menu of the groups qualified to use it. Then it can be executed directly from that menu.

Menu

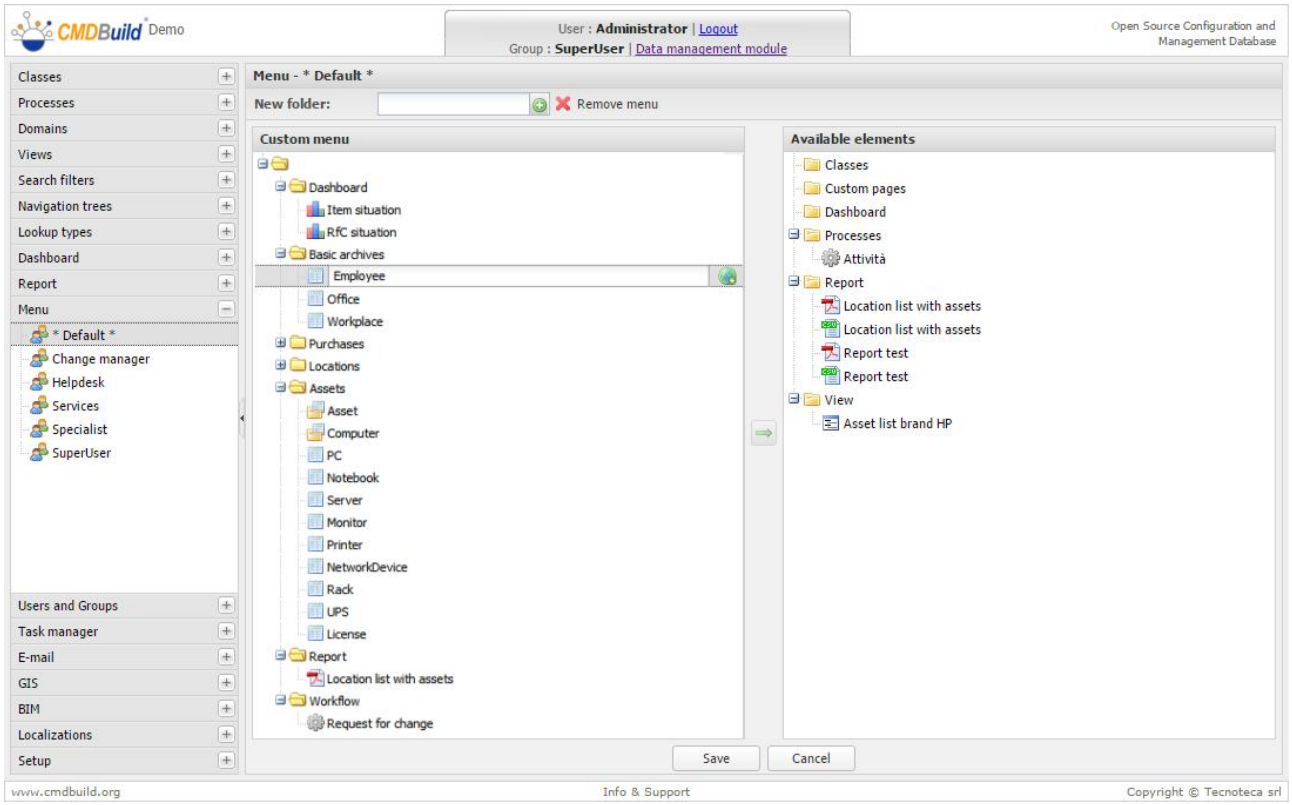
Custom menus

The system allows you to define custom menus for all groups or for specific groups.

Each menu consists of system elements (classes, processes, views, reports) and custom folders (used to organize entries).

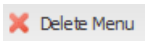
These menus are available in the Management Module as the main navigation menu (the one configured for the login group, the "default" otherwise).

Users with no custom navigation menu can still access the Management Module, but the system will show only classes for which they have read or write permissions.



It's possible to perform the following operations:

- edit the menu for the selected user group
- delete the menu for the selected user group



The setup of a menu includes the following operations:

- add a 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 

With a single operation you can insert a superclass containing subclasses.

Once confirmed the changes, the menu will be available in the Management Module (logout / login to refresh).

User groups

CMDBuild permissions are based on:

- user groups
- group permissions on classes, views and search filters (permissions on dashboards and reports are instead managed in the related sections)
- user - group associations

Therefore it's possible to:

- set user groups with specific permissions for each class defined in the system, also in rows and columns
- 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 personalized setups of the user interface, removing some standard functionalities of CMDBuild and simplifying the interaction interface for some users' groups with less technical skills.

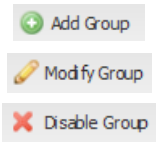
This chapter describes the groups management, the placement in the pre-existent users' group included, while the next deals with the users management.

Properties tab

The Properties tab allows you to create and edit groups.

There is the possibility of performing the following operations:

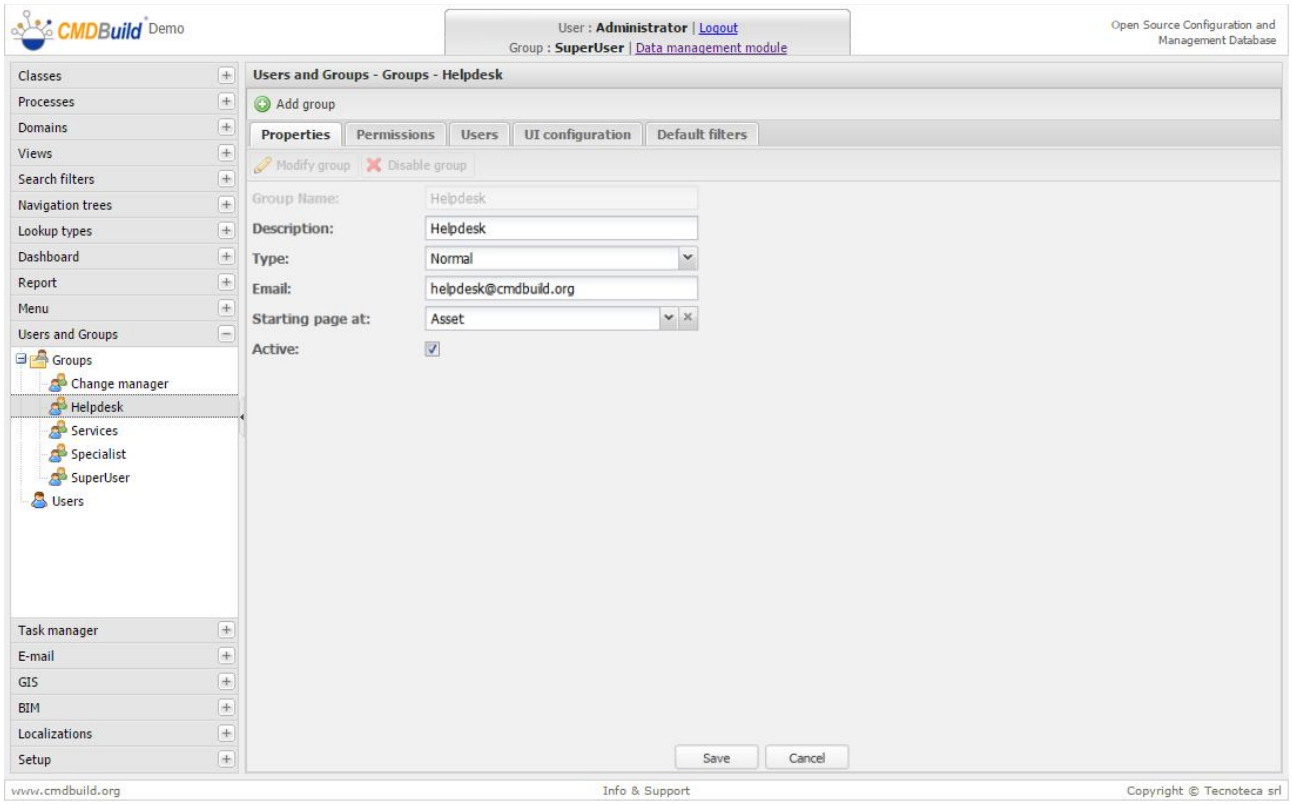
- add a new group
- edit an existing group
- disable a group



To create a new group you have to insert the following information:

- "Group name"
- "Description"
- “Type”: Normal (normal user of the application), Administrator (which can access the Administrator Module with complete permissions to manage and configure the system), limited Administrator (which can access the Administrator Module, except for the change functions of the data model)
- “Email”
- “Starting page at” - the default startup page when entering the Management Module
- “Is active”, indicates whether the group is active or has been deleted (logical deletion)

Here's an example of screenshot.


CMDBuild Demo
User : Administrator | Logout
Group : SuperUser | Data management module
Open Source Configuration and Management Database
Classes
Processes
Domains
Views
Search filters
Navigation trees
Lookup types
Dashboard
Report
Menu
Users and Groups
Groups
Change manager
Helpdesk
Services
Specialist
SuperUser
Users
Task manager
E-mail
GIS
BIM
Localizations
Setup
Users and Groups - Groups - Helpdesk
Add group
Properties
Permissions
Users
UI configuration
Default filters
Modify group
Disable group
Group Name:
Helpdesk
Description:
Helpdesk
Type:
Normal
Email:
helpdesk@cmdbuild.org
Starting page at:
Asset
Active:
Save
Cancel
www.cmdbuild.org
Info & Support
Copyright © Tecnoteca srl

Permissions tab

The Permissions tab allows you to define permissions for each class / process / view / filter / custom page defined in the system.

There is the possibility of performing the following operations:

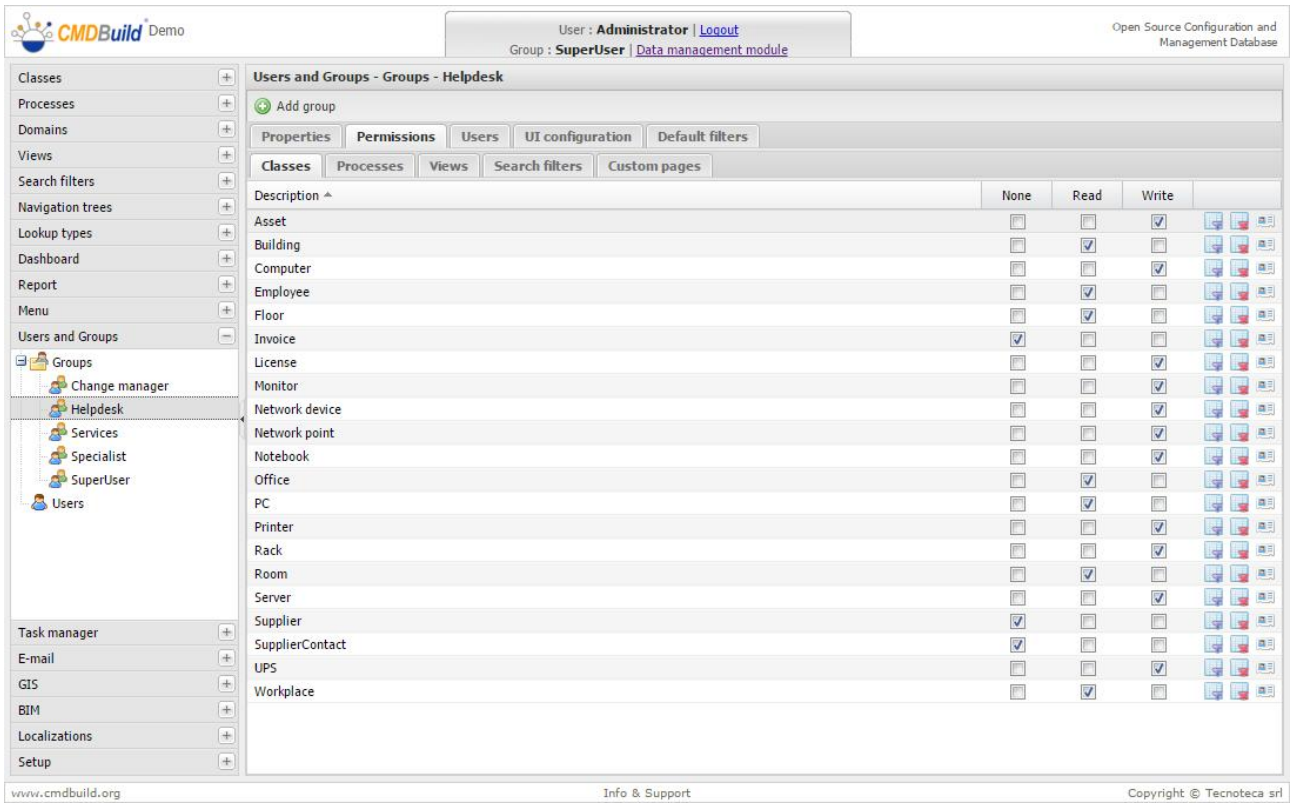
- for each kind of element (class / view / search filter), set / modify on the grid the kind of permission that accepts the values: none / read / write

- only for classes and processes, open a popup window where you can define a filter on the class rows (if necessary choosing a user filter) or columns

- only for classes and processes, delete the current filter criteria

TAB Classes

Here's the page related to the management of class permissions (class tab):



You can change permissions by clicking on none / read / write checkboxes.

If for the current group you want to reduce the access permissions to rows or columns of a class, you have to use the first icon placed at the end of every row in the scheme.

In this way you will access a popup window that presents two TABs called "Privileges on rows" and "Privileges on columns".

Classes: Privileges on rows

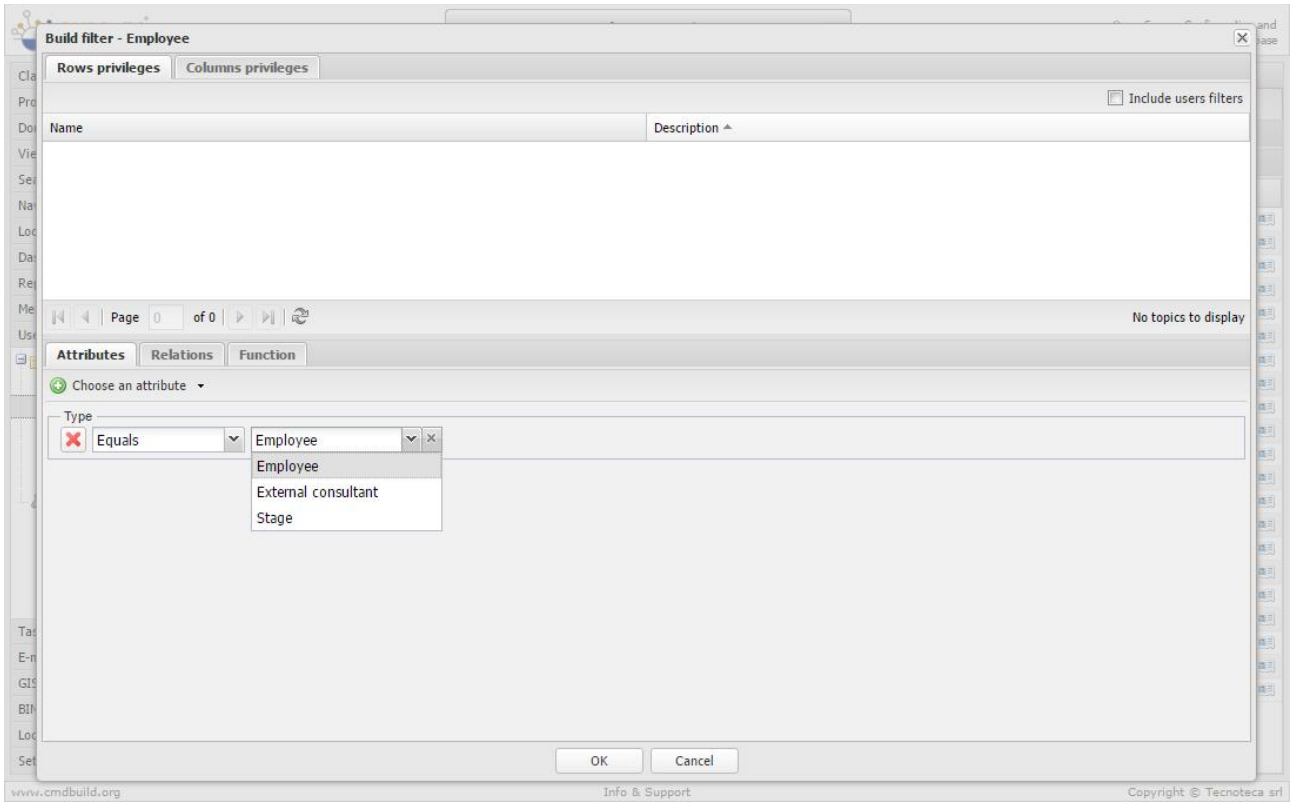
The management of Privileges in rows can be carried out in two ways:

- by reusing and cloning a pre-existing filter defined by the administrator or by a user (after ticking off the check)
- by defining a new search filter

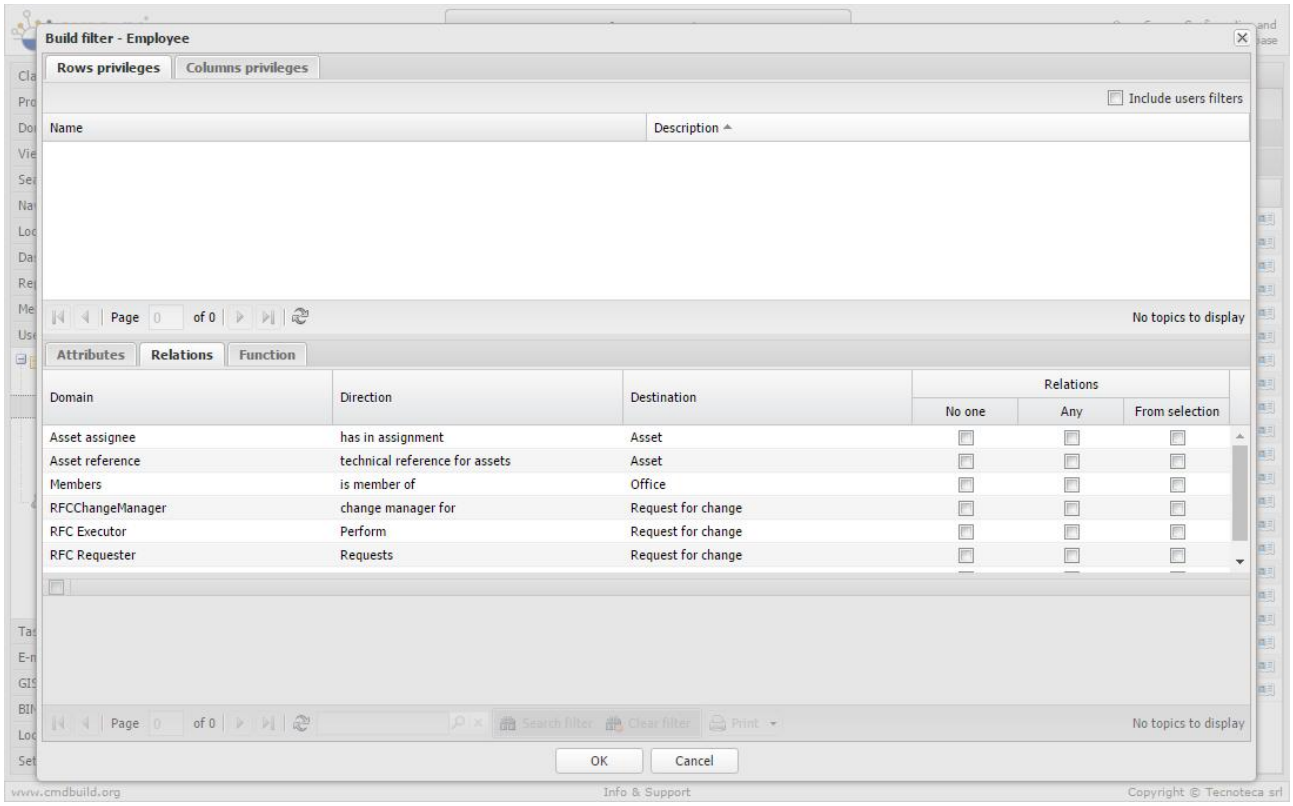
A new search filter, in turn, can be defined in three ways:

- by setting search criteria on the attributes of the specified source class (Attributes tab)
- by setting search criteria on the relations of the specified source class (Attributes tab)
- by using a predefined PostgreSQL function, that should be created according to the criteria described in the chapter related to Dashboards (Function tab)

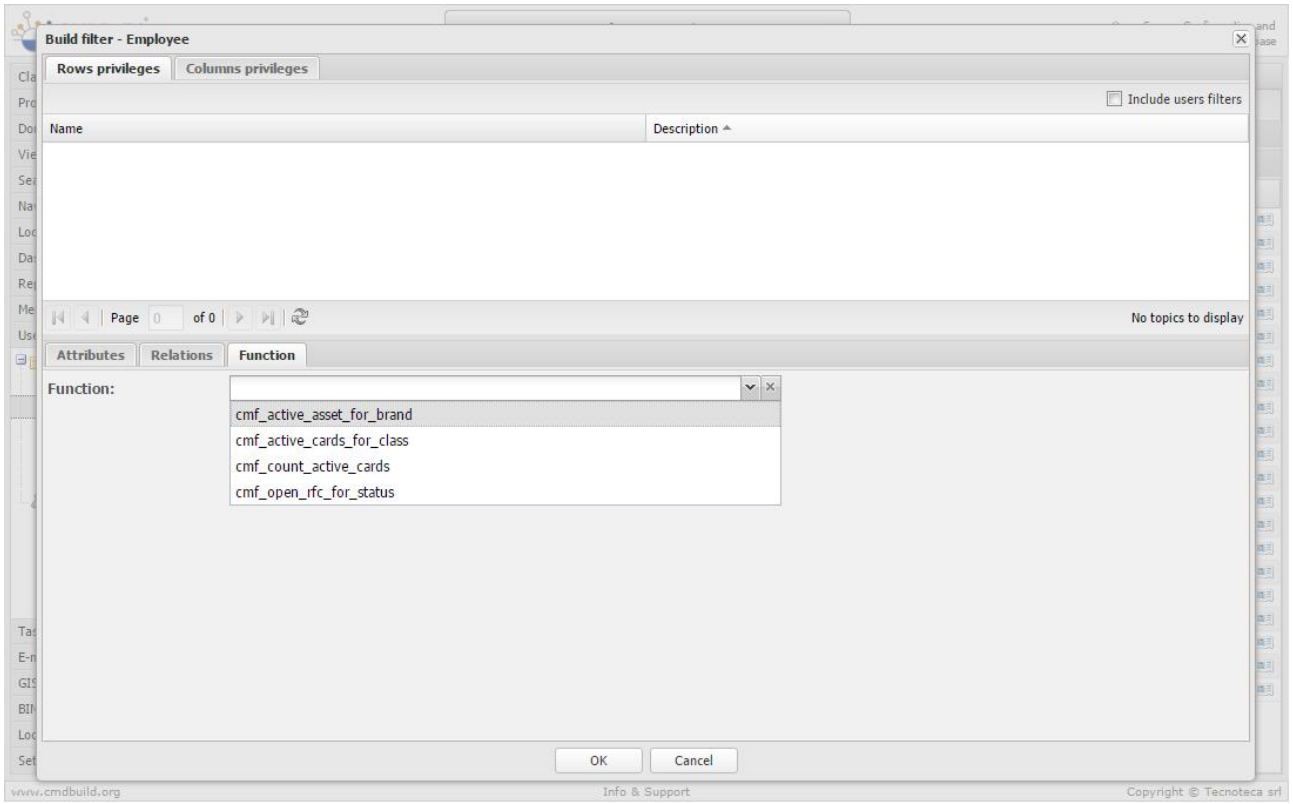
Attributes tab



“Relations” TAB



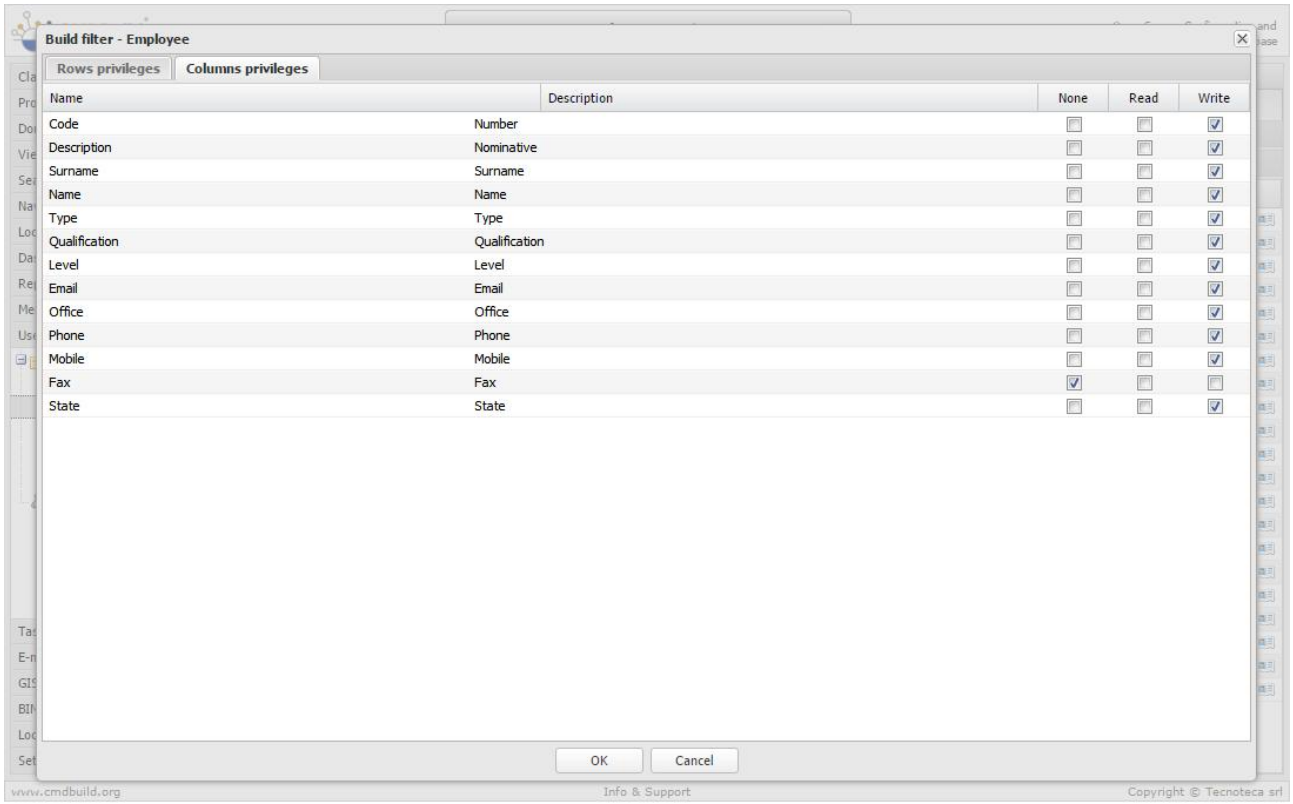
Functions tab



Classes: privileges on columns

Users belonging to the group under configuration are able to select the permissions of the class attributes:

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

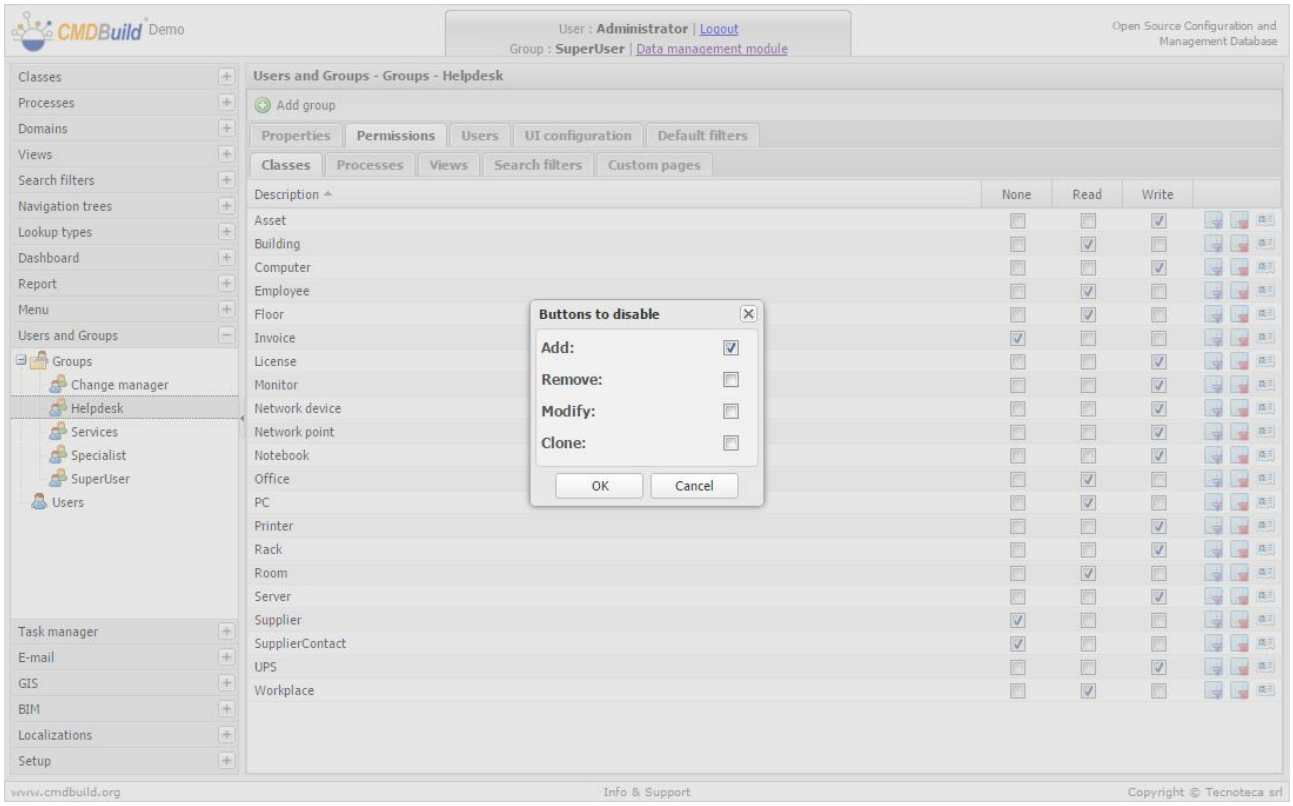


Classes: disabling GUI buttons

On the CMDBuild UI you can disable the permissions related to the insertion / modification / cloning / deleting of a card.

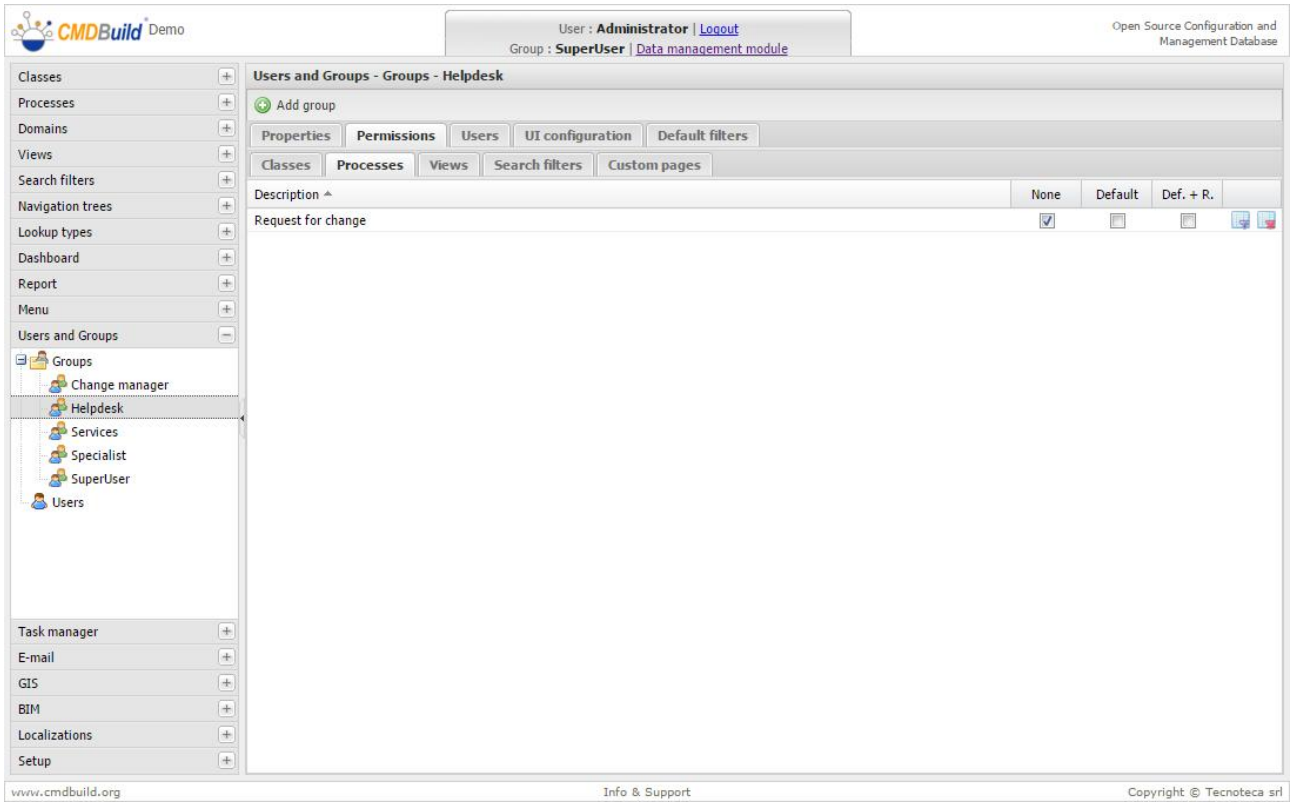
This option is only related to the CMDBuild GUI. It has no consequences outside it (e.g. by quering the permissions of a class through webservice).

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



Processes TAB

Here's the page related to the definition of process permissions, similar to the class one.

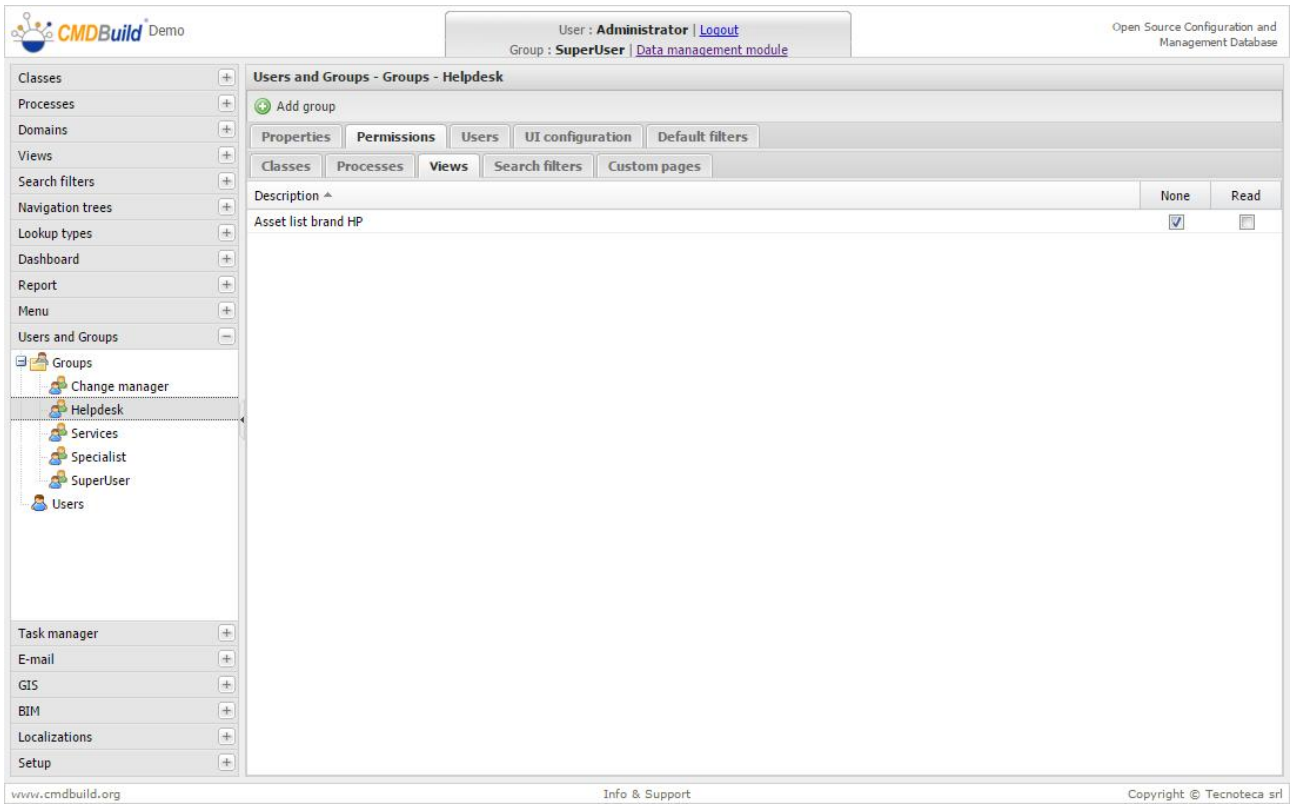


Only the available permissions are different:

- none, if the process cannot be visible
- default, if permissions derived from the XPDL descriptor of the process must prevail (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, if the possibility of viewing the process is enabled in addition to the permissions derived from the XPDL descriptor

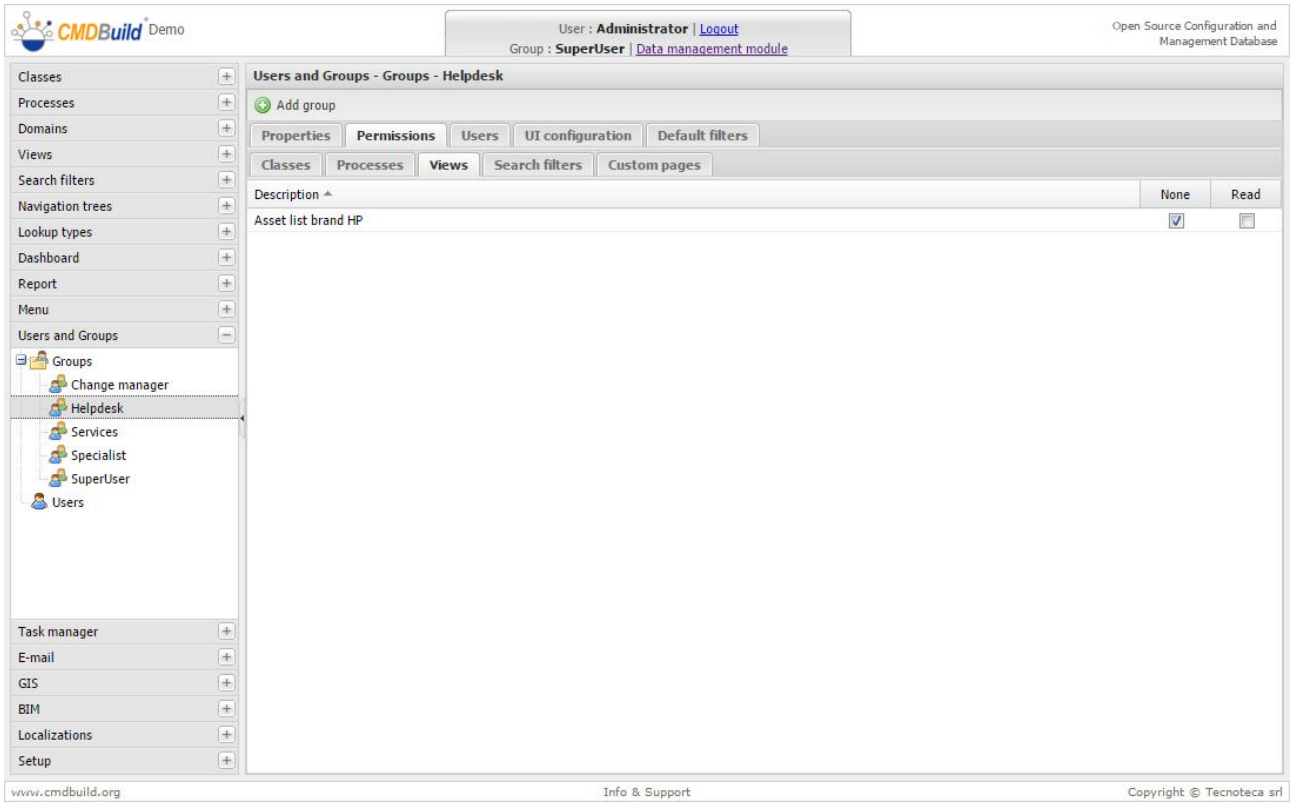
TAB Views

Here's the page related to the definition of views permissions.



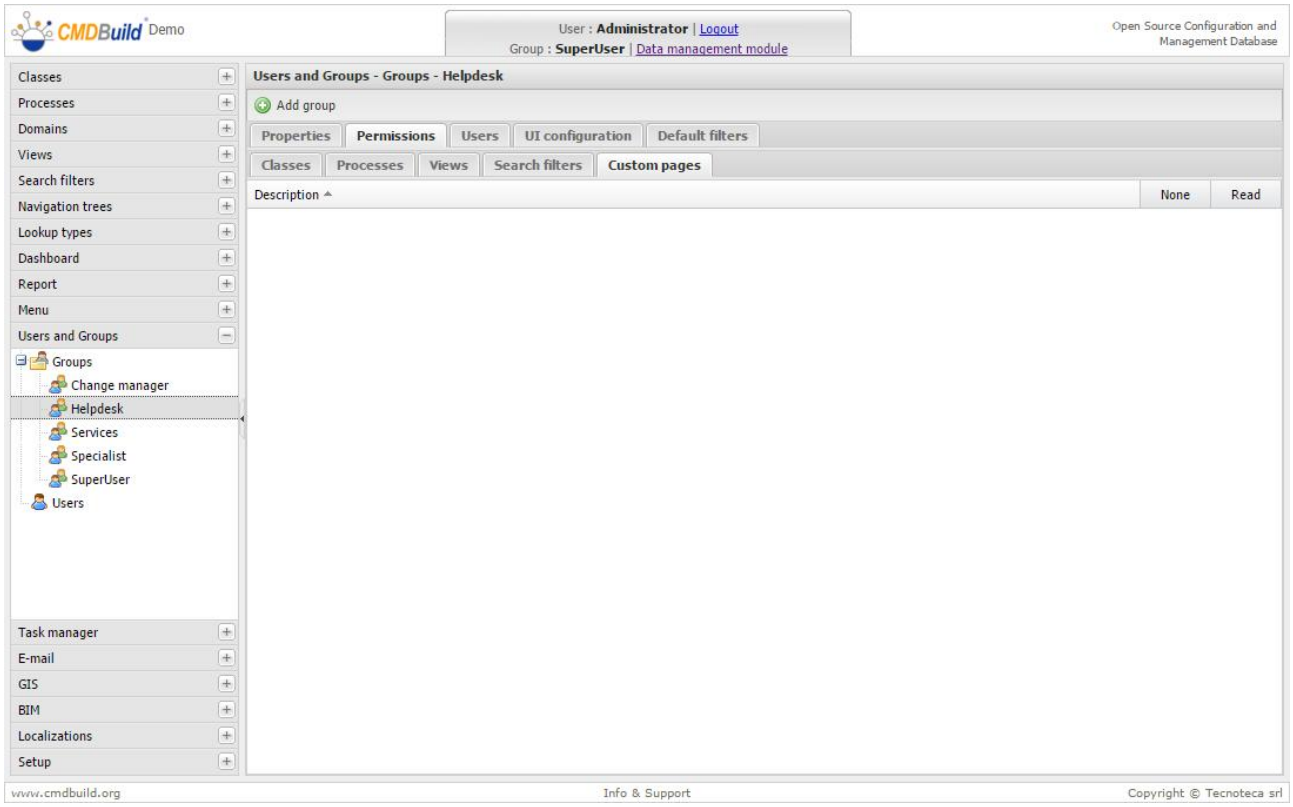
TAB Search filters

Here's the page related to the definition of search filter permissions.



TAB Custom pages

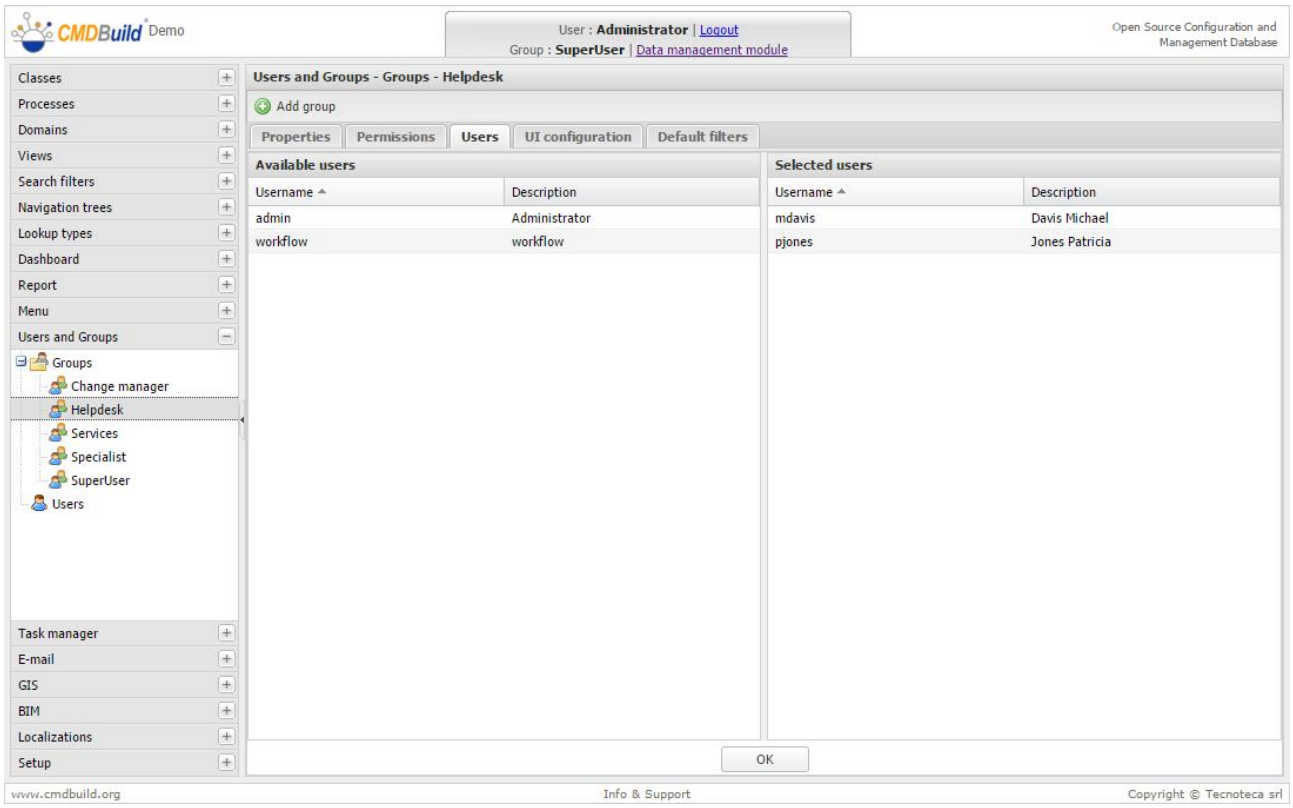
Here's the page related to the definition of permissions for the custom pages.



Users tab

The Users tab allows you to associate a user to one or more groups.

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



TAB UI setup

The TAB "UI setup" allows the personalization of the user interface of a specific users's group, excluding some examples or simplifying their global working.

In particular, you can work on:

- the elements shown in the accordion menu on the left, with an additional detail for the utility functions
- the available TABs for the card management
- the available TABs for the process management
- further options

Accordion menu

With the buttons available in the first box you can disable some components of the main menu of the application (accordion menu on the left) for the current group, in particular:

- Cards accordion
- Processes accordion
- Views accordion
- Dashboards accordion
- Report accordion
- Accordion “Custom pages”
- Utility - change password
- Utility - multiple update
- Utility - import CSV file
- Utility - export CSV file

TAB cards

With the buttons in the second box you can disable some TABs available for the cards management for the current group:

- “Details” TAB
- “Notes” TAB
- “Relations” TAB
- “History” TAB
- TAB “Email”
- “Attachments” TAB

Processes TAB

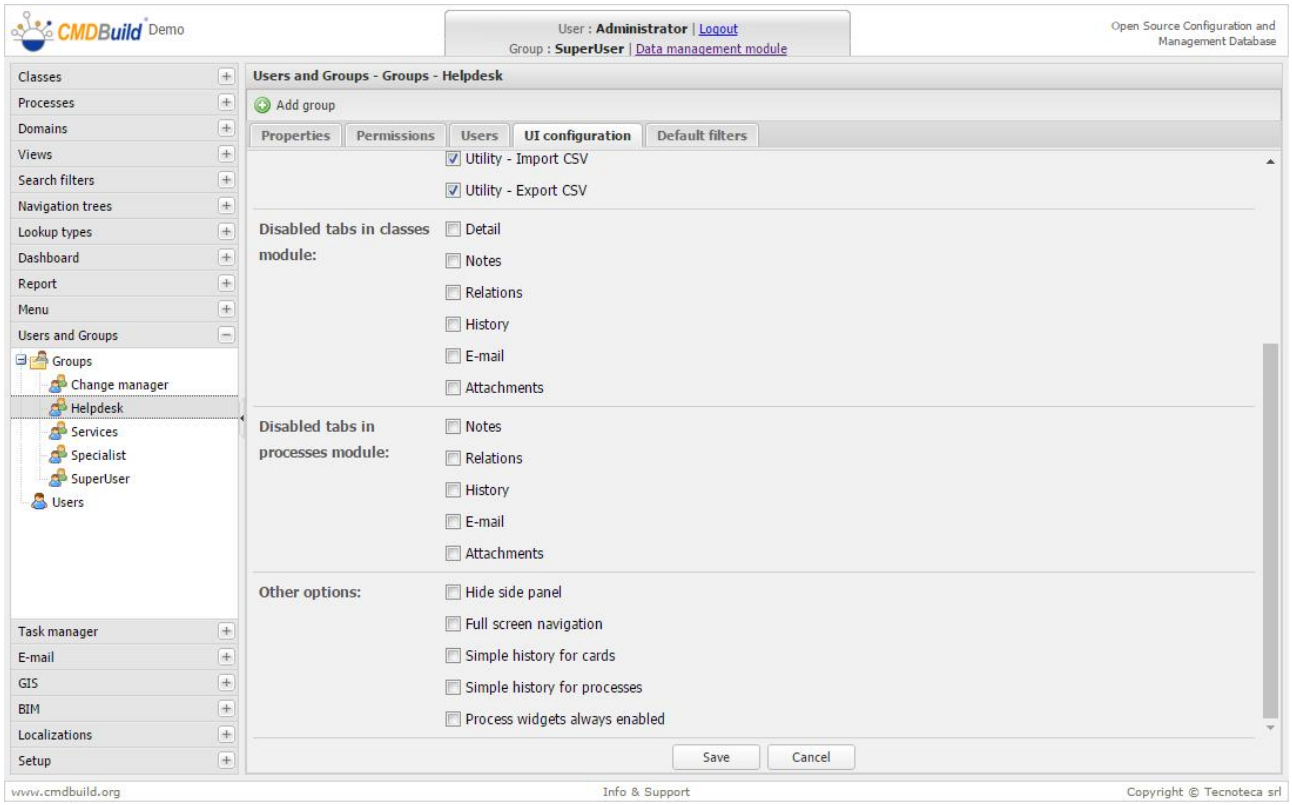
With the buttons in the third box you can disable some TABs available for the process management for the current group:

- “Notes” TAB
- “Relations” TAB
- “History” TAB
- TAB “Email”
- “Attachments” TAB

Further options

With the buttons in the last box you can:

- hide the whole accordion menu in the left side of the screen (which you can restore with the proper graphic control),
- widescreen display the cards management (the entry of a new card and the change of a card selected in the list will temporarily hide the list)
- demand the simplified display of a card history (reducing the summary information shown and blocking the entry of the complete historicized card)
- demand the simplified display of a process history (reducing the summary information shown and blocking the entry of the complete historicized card)
- always keep active the widgets provided for the current state



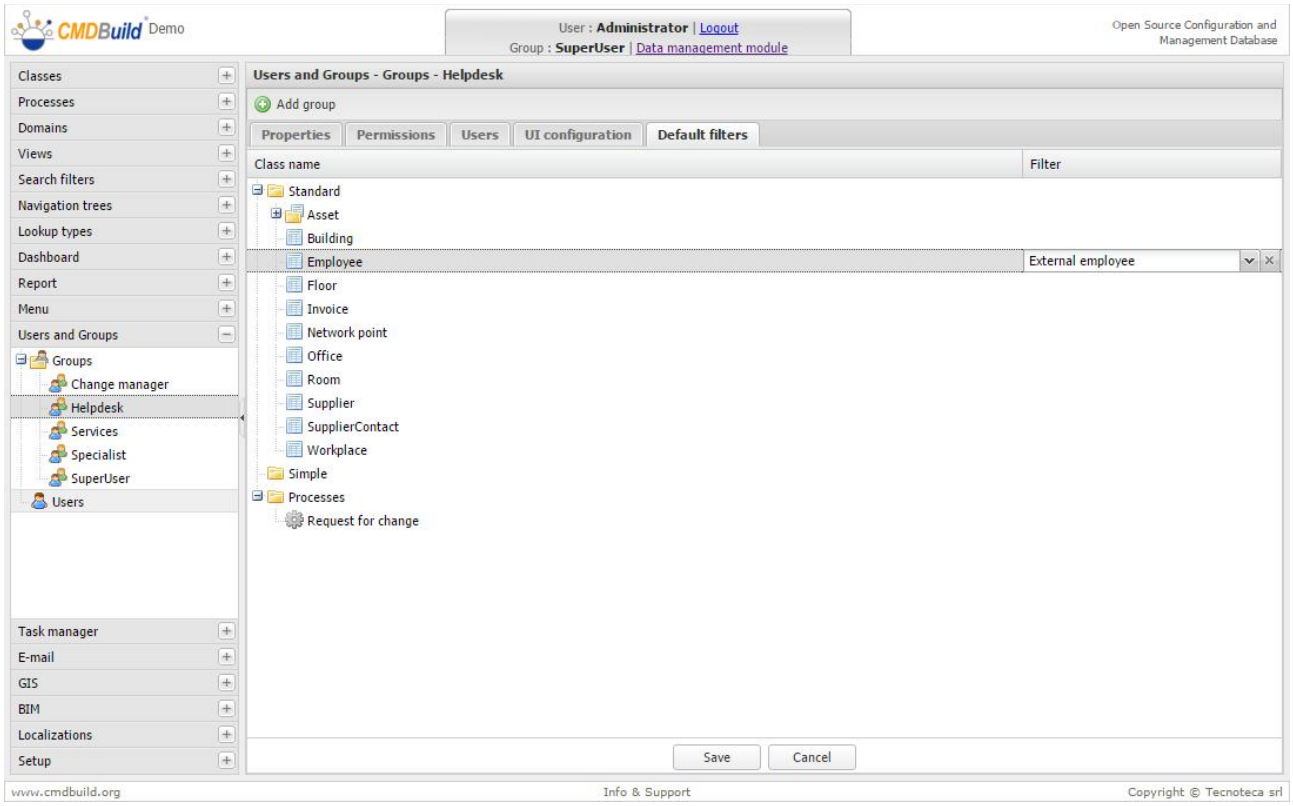
In the previous section related to permissions there is further info about the possibility of disabling the permissions related to the insertion / modification / cloning / deleting of a card on the standard CMDBuild UI.

TAB default filters

The TAB “Defult filters” allows you to specify a possible filter applied into the Module for the data Management, when a user of a specific group opens a class of a process

In particular, you can:

- view the application filters, from this interface or from the menu for the filter management
- add or edit a default filter which was already set

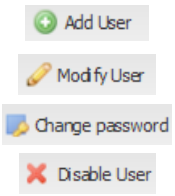


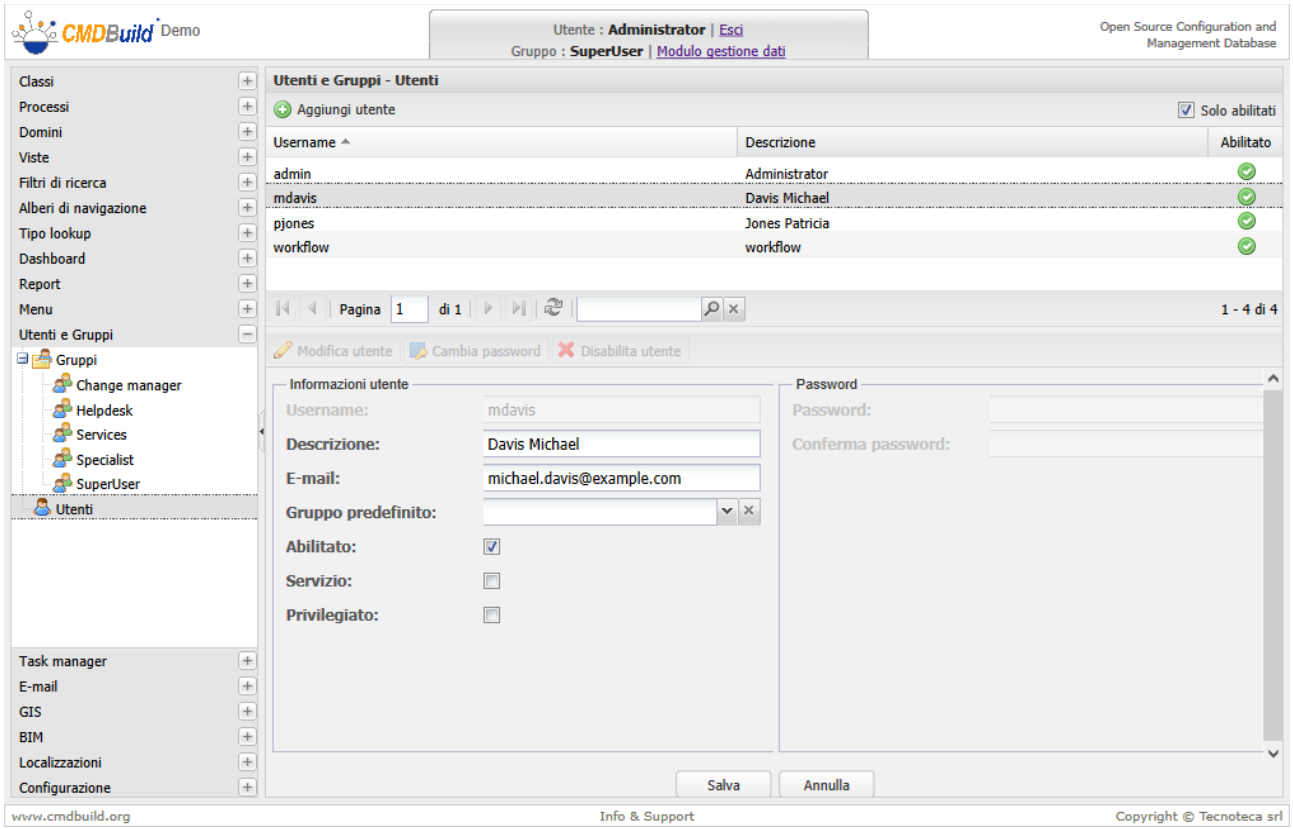
Users

This feature allows you to create and edit system accounts.

There is the possibility of performing the following operations:

- add a new user
- edit properties of an existing user
- change user password
- disable a user





To create a new user you have to insert the following information:

- "Username" - login username
- "Description"
- “Email”
- "Default group": if you select the empty option, the user will be prompted to choose a group (among the ones he belongs to) at login time, hence the user will inherit specific group permissions ; otherwise the user permissions will be the sum of all groups the user belongs to . The default group, on the other hand, is used to identify the group that starts the processes (in case of multiple starts) and set navigation menu (if defined).
- "Active", indicates whether the user is active or has been deleted (logical deletion)

- “Service”, it shows that it is a faked user used by an automatic service
- “Privileged”, it shows that it is a service user, with additional privileges
- “Password” - login password
- “Confirmation” - confirm login password

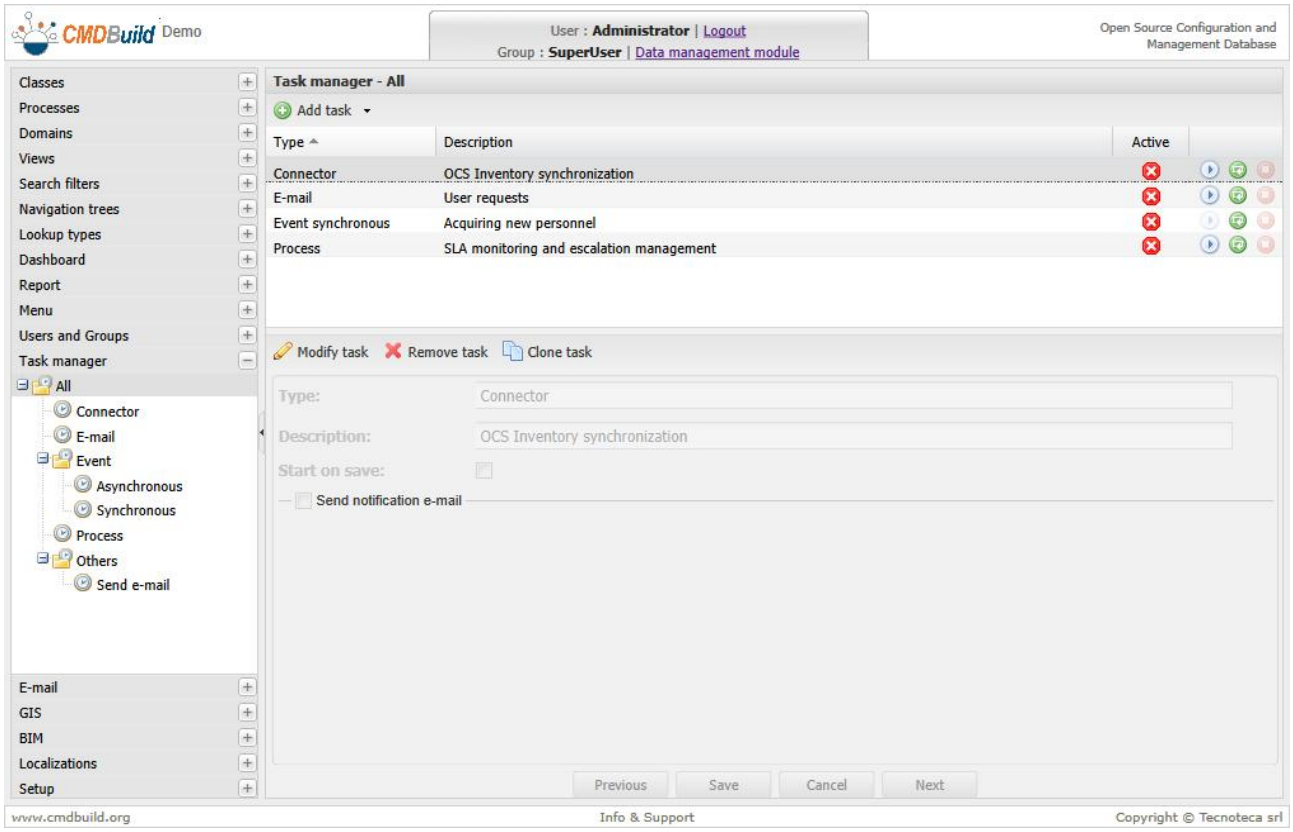
Task Manager

Basic functions

The Task Manager gathers all CMDBuild functions carried out in the background. In this way you can have a general view on the active tasks and the related parameters.

At the moment, the kinds of tasks taken into consideration are as follows:

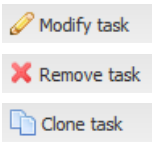
- management of incoming e-mails
- management of asynchronous events
- management of synchronous events
- workflow schedule
- Wizard Connector schedule
- scheduled sending of emails with attached parametric report



There is the possibility of performing the following operations:

- starting a non active task 
- running a single task 
- suspend a started task
- configure a new task according to the provided typologies 

- modify the configuration of an existing task
- remove an existing task
- clone an existing task



Before the modification of a started task, we suggest you to suspend it and then restart it.

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

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

Inbox management

For this specific task you have to wait for an e-mail incoming into a specific folder of a specific account, verify if it coincides with any prearranged filter criteria and carry out the recommended instructions.

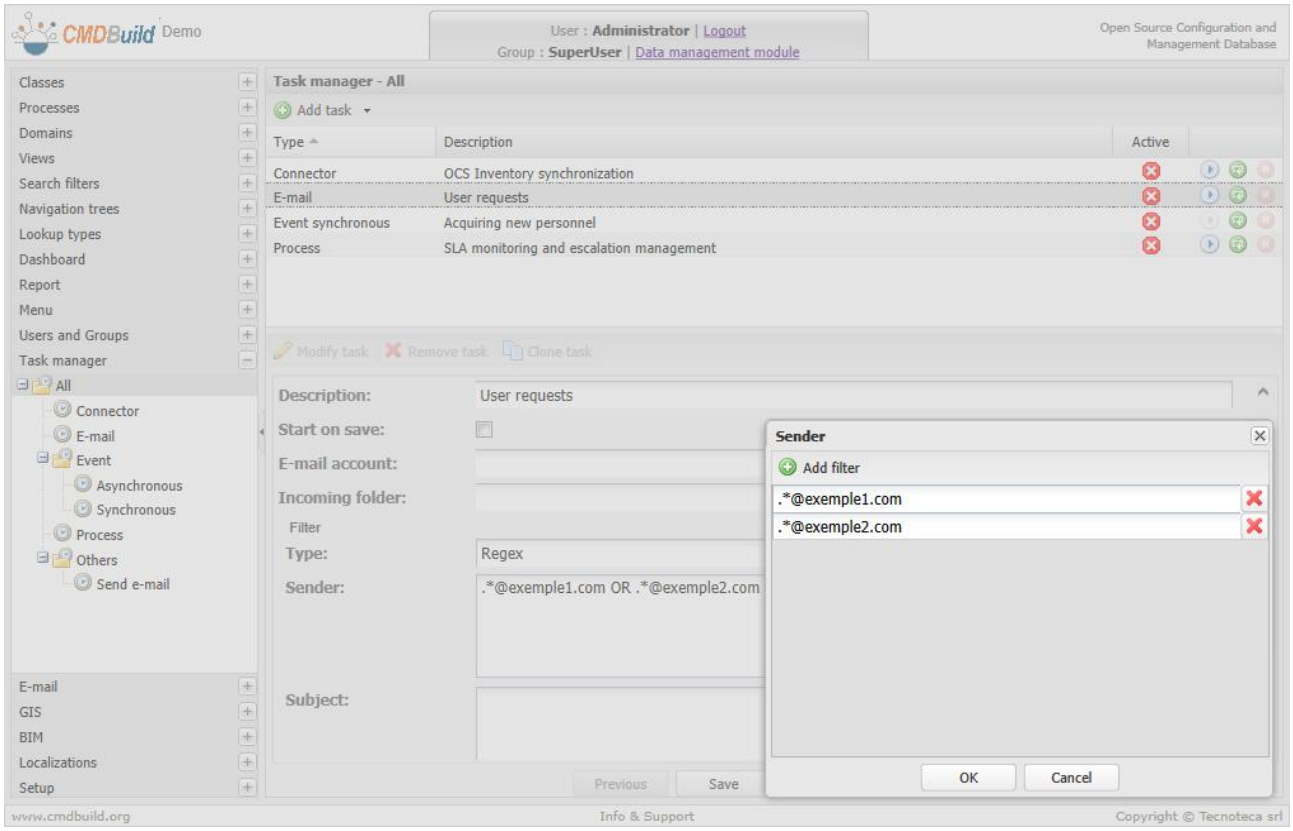
Such activities include also:

- 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.

Page 1 of the wizard

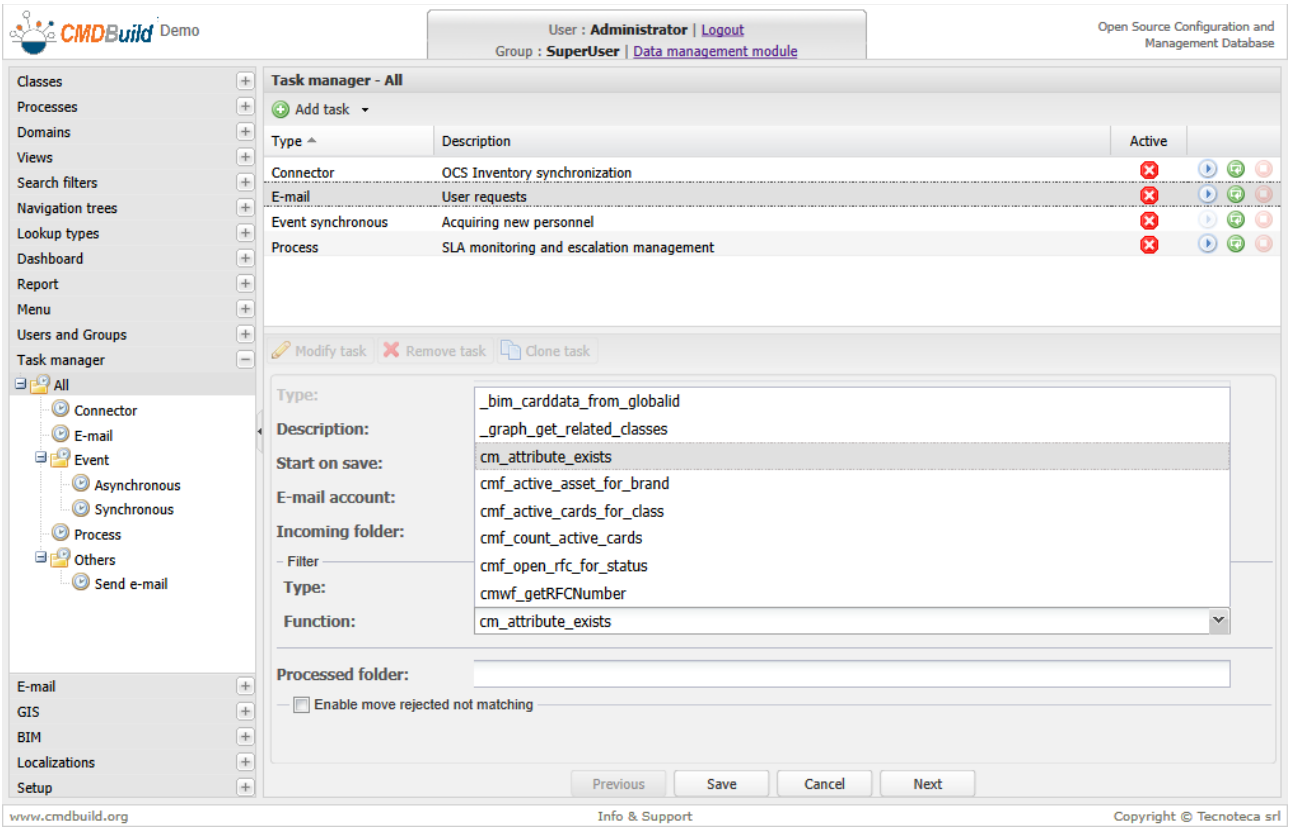


The following information is required:

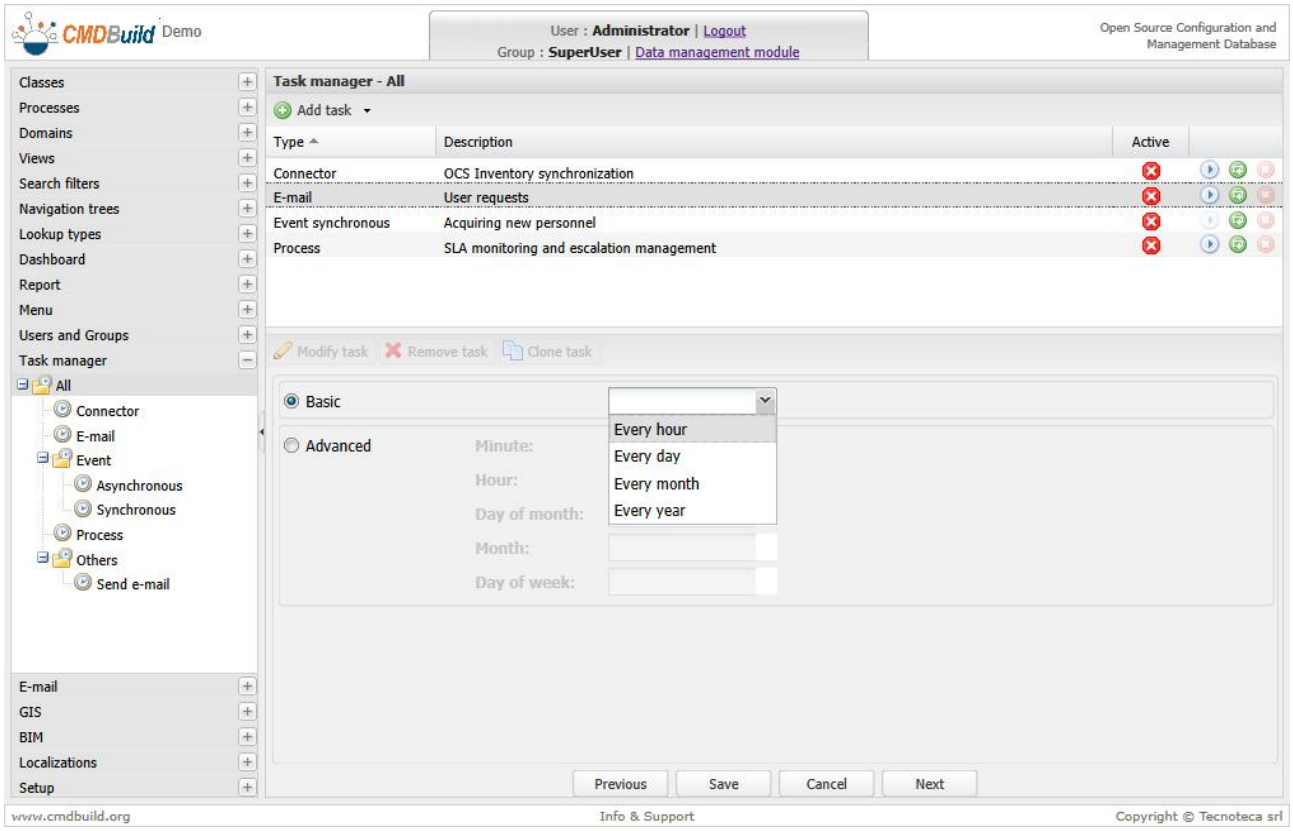
- "Description"
- “Start on save”: the task starts when the configuration is saved
- “E-Mail account”: post account to control the incoming e-mails (please see the related section of this manual)

- Kind of filter: None, Regex (filter with regular expression), PostgreSQL Function
- “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 (in the screenshot over the filter, it will take into consideration just those e-mails containing the string “example1.com” or “example2.com”)
- “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

Example of screenshot with PostgreSQL Function filter:



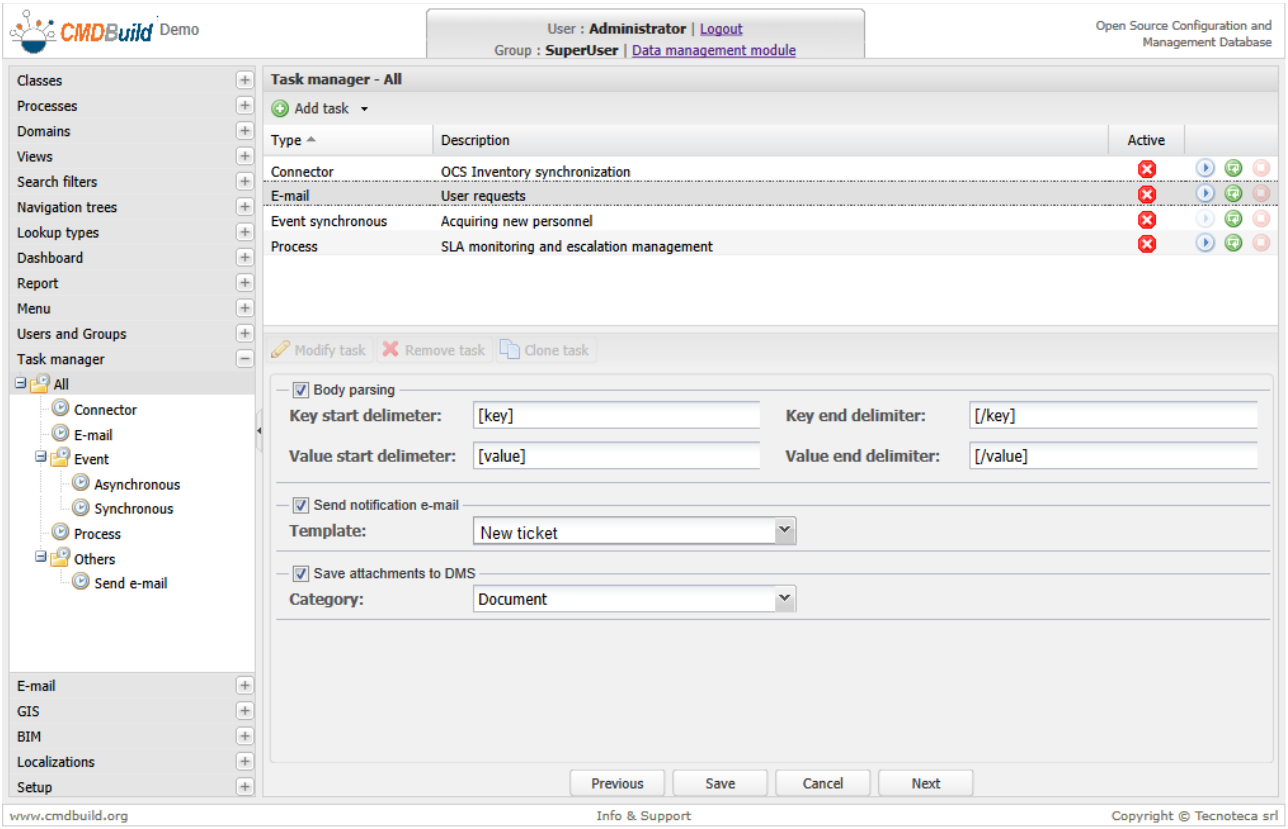
Page 2 of the wizard



The information related to the scheduling ways of the control task of the incoming e-mail 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

Page 3 of the wizard



The following information is required in groups:

- “Body parsing”
  - “Key start delimeter”: it allows to specify the initial delimeter of the possible key field available in the e-mail
  - “Key end delimeter”: it allows to specify the final delimeter of the possible key field available in the e-mail
  - “Value start delimeter”: it allows to specify the initial delimiter value field available in the e-mail
  - “Value end delimeter”: it allows to specify the final delimiter of the value filed available in the e-mail
- “Send notification email”
  - “Email template”: it requires to select one of the templates preconfigured with the proper function (e-mail management menu)
- “Save attachment to Alfresco”
  - “Category”: it requires to select the category that should be associated with the attachment (that will be saved in the DMS, associated with the e-mail card)

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:

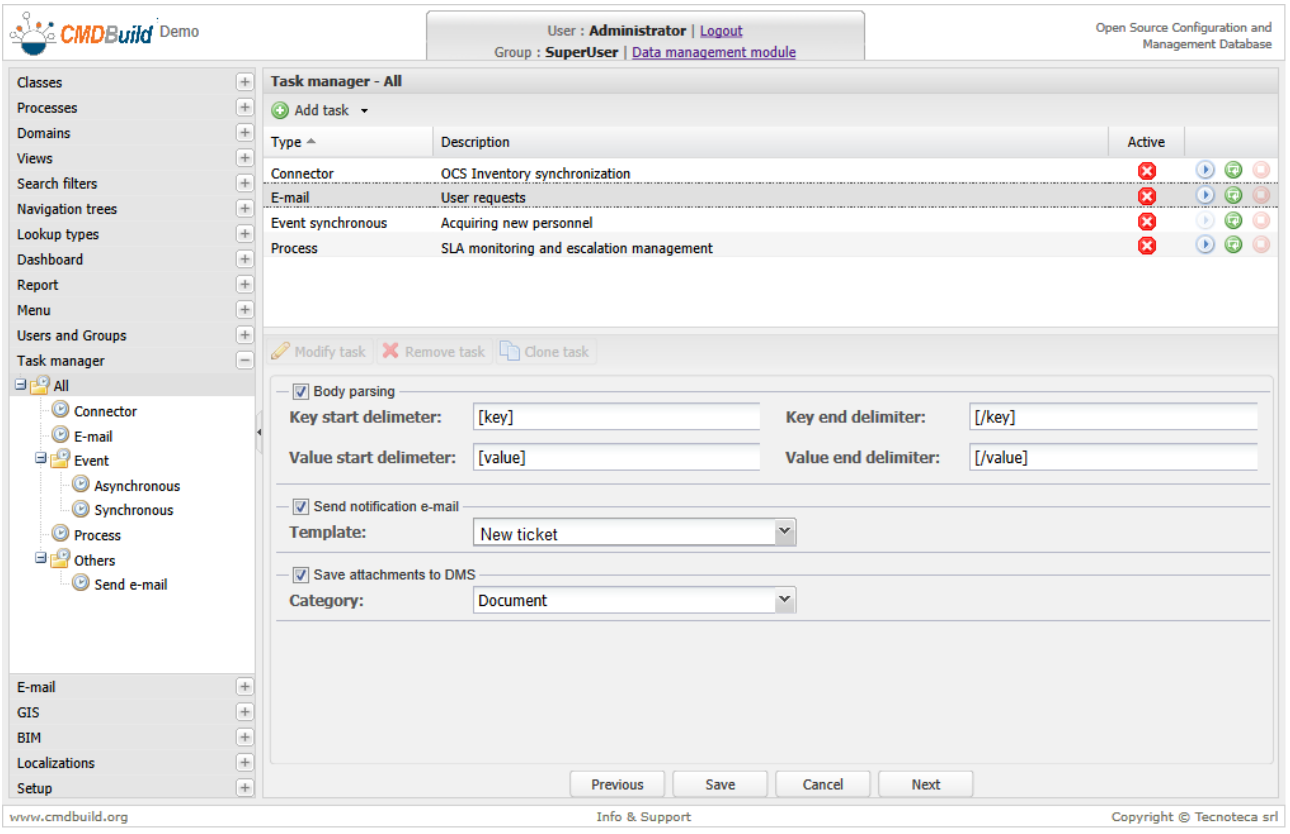
```
Email body

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

Page 4 of the wizard

It includes the test useful to require the instant task starting and to specify then:

- process name
- mapping of the mail fields with the process attributes



Synchronous events

For this specific task you have to check certain conditions on the CMDB data by applying a predefined filter and carry out the recommended instructions.

Such activities include also:

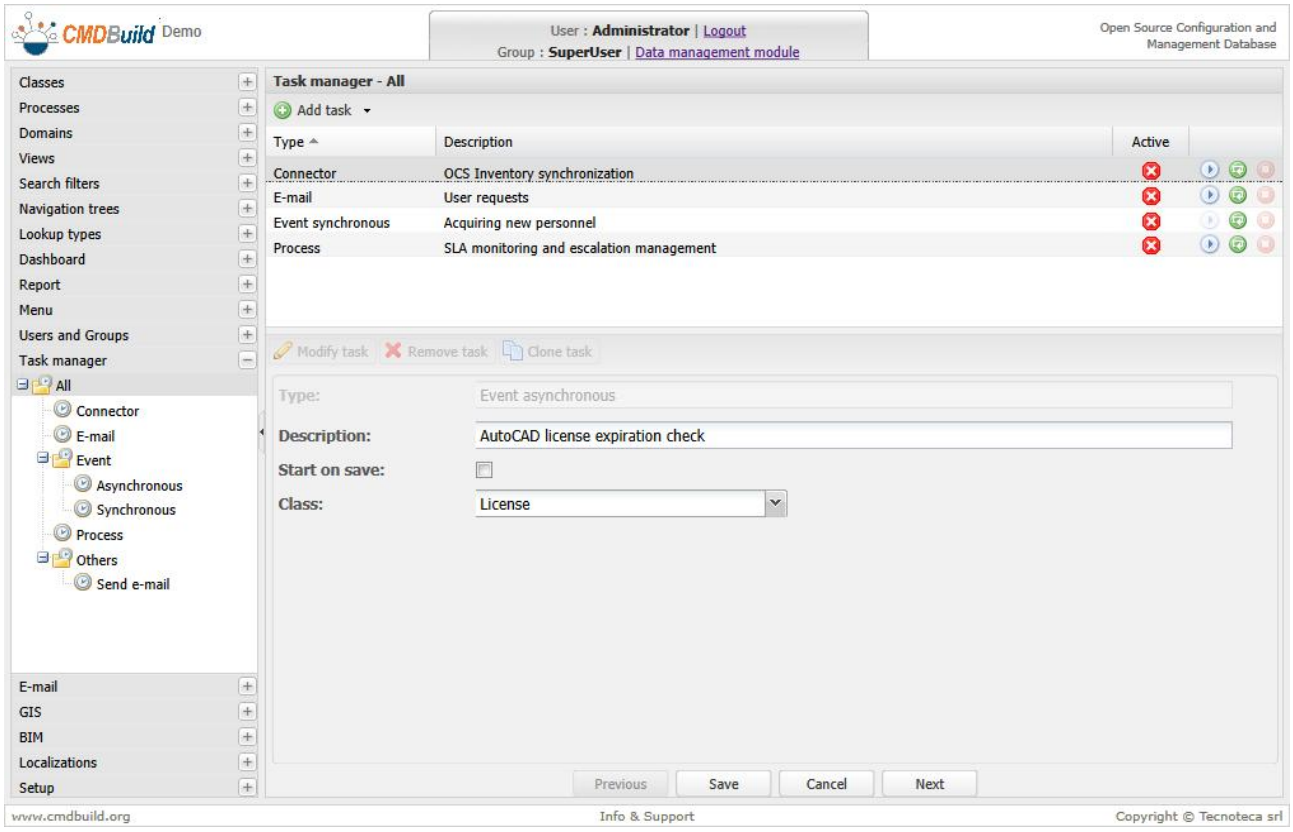
- sendinf notifications with attached report
- starting of workflows
- script execution

This function can be used to configure the e-mail starting if the warranty of an asset is going to expire or to start a solicit workflow if a ticket is exceeding the period permitted for its resolution, etc.

At the moment you can configure events on standard classes, not on workflows.

The wizard includes five pages described below.

Page 1 of the wizard

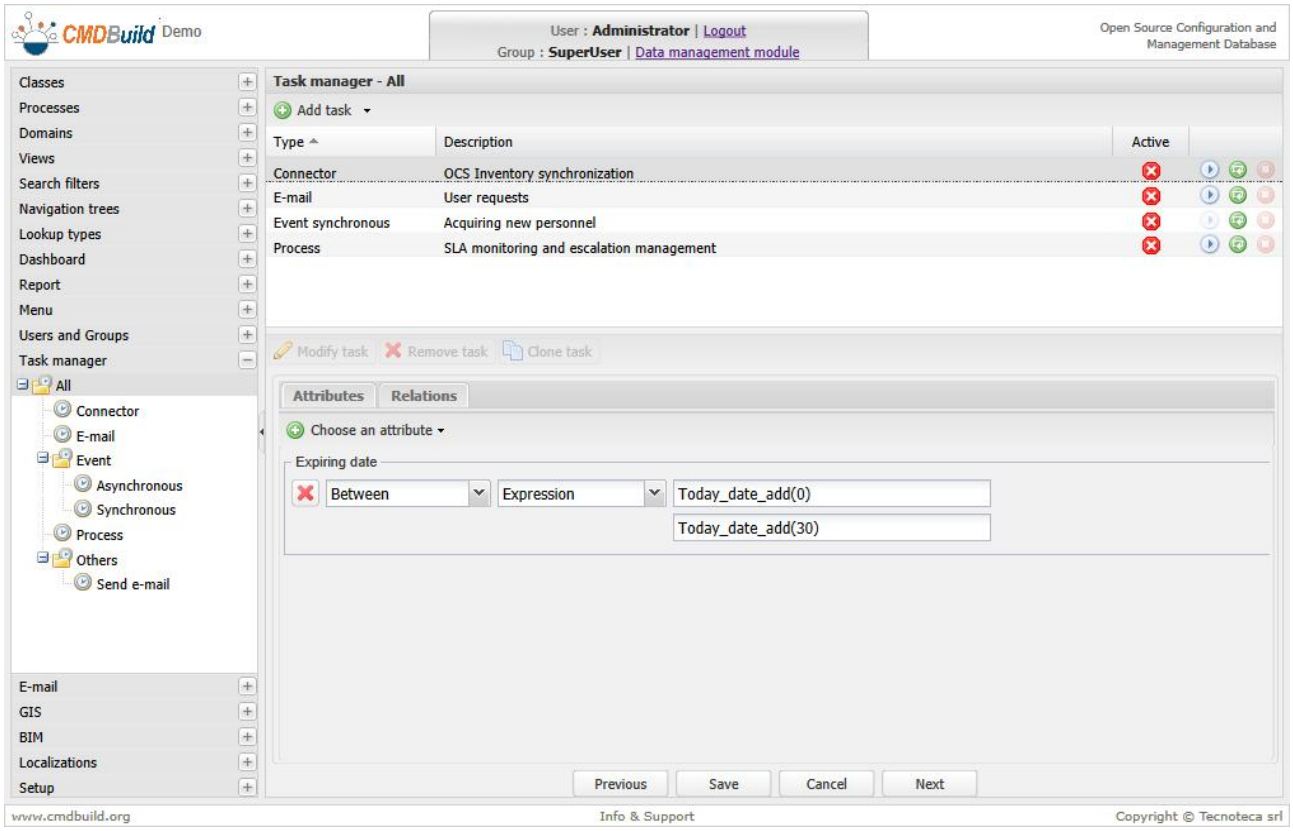


The following information is required:

- "Description"
- “Start on save”: the task starts when the configuration is saved
- “Target class”: in this class you have to control its data in order to evaluate if the following conditions are verified

Page 2 of the wizard

The second page of the wizard allows you to define a control filter with the same mechanisms available for the configuration of the CMDBuild standard filters.



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

Page 3 of the wizard

The information related to the task schedule mode is required. The task will periodically control the data to evaluate if the filter conditions are verified:

- “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

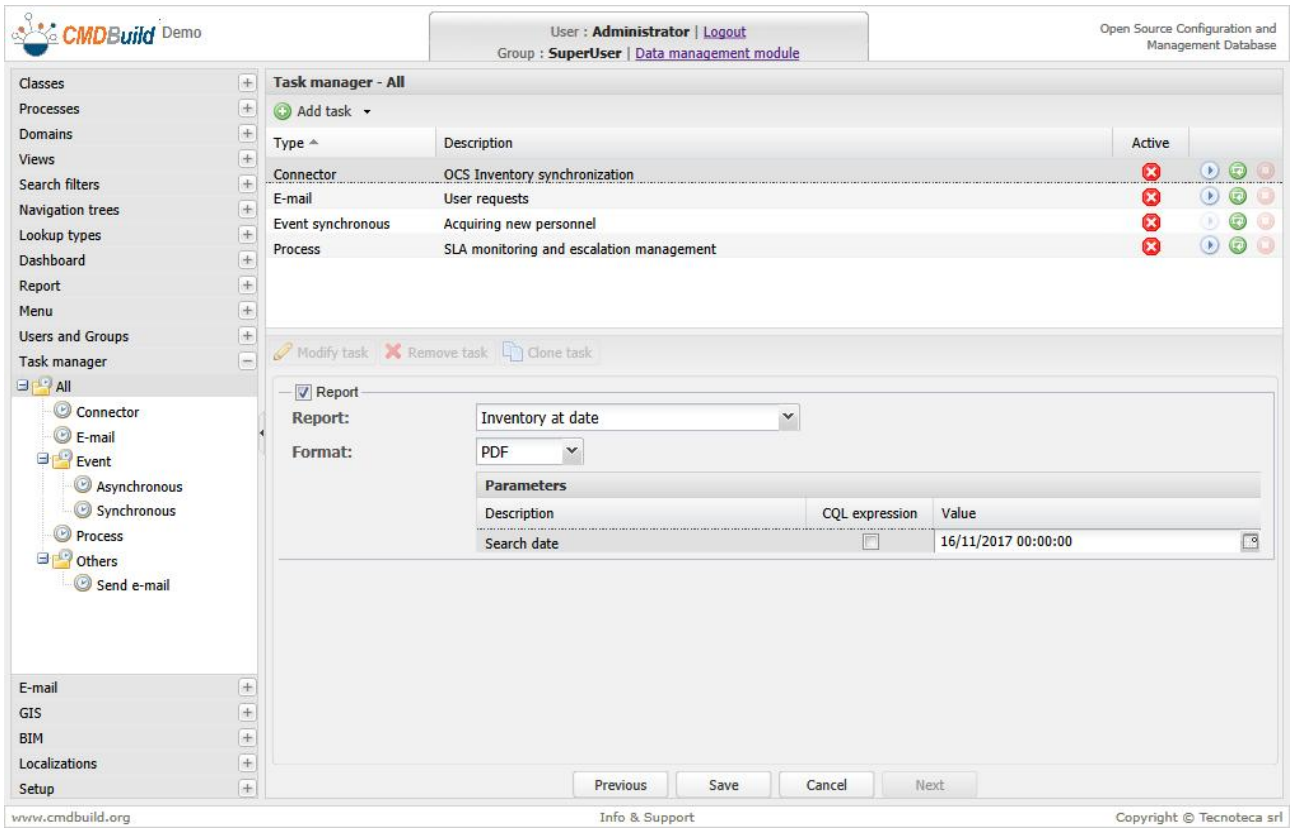
Page 4 of the wizard

It includes only the test useful to require the notification e-mail sending by specifying the account and template.

Page 5 of the wizard

It allows to:

- choose the report to enclose to the email
- choose the format
- in case the report has certain launch parameters, they should be optimized with constants or with a CQL expression (by summing for example a number of days)



Asynchronous events

For this specific task you have to intercept the CMDB operations and carry out the recommended instructions.

Such activities include also:

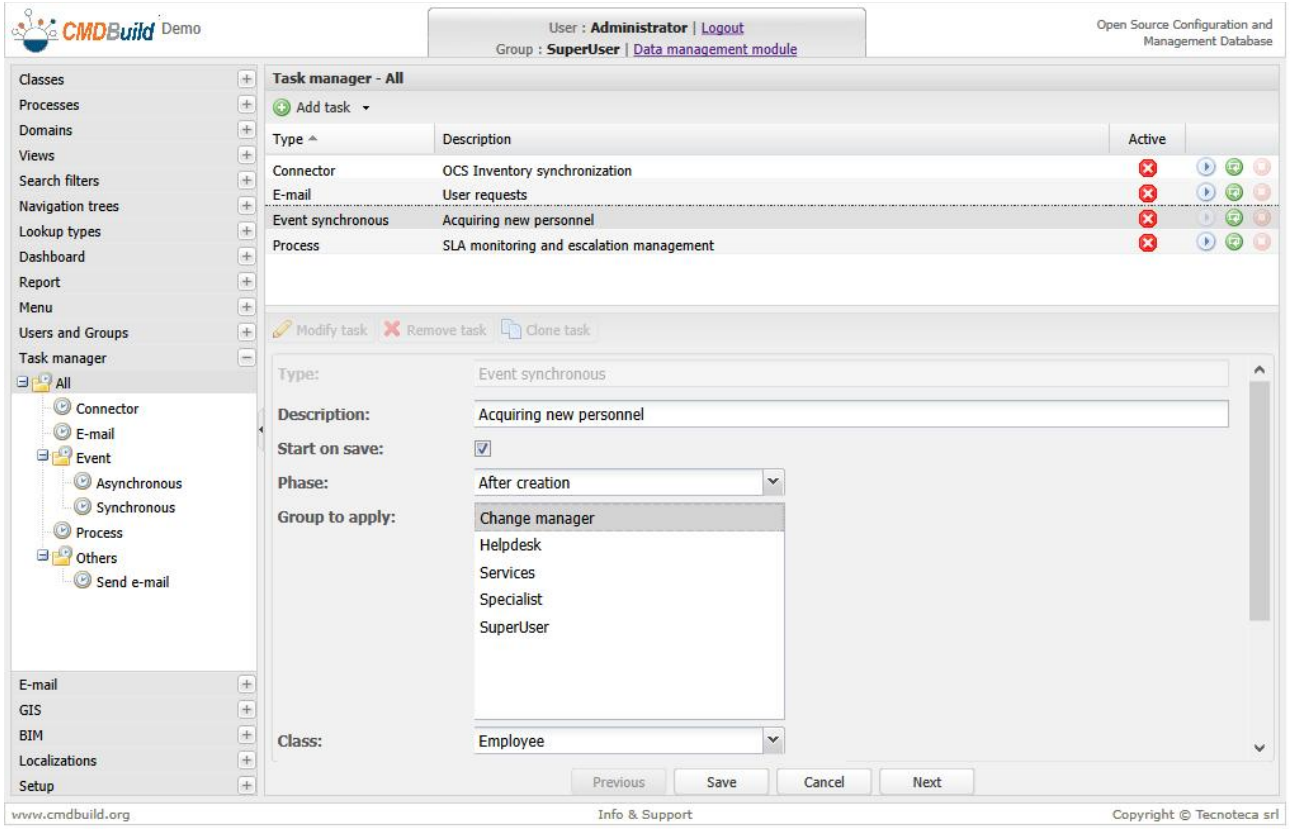
- sending notifications
- starting workflows
- script execution

This function can be used to configure the e-mail starting in case that a new employee was inserted in CMDBuild, to start a deprovisioning workflow (removing the account, recovering the job role, transfering the tasks to a collegues, etc.) if an employee resigns, etc.

At the moment you can configure events on standard classes, not on workflows.

The wizard includes three pages, described below.

Page 1 of the wizard



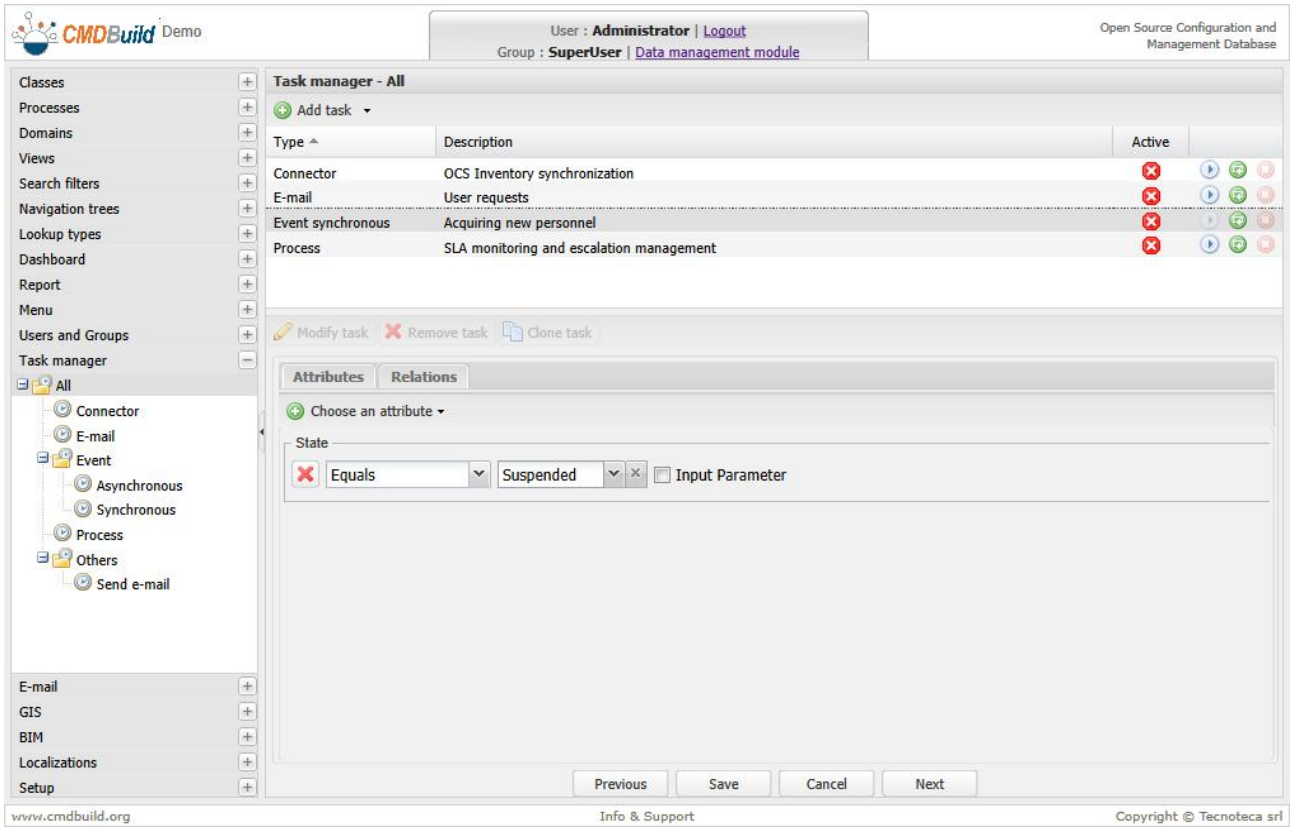
The following information is required:

- "Description"
- “Start on save”: the task starts when the configuration is saved
- “Step”: it shows the data for the control (as soon as a card is inserted, before a card is

modified, as soon as a card is modified, before a card is deleted)

- “Group”: to limit the control to the modifications of certain predefined groups
- “Class”: to control the carried out operations

Page 2 of the wizard



The second page of the wizard allows you to define a control filter with the same mechanisms available for the configuration of the CMDBuild standard filters.

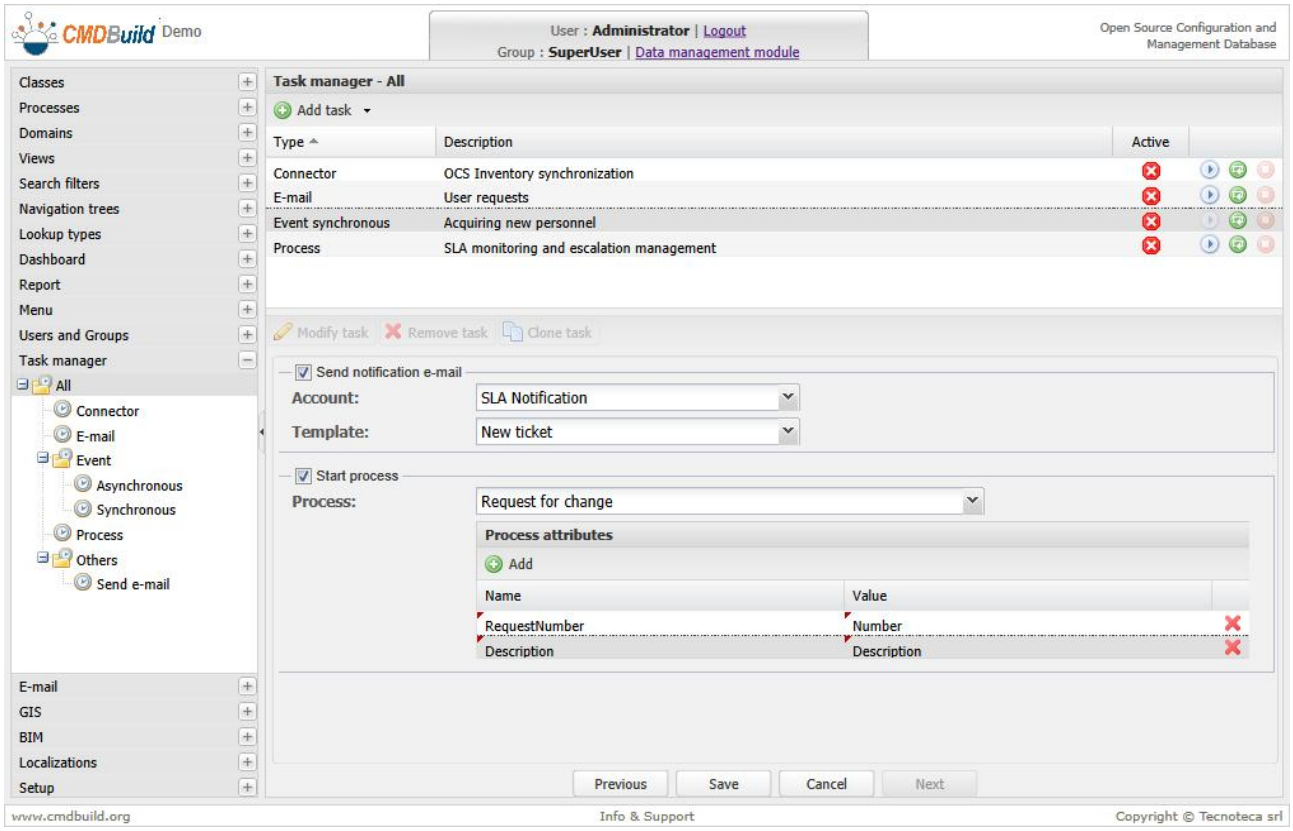
For a complete description of the configuration mode, please see the related section of this manual.

Page 3 of the wizard

It includes the test useful to require the notification e-mail sending by specifying the account and template.

It allows also the automatic starting of a Change management workflow, specifying

- “Process”: starting workflow
- “List of attributes conformity”, to initialize one or more starting process attributes

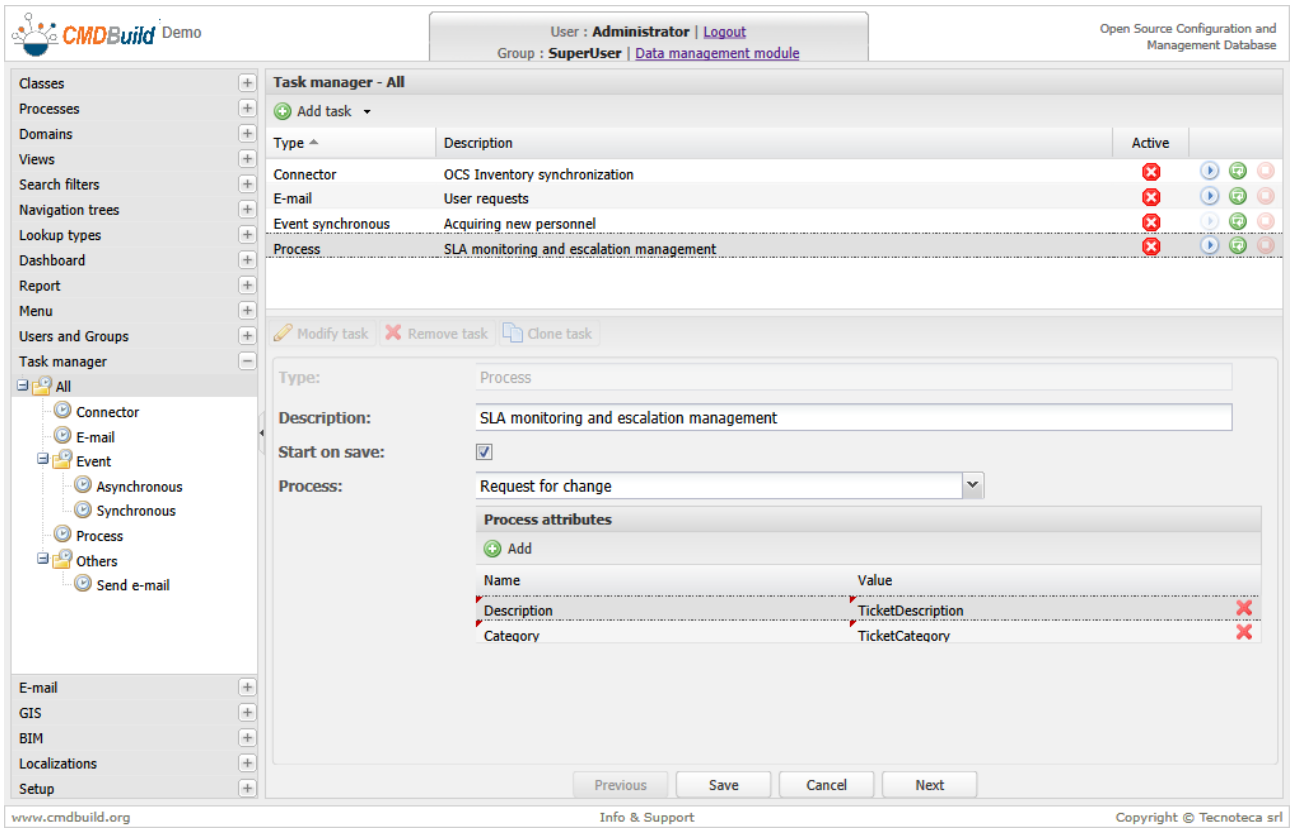

CMDBuild Demo
User : Administrator | Logout
Group : SuperUser | Data management module
Open Source Configuration and Management Database
Classes
Processes
Domains
Views
Search filters
Navigation trees
Lookup types
Dashboard
Report
Menu
Users and Groups
Task manager
All
Connector
E-mail
Event
Asynchronous
Synchronous
Process
Others
Send e-mail
E-mail
GIS
BIM
Localizations
Setup
Task manager - All
Add task
Type
Description
Active
Connector
OCS Inventory synchronization
E-mail
User requests
Event synchronous
Acquiring new personnel
Process
SLA monitoring and escalation management
Modify task
Remove task
Clone task
Send notification e-mail
Account:
SLA Notification
Template:
New ticket
Start process
Process:
Request for change
Process attributes
Add
Name
Value
RequestNumber
Number
Description
Description
Previous
Save
Cancel
Next
www.cmdbuild.org
Info & Support
Copyright © Tecnoteca srl

Workflow

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

The wizard includes two pages described below.

Page 1 of the wizard



The following information is required:

- “Description” of the new task
- “Start on save”: the task starts when the configuration is saved
- “Process”
- “List of attributes conformity”, to initialize one or more starting process attributes

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

Wizard Connector

This function allows you to schedule a data synhronizer connector between external sources and CMDBuild.

The management of IT services by large and medium-sized institutions and companies is necessarily carried out through more specialized information systems that must be able to cooperate in the management of their activities and information.

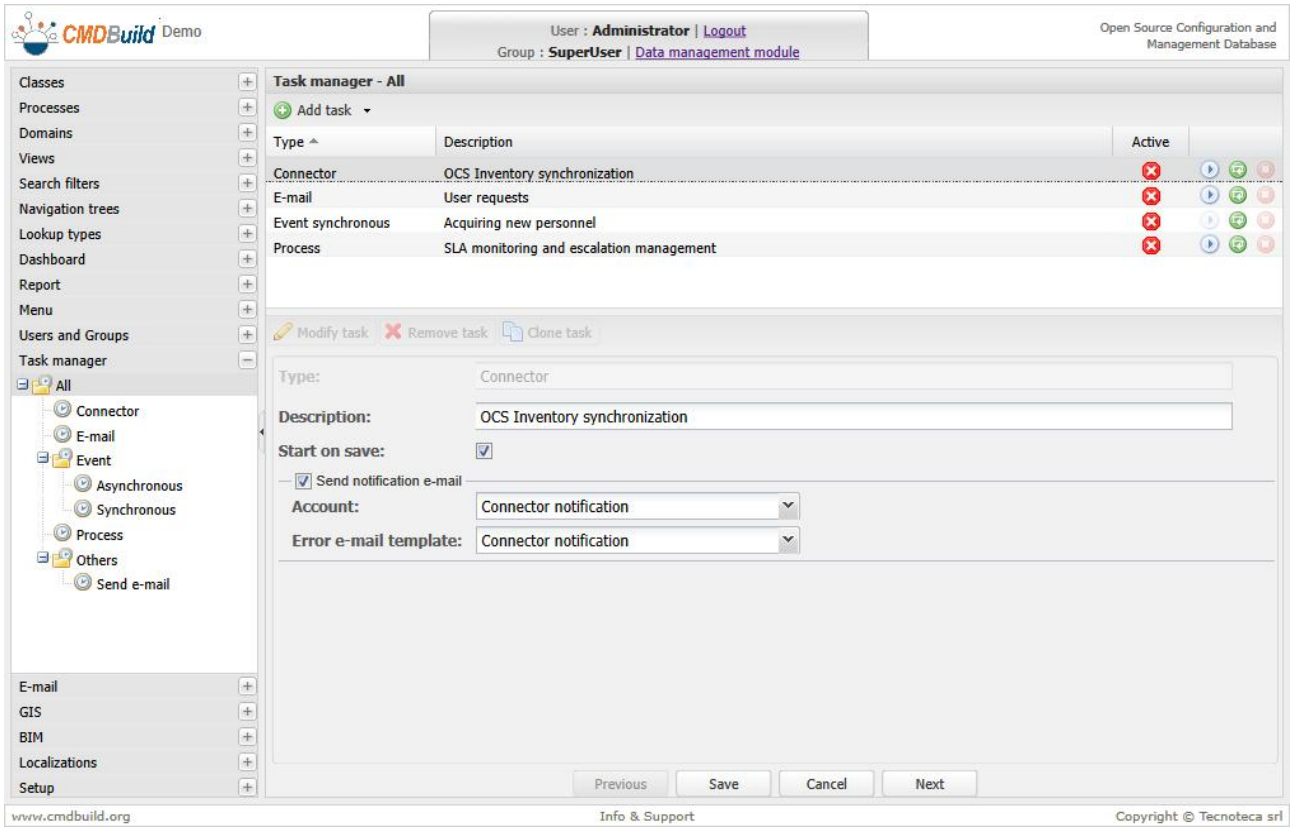
Gathering and checking manually the information managed in the CMDB can cause delay issues or create inaccuracies when updating data. Therefore it is better to update it automatically.

Therefore the configuration of connectors through external systems becomes important in order to sync in CMDBuild (the central CMDB system) data that are mainly managed on other specialist applications, (inventory, monitoring systems, LDAP repository, ERP and HR applications, etc).

The Wizard Connector allows you to resolve the simplest cases in which the "mapping" rules do not require a specific application logic. For more complex needs you can use the Basic Connector and the Advanced Connector.

We are describing below the configuration forms of the Wizard Connector. Please see the CMDBuild Connectors Manual for a more detailed comparison of the three solutions.

Page 1 of the wizard



The following information is required:

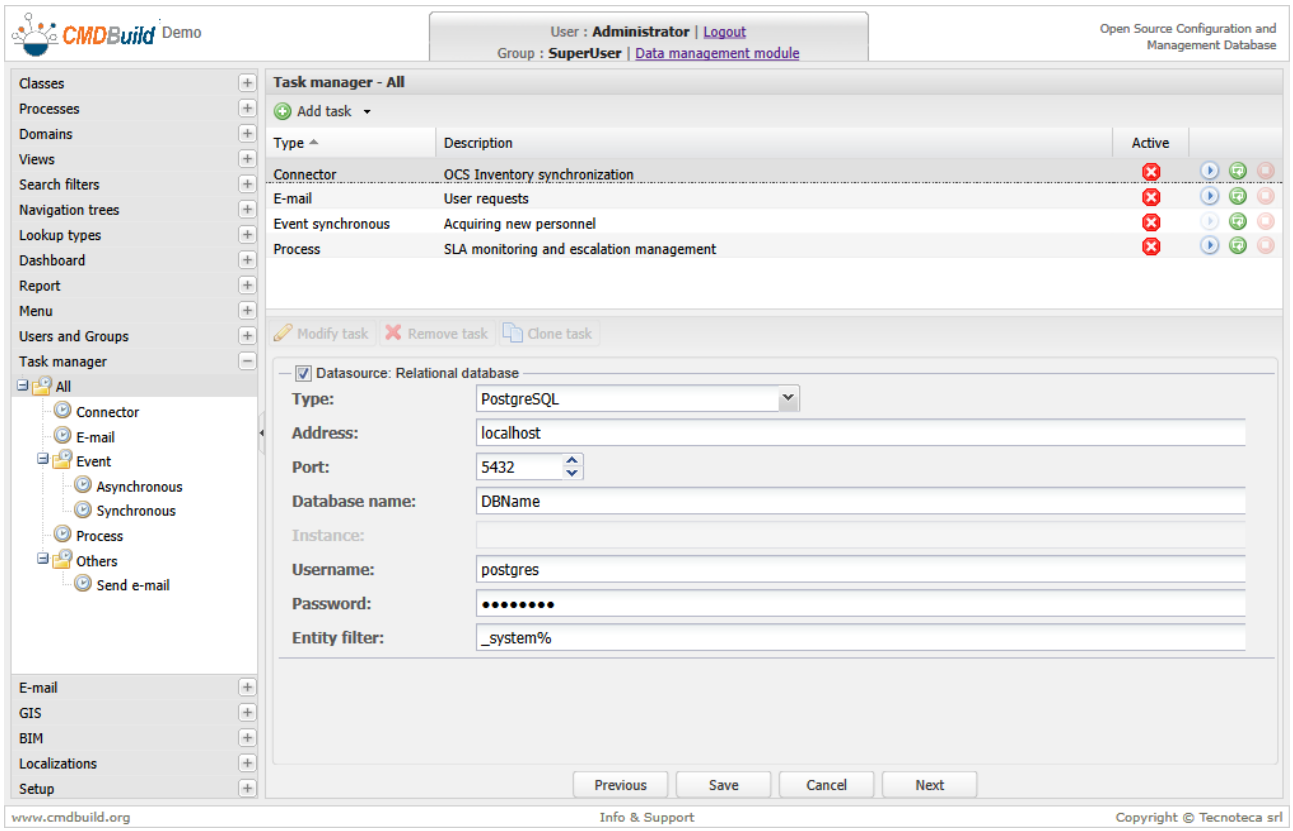
- “Description” of the new task
- “Start on save”: the task starts when the configuration is saved
- “Send an e-mail notification”, specifying if necessary:
  - “account”
  - “template” of the e-mail

Page 2 of the wizard

The information related to the task schedule mode is required. The task will periodically control the data to evaluate if the filter conditions are verified:

- “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

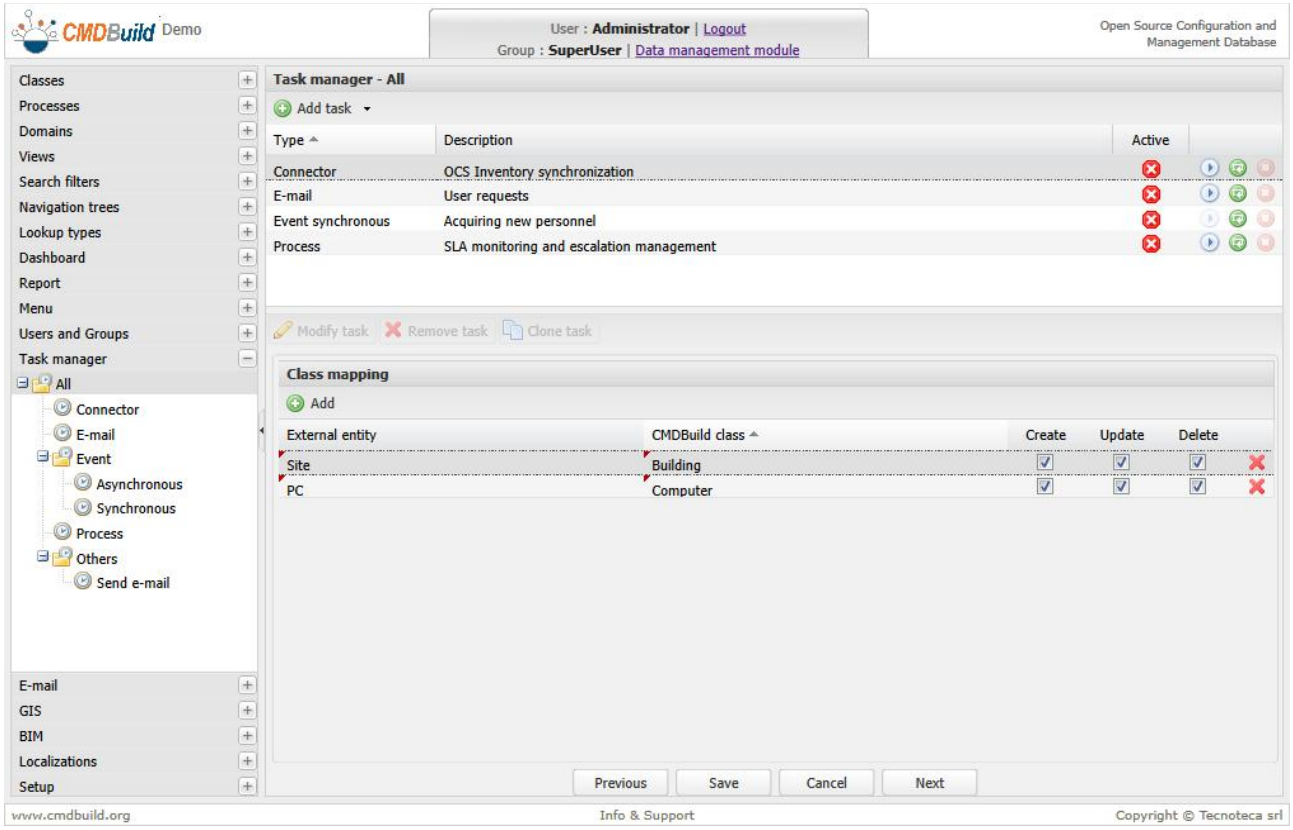
Page 3 of the wizard



The following information related to the data source is required:

- “Type” of the origin database (databases PostgreSQL, MySQL, Oracle and SQLServer are supported, having installed the proper access drivers)
- “Address” of the source database
- “Port” of the source database
- “Name” of the source database
- “Username” for the access
- “Password” for the access
- “Entity filter”, you can select the entities you want to show in the mapping criteria definition (e.g. some views with the same prefix, prearranged for connector query)

Page 4 of the wizard

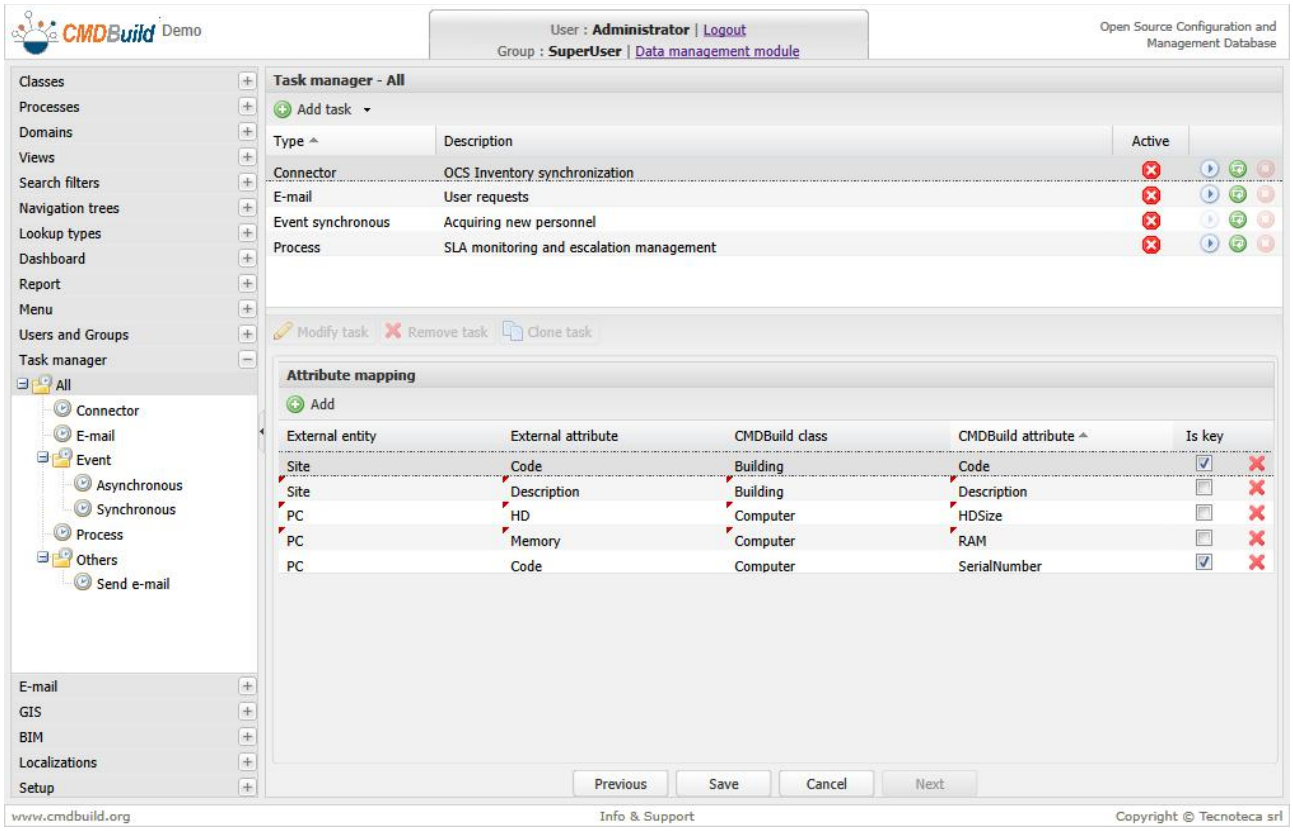


For each entity of the external data source involved in the synchronization activities, the following info is required:

- “External entity”: name of the table, view or other entity
- “CMDBuild class”: CMDBuild class for the data synchronization
- “Creation”: if enabled, it creates in CMDBuild any data card available in the external system and not in CMDBuild
- “Modification”: if enabled, it updates in CMDBuild any data cards available in the external system with information different from CMDBuild

- “Cancellation”: if enabled, it deletes any data cards not available in the external system and available in CMDBuild

Page 5 of the wizard



For every attribute of each entity to synchronize, the following info is required:

- “External entity”: name of the table, view or other entity
- “External attribute”: the name of the column / information of the external entity
- “CMDBuild class”: CMDBuild class to sync
- “CMDBuild attribute”: CMDBuild attribute to sync
- “Key”: it states the key attribute of each entity (necessary to compare data and decide the kind of synchronization)

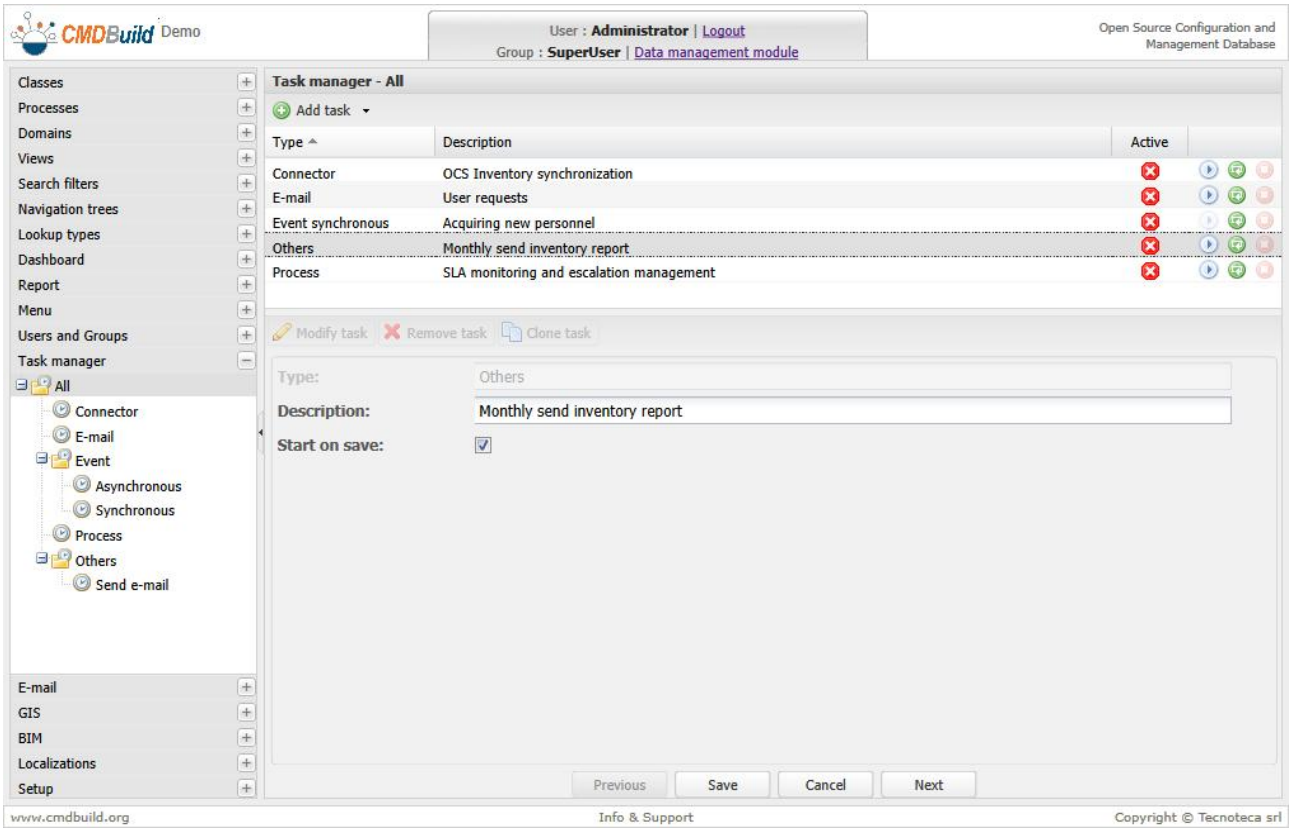
Send email with attached report

This function allows you to schedule the automatic sending of an email with an attached 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 five pages described below.

Page 1 of the wizard



The following information is required:

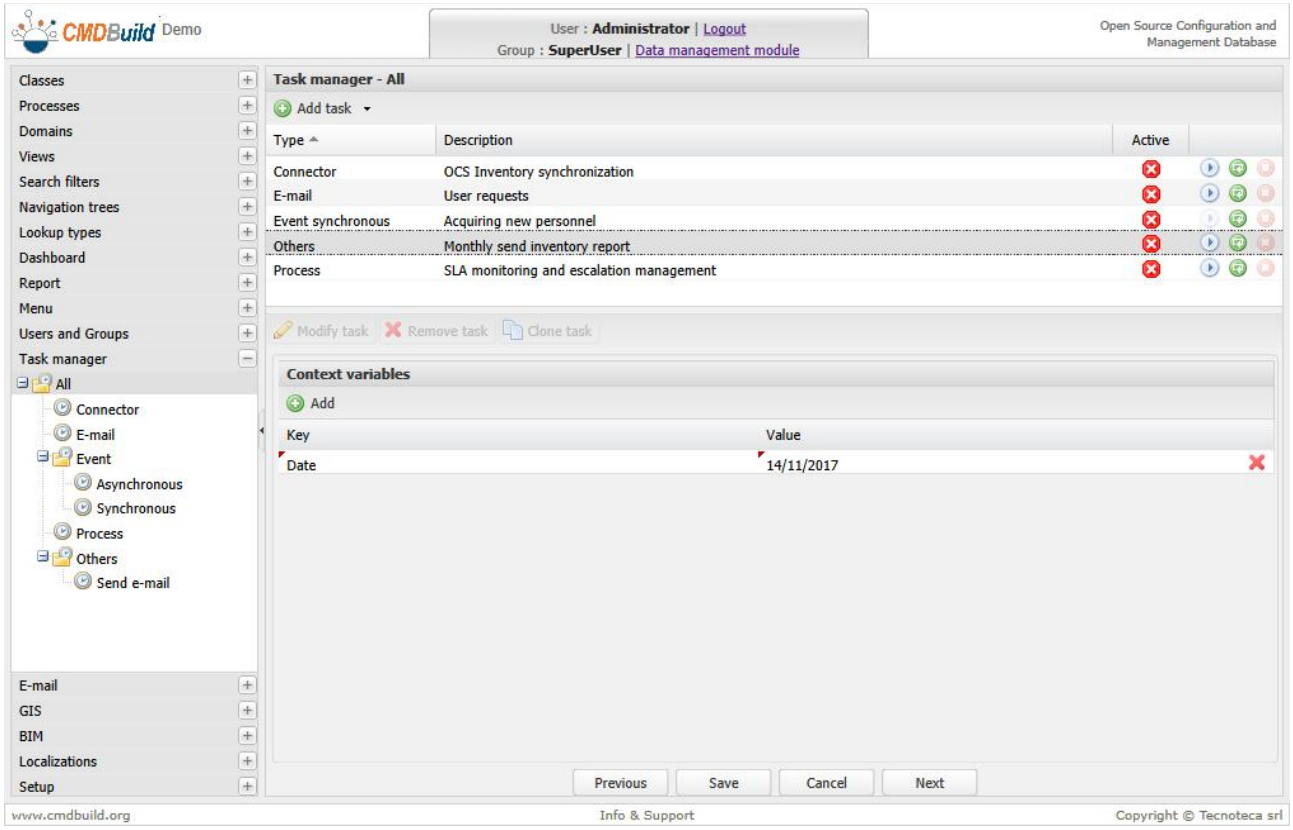
- “Description” of the new task
- “Start on save”: the task starts when the configuration is saved

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

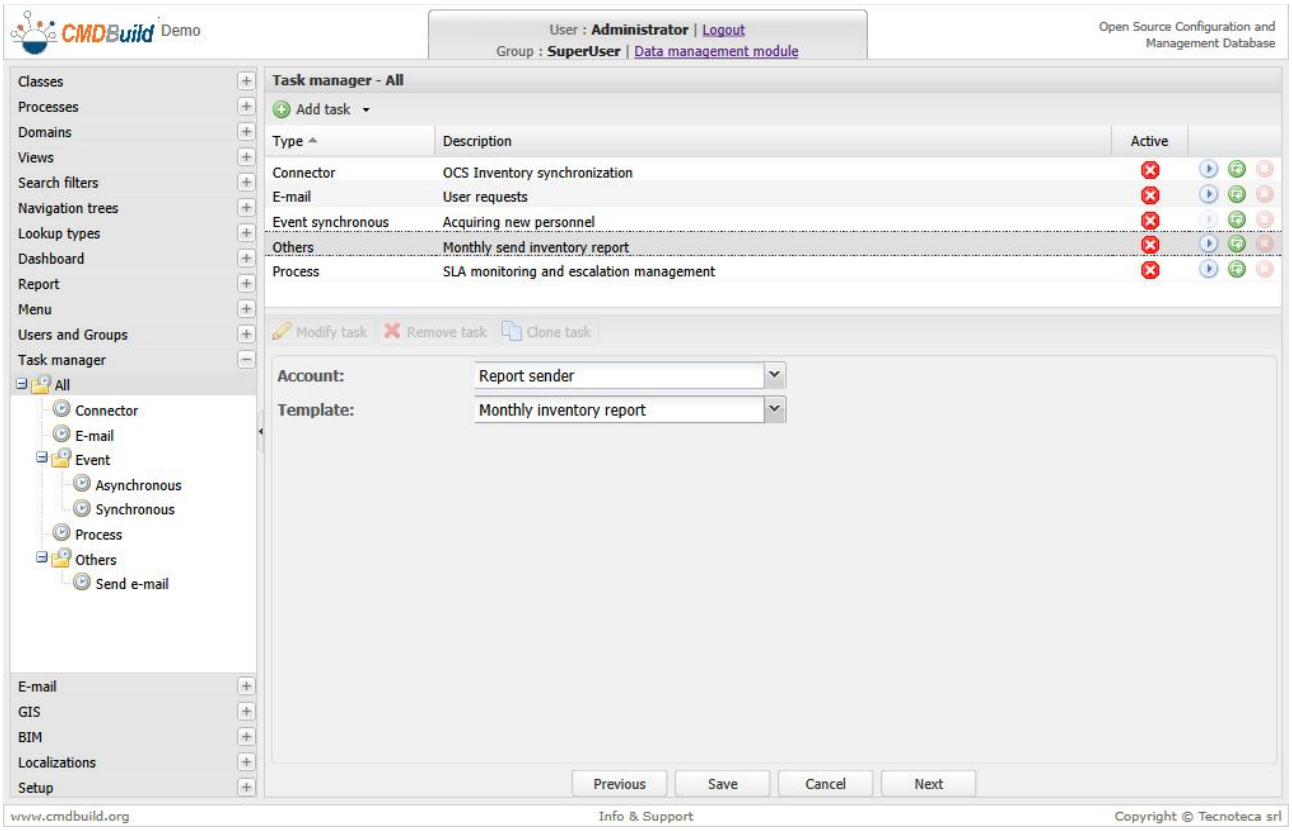
Page 3 of the wizard



It allows to define the context variables which can be used for the calculation of the report parameters.

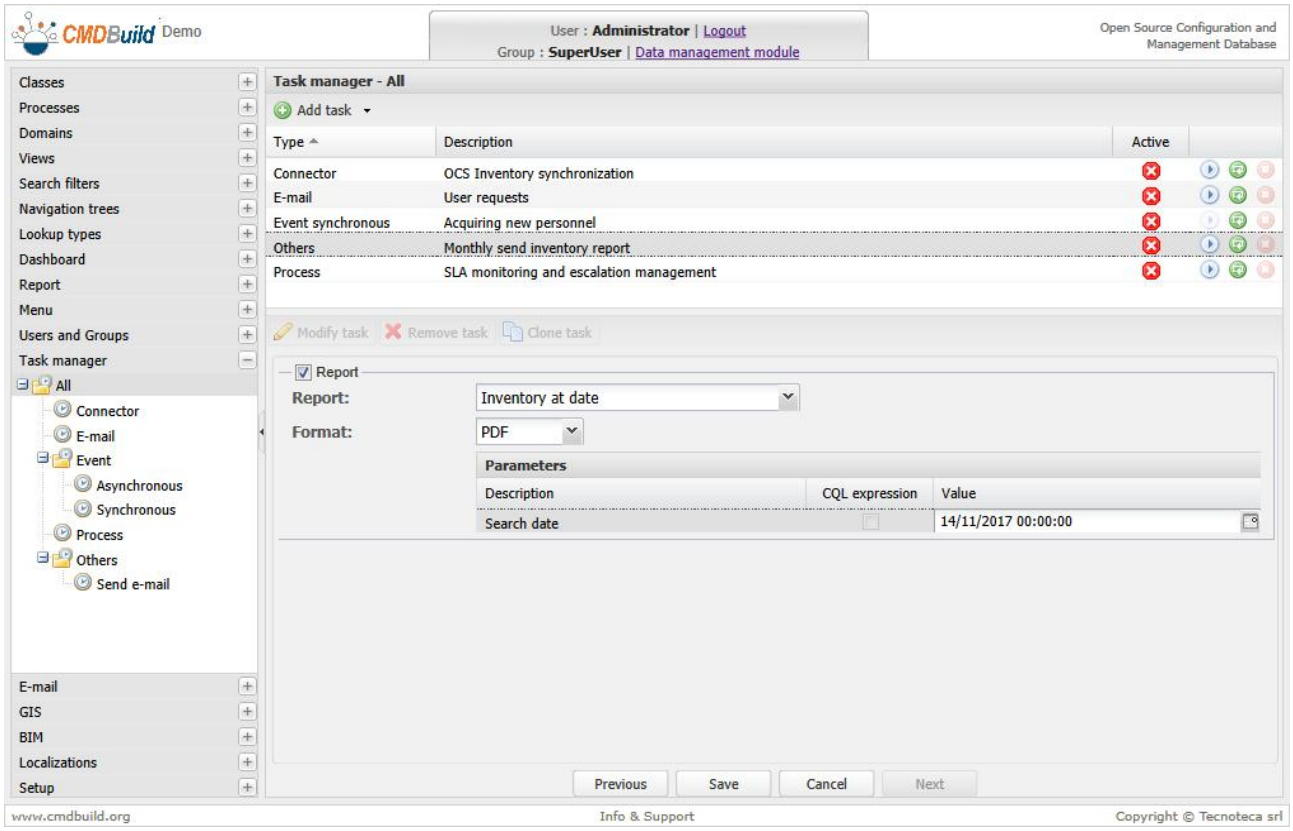
Each variable can be defined as constant or as CQL expression.

Page 4 of the wizard



It allows to choose the account to which you have to send the email and the template to write the text.

Page 5 of the wizard



It allows to:

- choose the report to enclose to the email
- choose the format
- in case the report has certain launch parameters, they should be optimized with constants or with a CQL expression (by summing for example a number of days)

Email management

In CMDBuild, e-mails are controlled from the core for a better control and a simpler updating of parameters.

This function allows you to register the information related to each account and to preconfigure certain templates of e-mails used in the CMDBuild mechanisms:

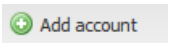
- Workflow system
- widget
- Task Manager

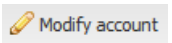
Account

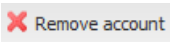
You can preconfigure the accounts used to send e-mails

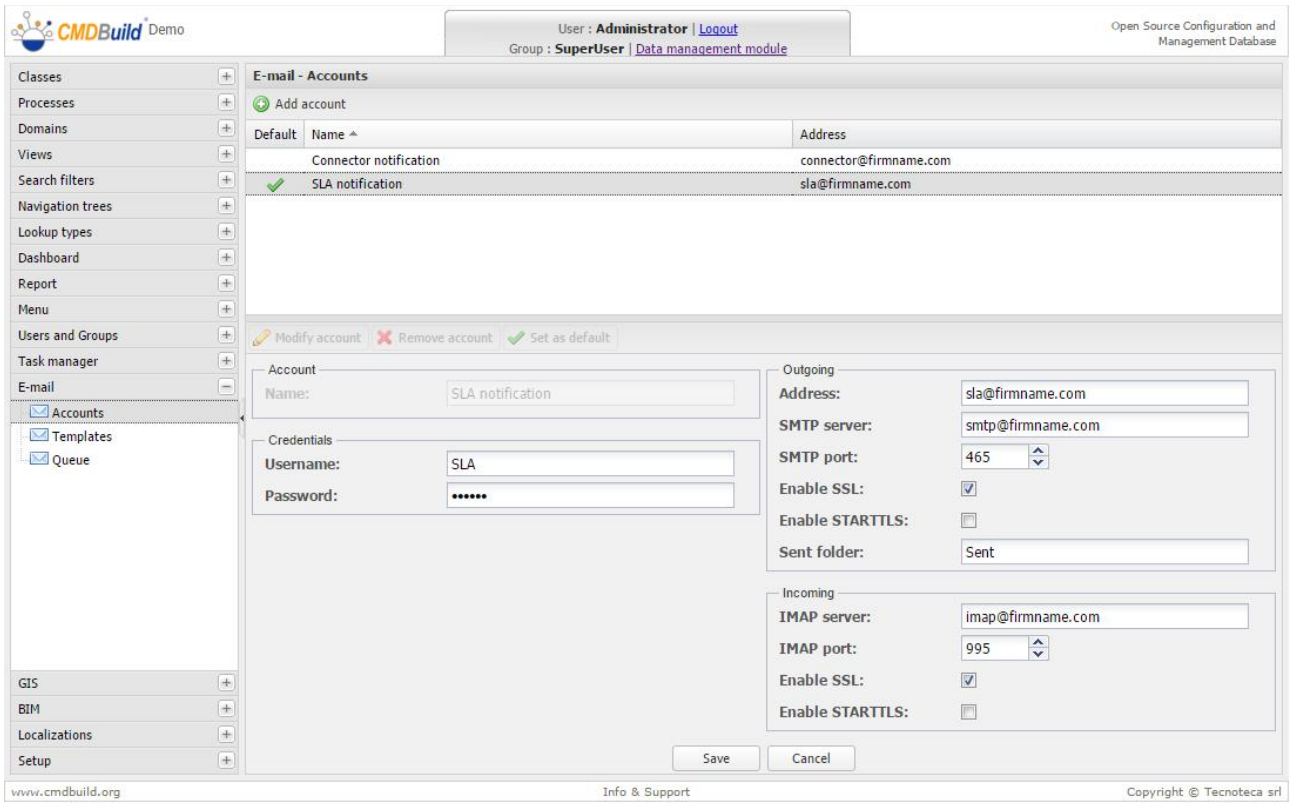
It's possible to perform the following operations:

- create a new account
- modify an account
- remove an account
- set an account as default









In order to create a new account, the following information is required:

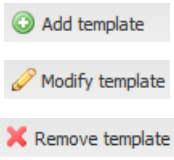
- “Account”
  - “Name”, account name
- “Credentials”
  - “Username”
  - “Password”
- “Sent e-mails”
  - “Address”
  - “SMTP server”
  - “SMTP port”
  - “SSL activation”
  - “StartTLS Activation”
  - “Folder Sent e-mails”
- “Inbox”
  - “IMAP server”
  - “IMAP port”
  - “SSL activation”
  - “StartTLS Activation”

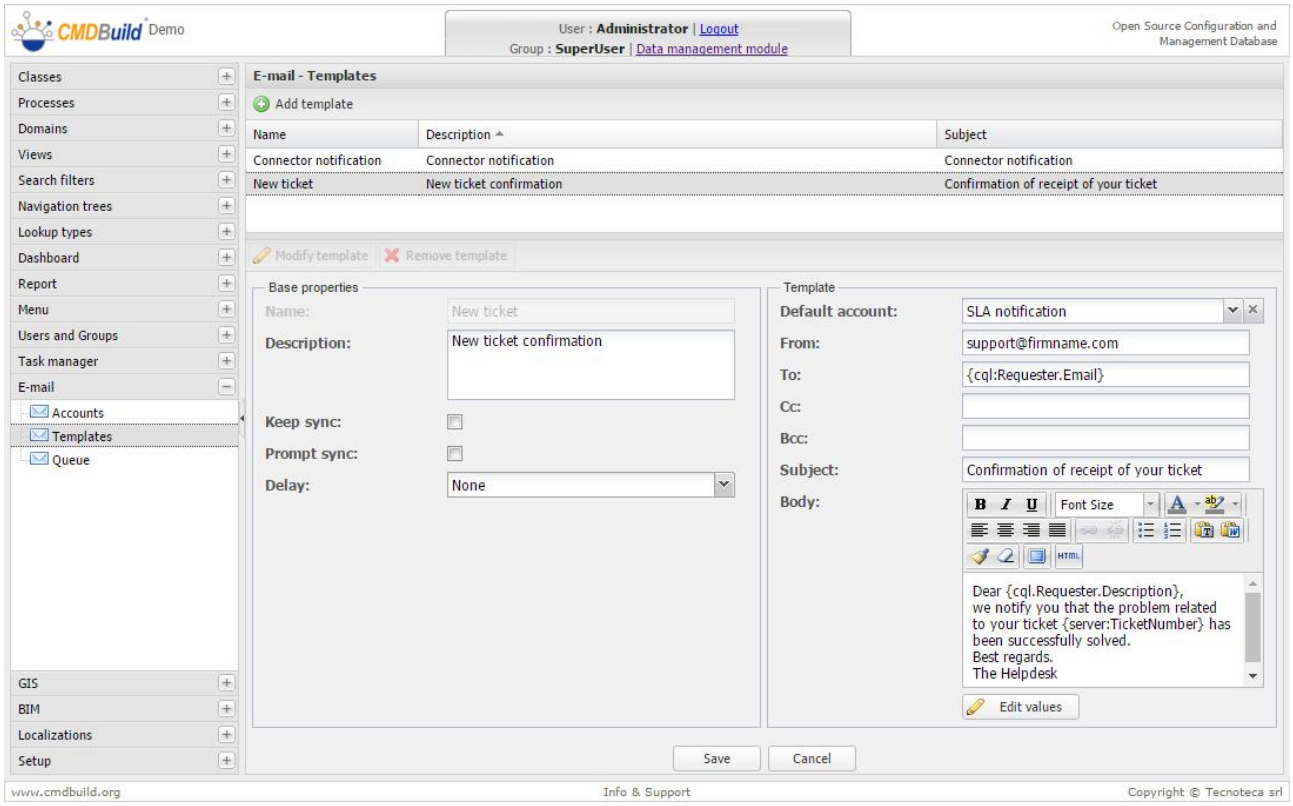
Email template

You can create preconfigured e-mail templates, which can be reused in the CMDBuild functions that use it.

It's possible to perform the following operations:

- create a new template
- modify a template
- remove an account





In order to create a new account, the following information is required:

- “Basic properties”
  - “Name”, name of the template
  - "Description"
  - “Keep sync”, it shows you that the text of the e-mail is automatically updated when attributes used like parameters vary (default = yes)
  - “Confirm sync”, it shows you that you have to confirm every time the e-mail text varies 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)

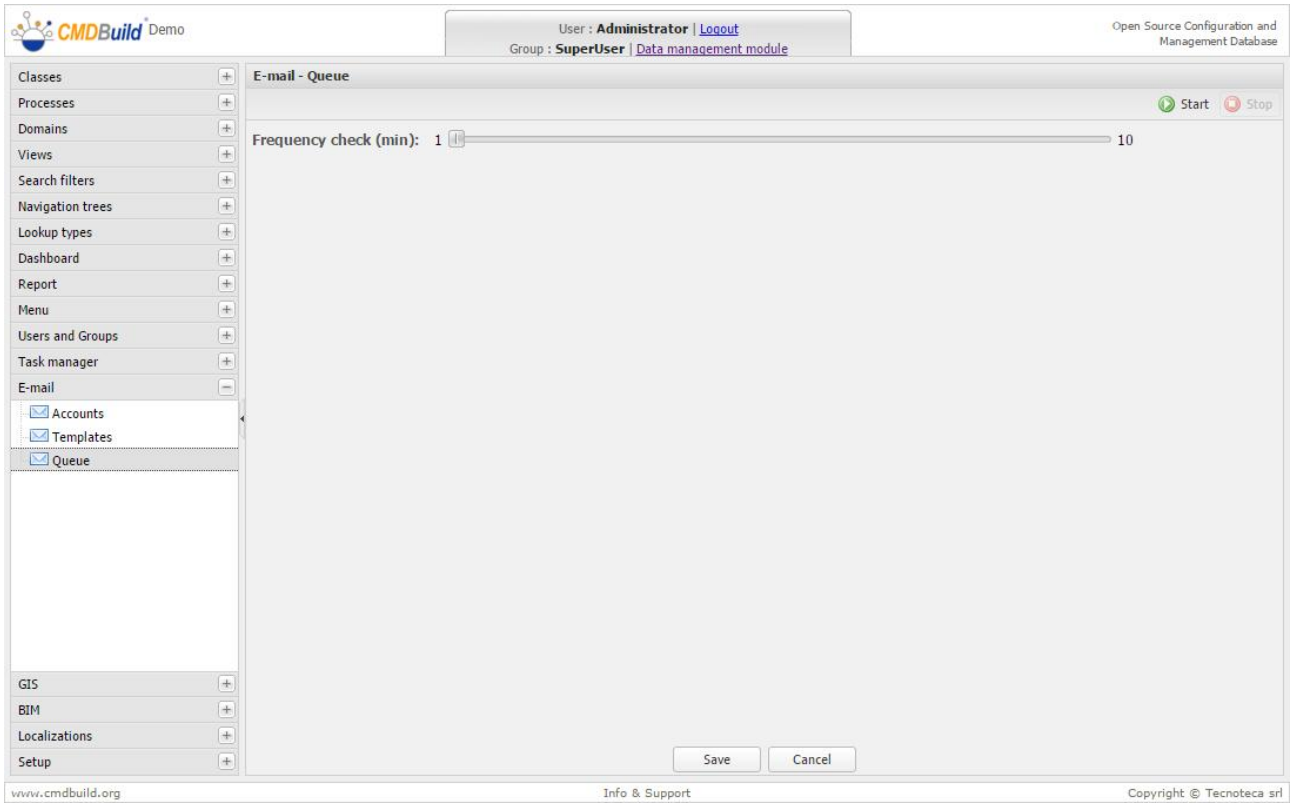
- “Template”
    - “Default Account”
    - “To”, addressee (or list of several addresses separated by comma)
    - “Cc”, copy carbon address (or list of several addresses separated by comma)
    - “Bcc”, blind copy carbon address (or list of several addresses separated by comma)
    - “Subject”, e-mail subject (it can include parametric fields expressed through CQL)
    - “Body”, e-mail body (it can include parametric fields expressed through CQL)

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

Email queue

The e-mail created by CMDBuild pass through a "queue", which is periodically controlled. After that, the e-mails are sent through the server.

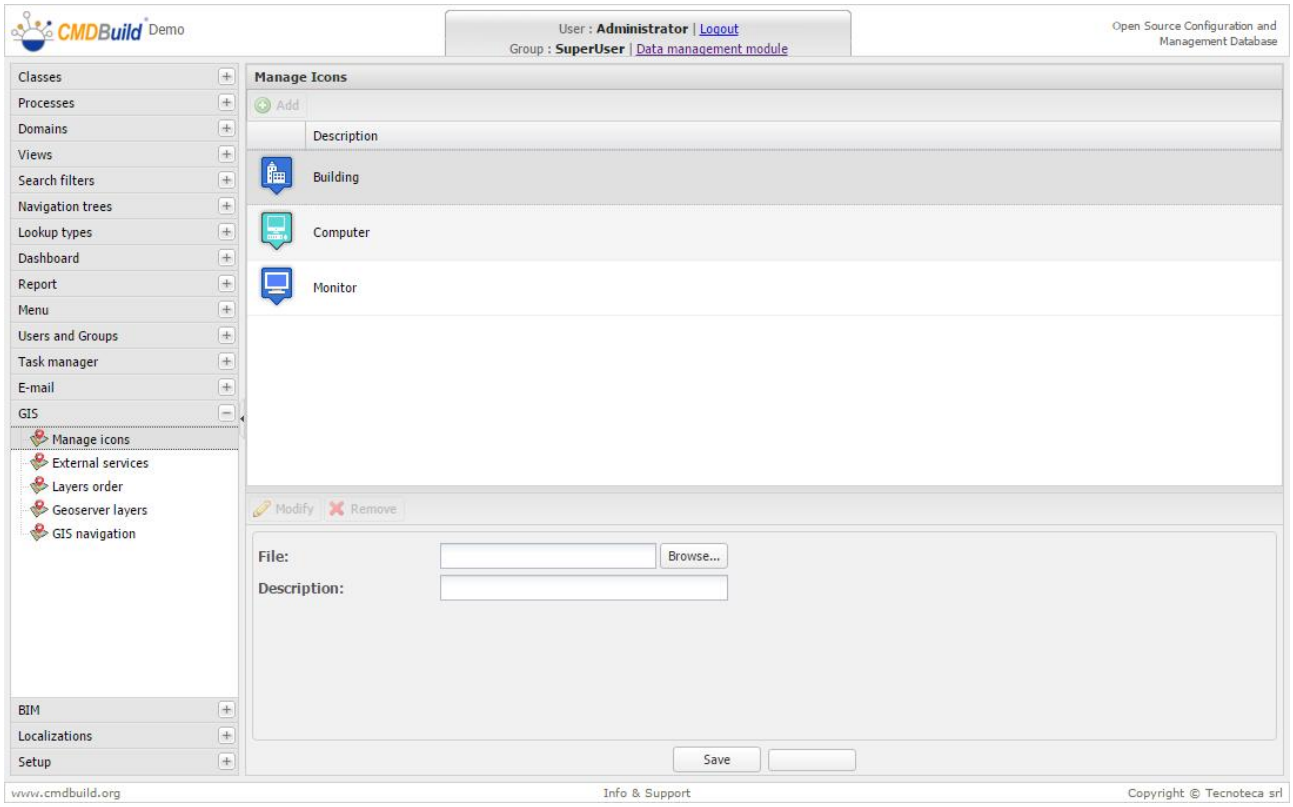
Thanks to this function, you can set the frequency for the "queue" control (1-10 minutes)



GIS

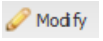
Icons management

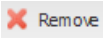
This feature allows you to create and update icons that are used to represent markers.



There is the possibility of performing the following operations:

- uploading a new icon 

- editing properties of an existing icon 

- deleting an icon 

External services

This feature allows you to activate the external services necessary for the map representation.

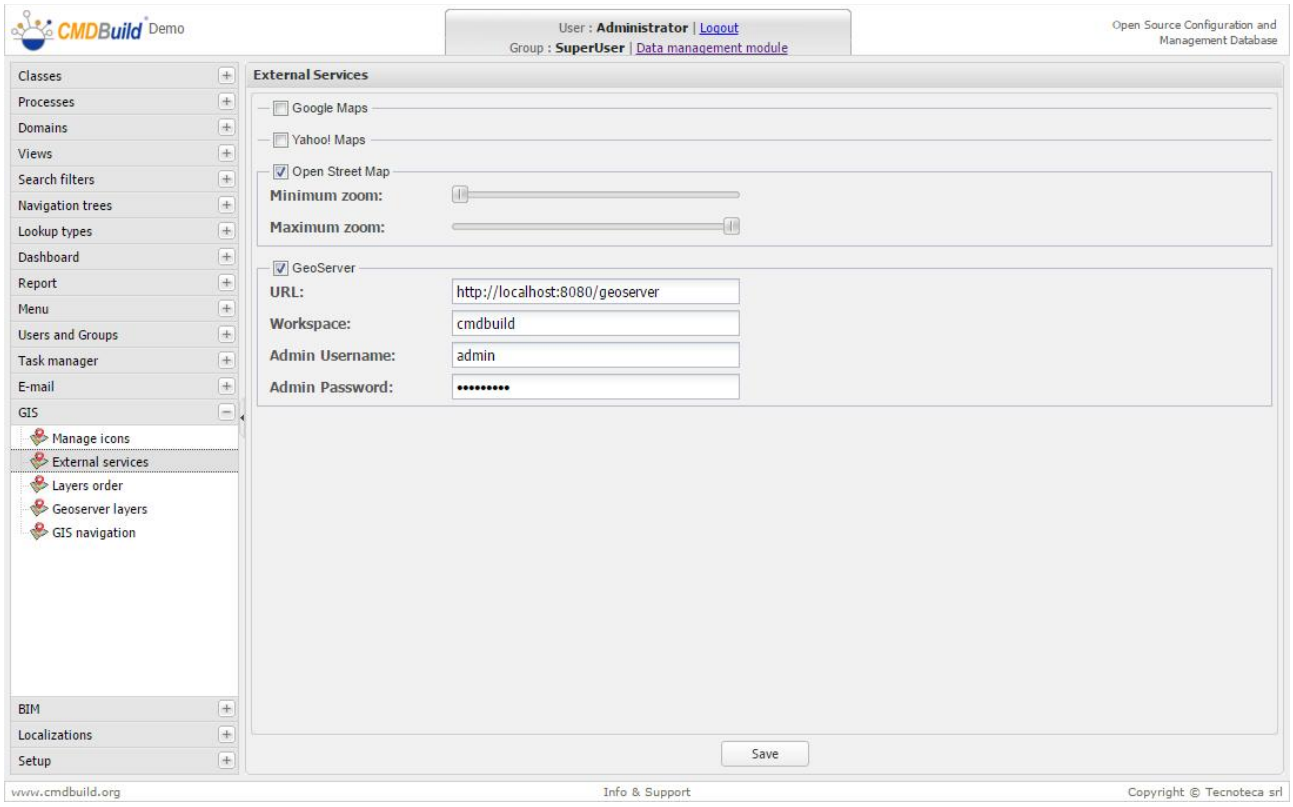
The external services currently supported are:

- Open Street Map, Google Maps and Yahoo! Maps)
- GIS Server (GeoServer) to upload and display raster and vector data (plans, etc.)

For the map services it's possible to set max and min zoom levels available in CMDBuild.

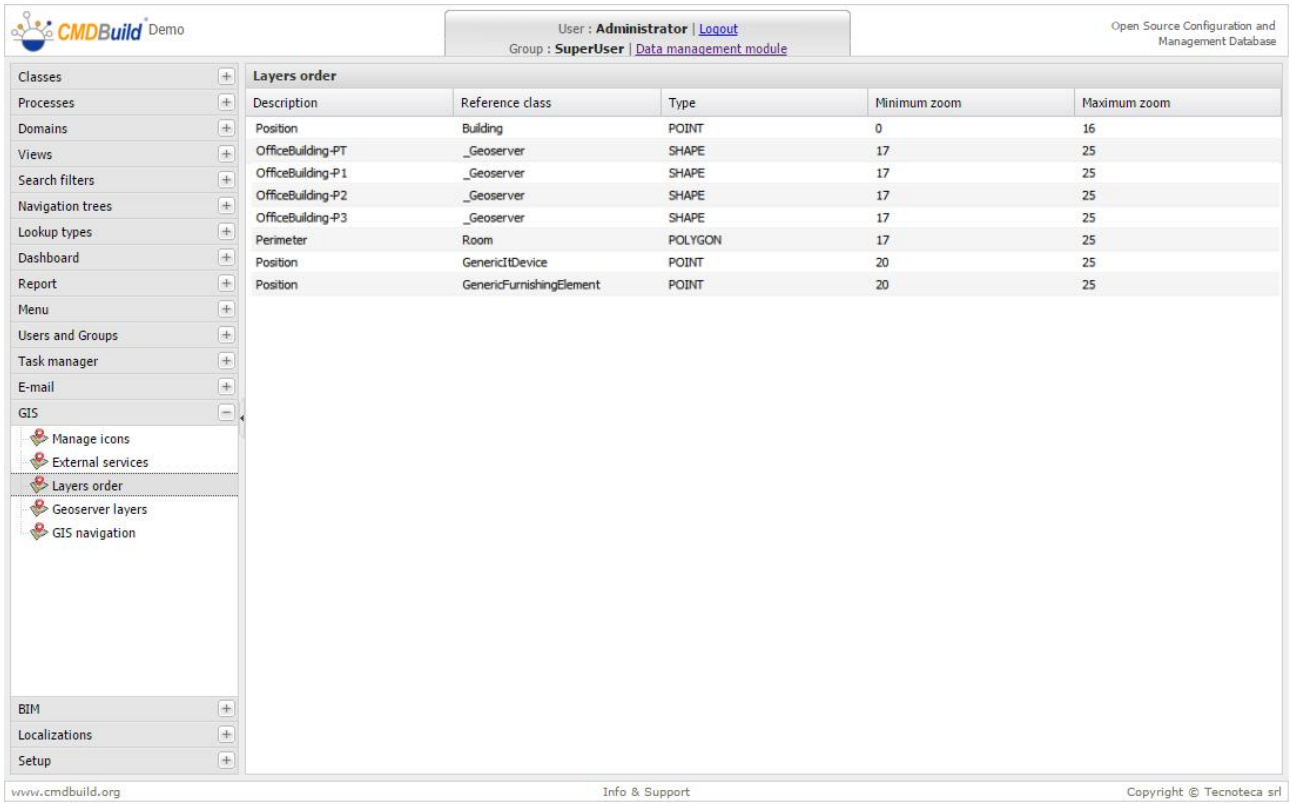
To use GeoServer you have to specify:

- URL
- workspace name
- admin username
- admin password



Layer order

This feature allows you to set layers order (there's a layer for every geographical attribute defined). The layers order can be modified by moving the lines in the grid (drag and drop).



Geoserver layer

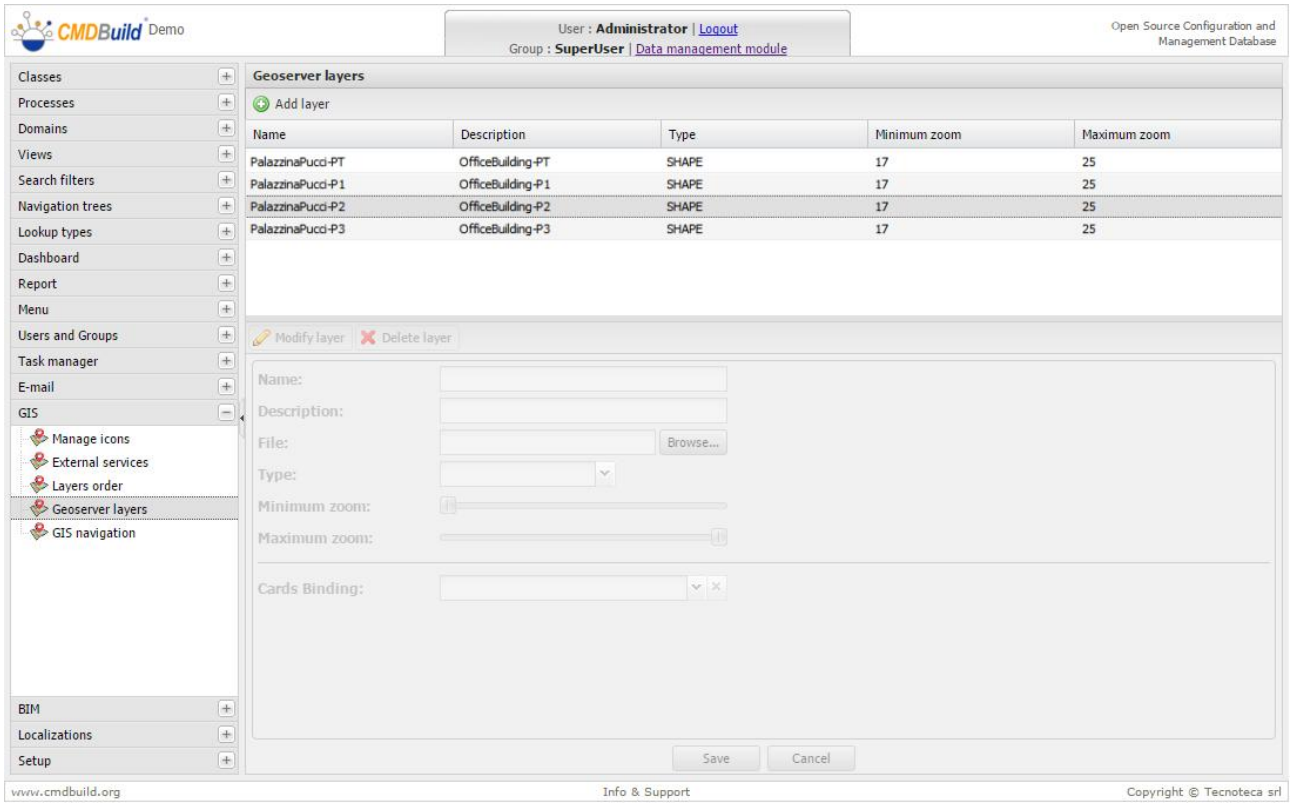
This feature allows you to:

- upload to GeoServer the layers you want to display in CMDBuild
- edit layers order

To upload a new layer you have to insert:

- name
- description
- layer file
- file type (GeoTiff, WorldImage, Shape)
- min zoom level
- max zoom level
- the possible class and the CMDBuild card associated to that layer, useful in the Management Module to insert the layer into the list of the items contained in an element of the hierarchical GIS navigation menu, which is described in the next paragraph (e.g. to find the Geoserver layer among the items related to a "Floor" card)

The layers order can be modified by moving the lines into the grid (drag and drop).



GIS Navigation

The function allows the definition of the domain tree, which will be then provided to the user in the Management Module in order to help him/her with the navigation among the various class / card types georefered on the map.

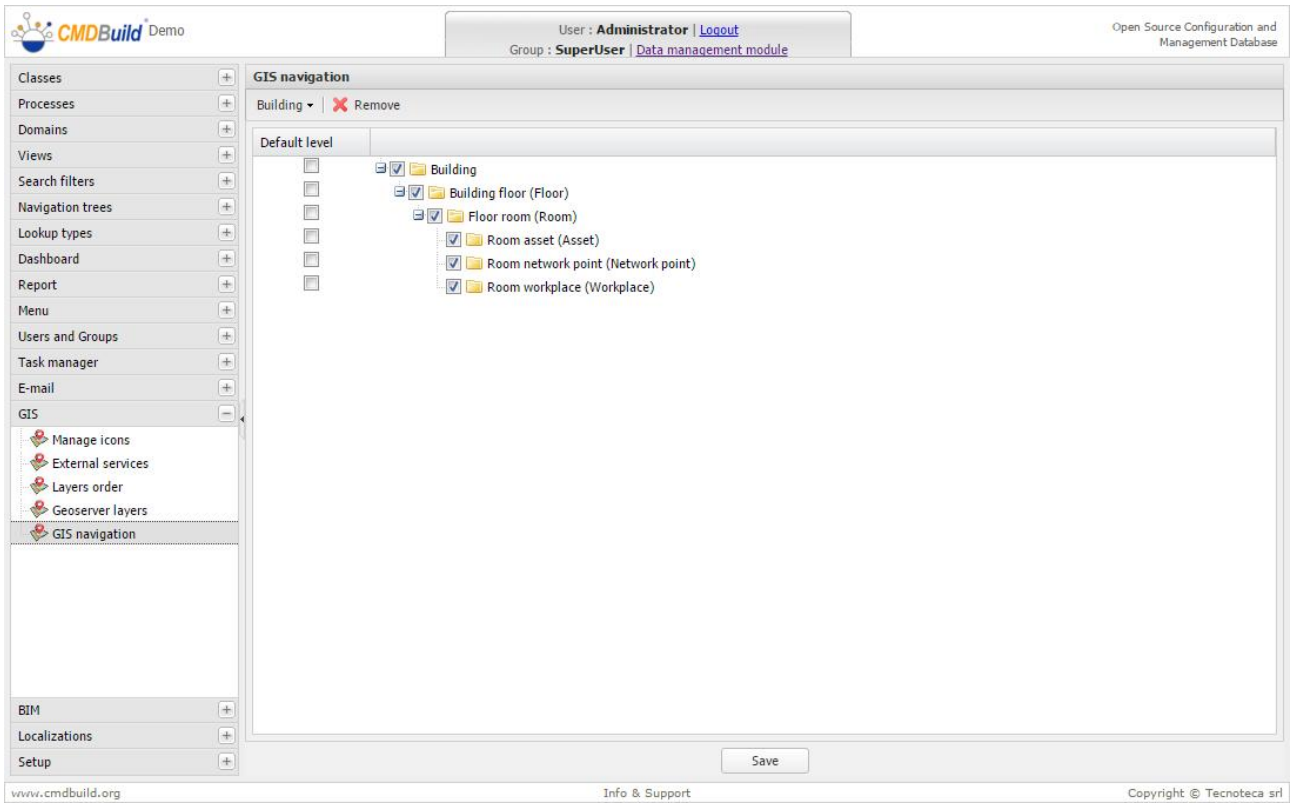
In other words, this function defines the structure of the hierarchical menu in order to move from a class to the contained ones (in term of space).

A higher class can include more lower classes, for example a room can include information assets, jobs (understood as assignees or logic grouping of assets), furnishing (tables, chairs, cupboards, technical equipment, pictures), etc.

Providing the necessary domains, please find below a sample of configuration made up of the following classes:

“Building” =>
“Floor” =>
“Room” =>
“Item”

Through the "Default level" check, you are required to mark the "vertical development" layer, which is the one capable of overlapping and needs to be displayed selectively (usually related to the "Floors" layer).



BIM

Projects

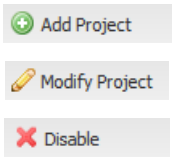
This function allows to manage BIM projects managed in CMDBuild. Every project is usually associated to a building or to a built up area.

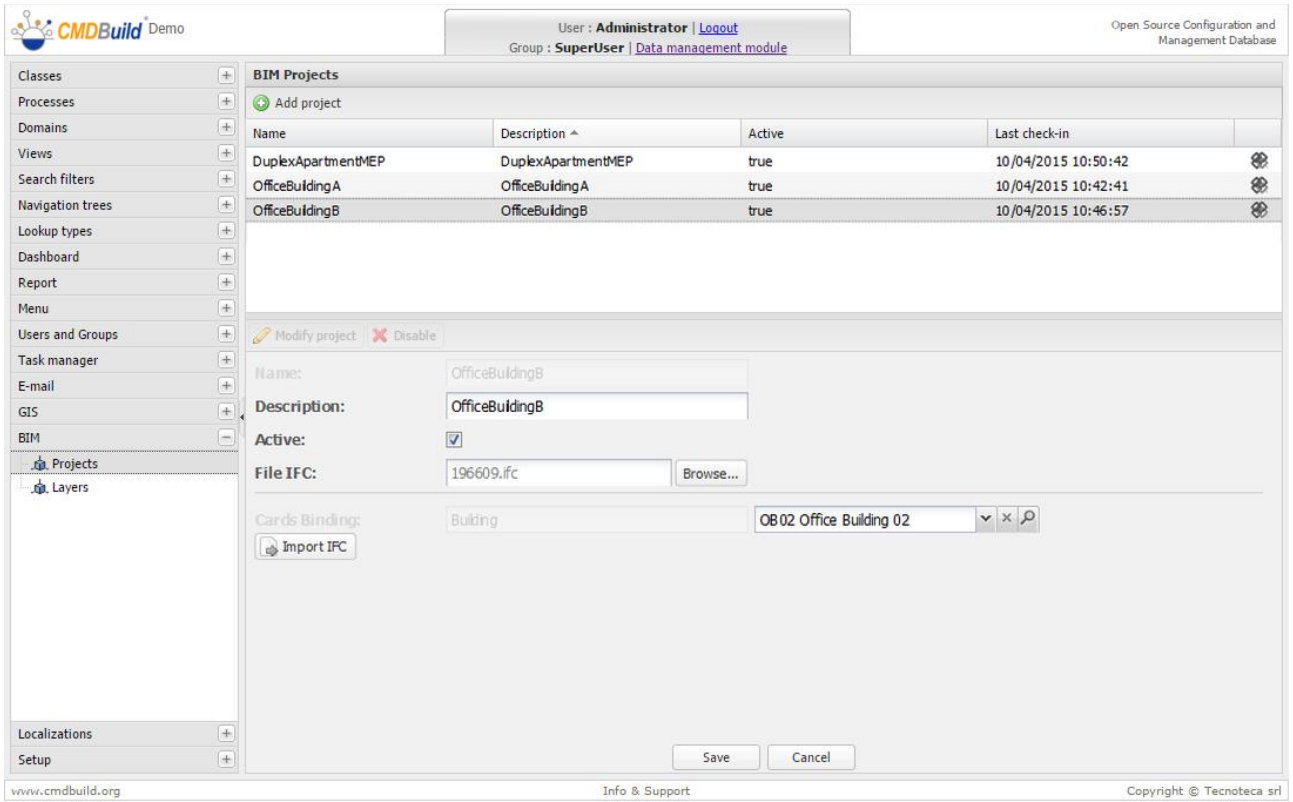
A BIM project coincides to a IFC file in various versions.

For IFC files, CMDBuild uses as embedded repository the open source tool BiMserver, which contains files uploaded from CMDBuild.

It's possible to perform the following operations:

- create a new project
- modify a project
- disable a project
- export an integrated project containing information managed / edited in CMDBuild





In order to create a new project, the following information is required:

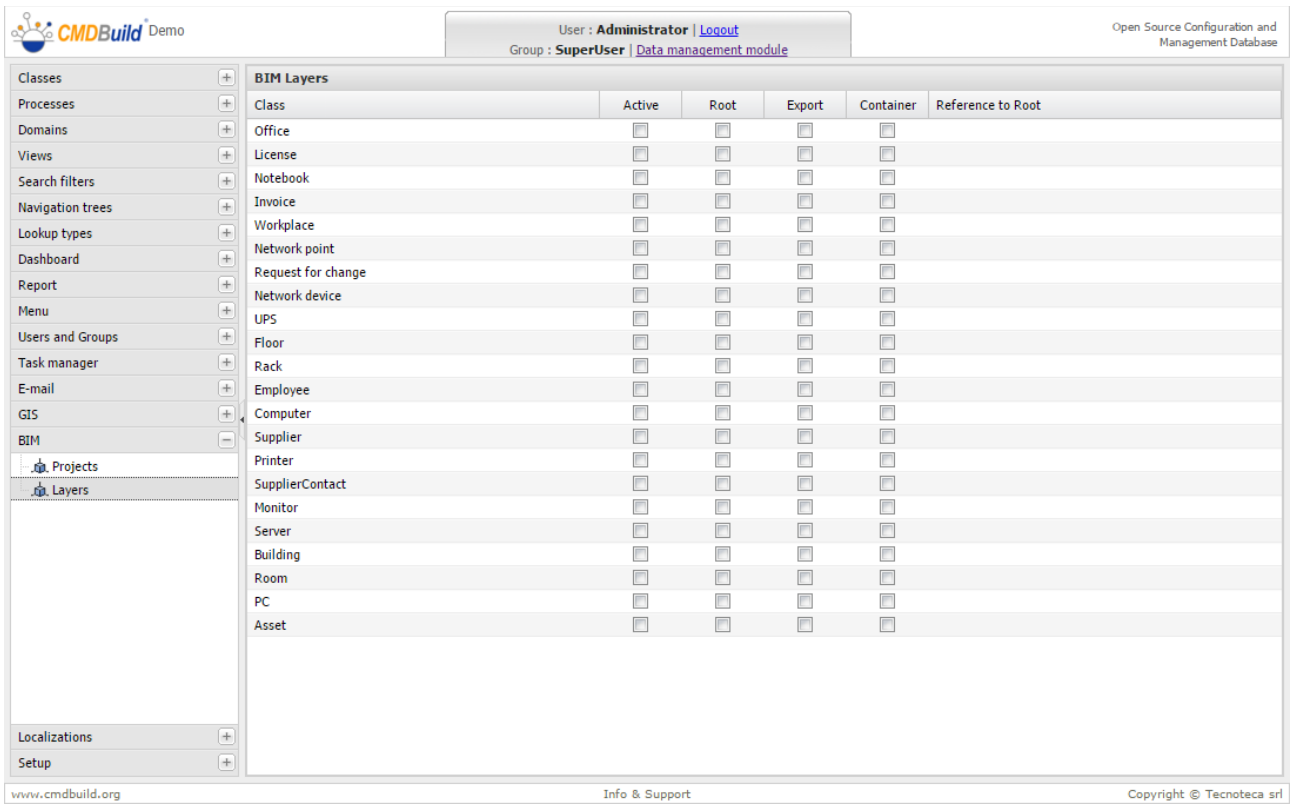
- “Name”, name of the project
- “Description”, description of the project

- “IFC file”, it requires the uploading of the IFC file to in order to file in into BiMserver
- “Active”, if the project is active
- "Card connection", it requires to select the card of the CMDBuild class for the project connection; it is usually a building or a built up area (it must coincide with the CMDBuild class indicated as "Root" in the following function of the layer management)

Layer

The UI shows the list of CMDBuild classes and "check" columns as follows:

- “Active”, where you can select those CMDBuild classes having a correspondence with the entities in the IFC file
- “Root”, where you can select a CMDBuild class corresponding to the "root" class of the IFC file, e.g. the "building" class (single selection)
- "Export", where you can select the exportable classes of CMDBuild, so that all the modifications executed in CMDBuild will be reported in the architectural designing external tool (they differ from other classes due to the inclusion of the geometrical coordinates management)
- “Container”, where you can select the CMDBuild class used as "container" of objects included in the IFC file, e.g. the class "room" (single selection)



Localization

In CMDBuild there are two localization typologies:

- first-level localization, related to the only base CMDBuild interface (menu texts and standard buttons, headers, footers, etc.)
- second-level localization, related to all application elements (class names, attributes, domains, lookups, etc.)

The first-level localization is managed through external system files (jason format) and it is supplied by external contributors.

The second-level localization is managed in this Administration Manual, by working on single texts and directly in the function for the element definition, or through the features available in this menu.

The official releases of CMDBuild include all first-level localizations supplied by contributors, while for the second-level localizations there is a repository where contributors can upload additional languages for CMDBuild Ready2Use and openMAINT.

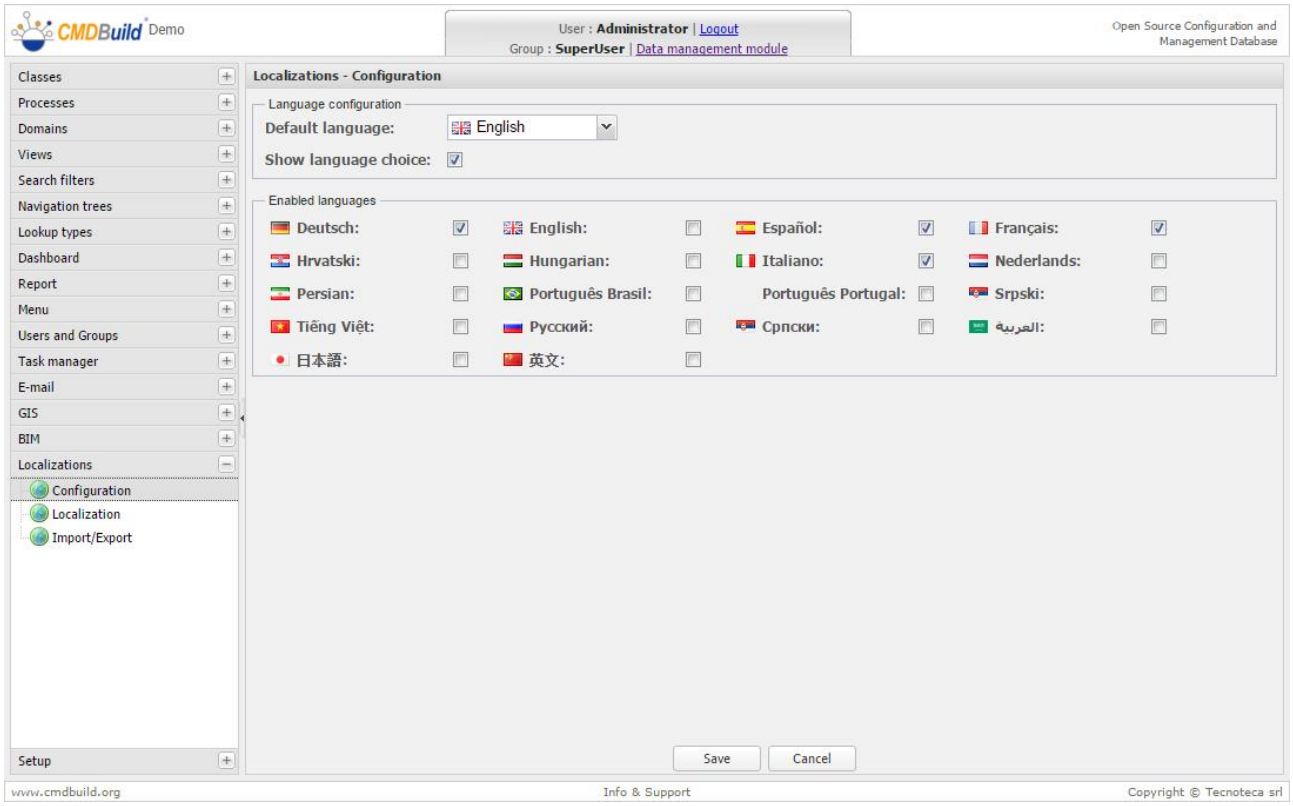
Configuration

In the configuration page you can select:

- the default language of the CMDBuild instance (among the available ones)
- the introduction of the default language when logging in (in this case, you will still have the possibility to access then with another language)

Then you can select the languages enabled for the localization, among those ones with the first-level localization.

Here's a sample of interface screenshot.


CMDBuild Demo
User : Administrator | Logout
Group : SuperUser | Data management module
Open Source Configuration and Management Database
Classes
Processes
Domains
Views
Search filters
Navigation trees
Lookup types
Dashboard
Report
Menu
Users and Groups
Task manager
E-mail
GIS
BIM
Localizations
Configuration
Localization
Import/Export
Setup
Localizations - Configuration
Language configuration
Default language:
English
Show language choice:
Enabled languages
Deutsch:
English:
Español:
Français:
Hrvatski:
Hungarian:
Italiano:
Nederlands:
Persian:
Português Brasil:
Português Portugal:
Srpski:
Tiếng Việt:
Русский:
Српски:
العربية:
日本語:
英文:
Save
Cancel
www.cmdbuild.org
Info & Support
Copyright © Tecnoteca srl

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
- Views: view description
- Search filters: search filter description
- Lookup: lookup list heading description
- Report: report description
- Menu: folder description, description of element inserted into the menu

Each kind of element is managed in a proper TAB of the interface.

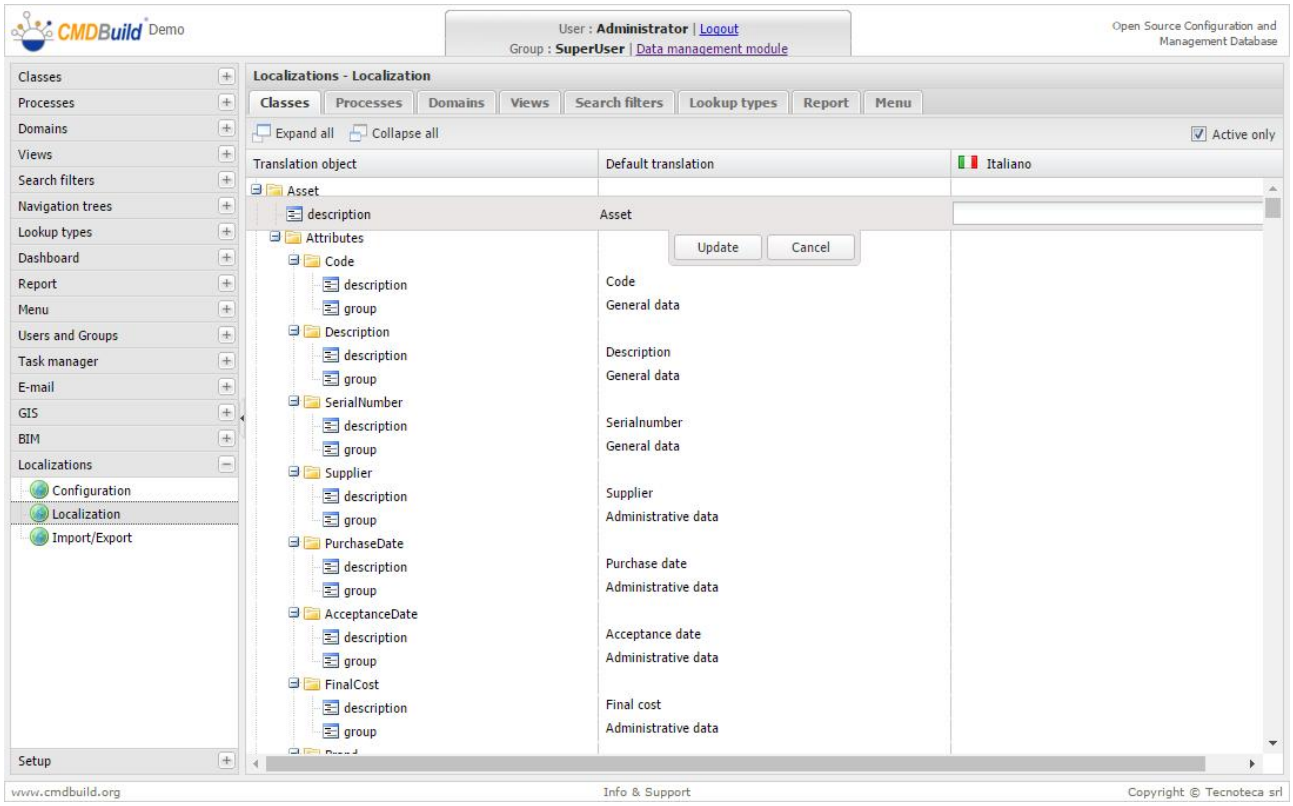
In every TAB you can:

- expand all headings 

- reduce all headings 

For each language enabled to the localization, there is a column where you can view, add and change the translation of that element.

Here's a sample of interface screenshot for the TAB Classes, the management of other TABs is similar.


CMDBuild Demo
User : Administrator | Logout
Group : SuperUser | Data management module
Open Source Configuration and Management Database
Classes
Processes
Domains
Views
Search filters
Navigation trees
Lookup types
Dashboard
Report
Menu
Users and Groups
Task manager
E-mail
GIS
BIM
Localizations
Configuration
Localization
Import/Export
Setup
Localizations - Localization
Classes
Processes
Domains
Views
Search filters
Lookup types
Report
Menu
Expand all
Collapse all
Active only
Translation object
Default translation
Italiano
Asset
description
Asset
Attributes
Update
Cancel
Code
description
Code
group
General data
Description
description
Description
group
General data
SerialNumber
description
Serialnumber
group
General data
Supplier
description
Supplier
group
Administrative data
PurchaseDate
description
Purchase date
group
Administrative data
AcceptanceDate
description
Acceptance date
group
Administrative data
FinalCost
description
Final cost
group
Administrative data
www.cmdbuild.org
Info & Support
Copyright © Tecnoteca srl

Import / Export

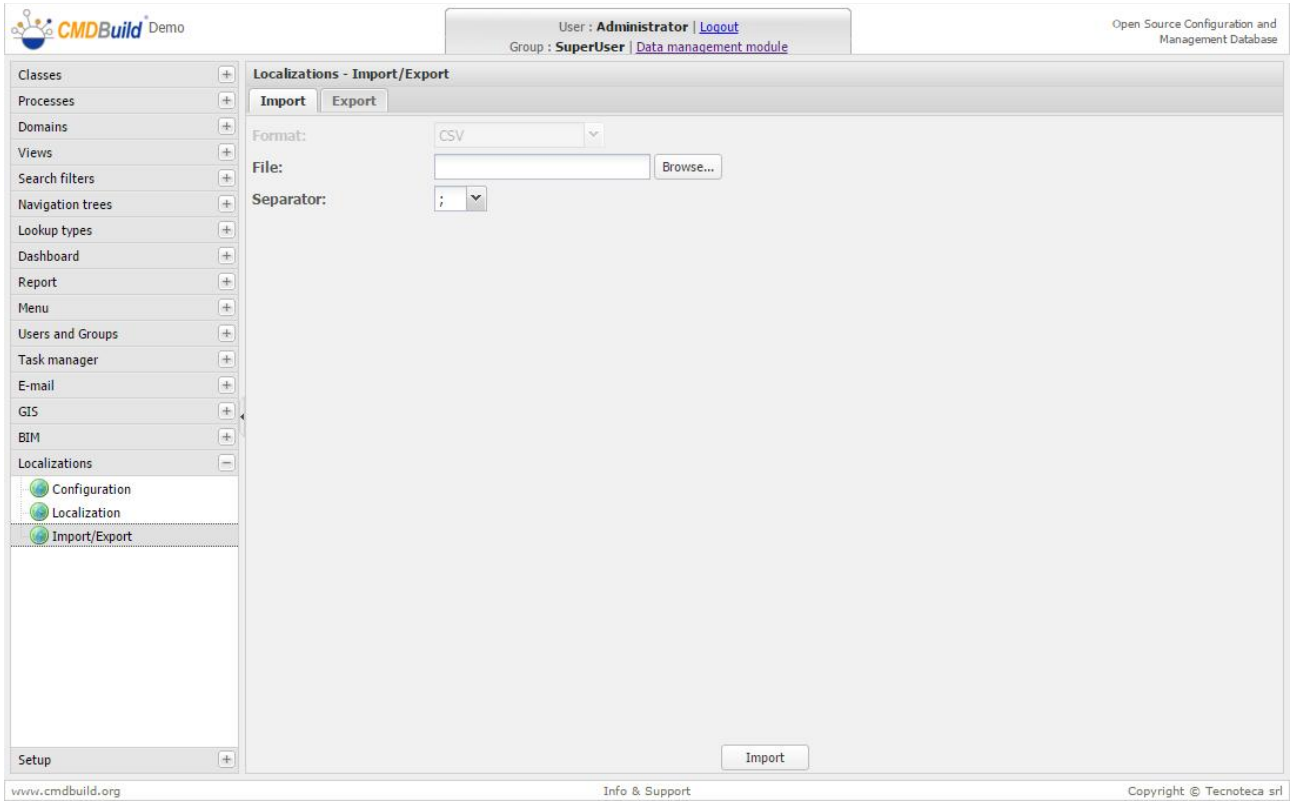
The import / export feature allows you to import a localization provided by other contributors, or to export your own localization.

There two features are available in two different TABs.

In the page from where you can import there are the following fields:

- Format (CSV, not-editable)
- File
- Separator (it can be selected among the symbols ";", "," and "|")

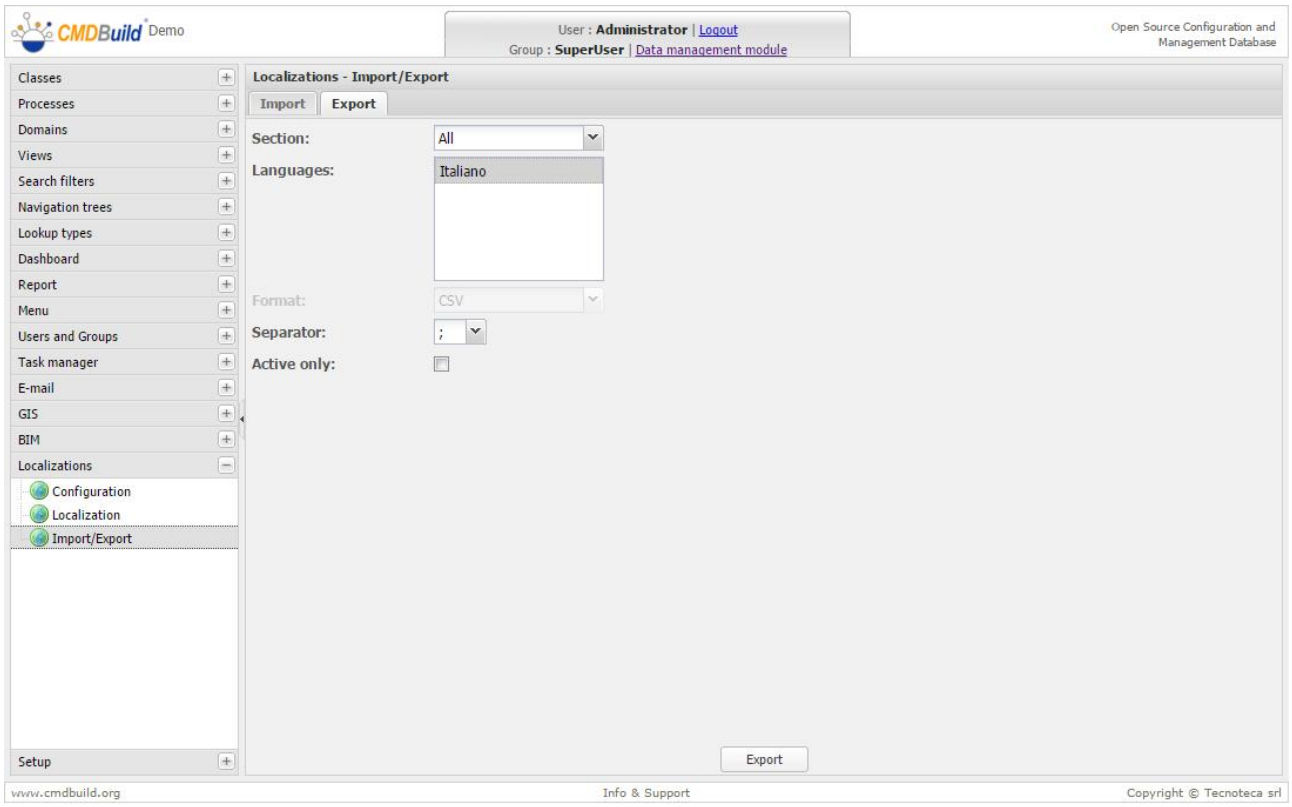
Here's a sample of interface screenshot.



In the page from where you can export there are the following fields:

- Section (chosen among All, Classes, Processes, Domains, Views, Search filters, Lookups, Reports, Menu)
- Languages ("multiselect" field where you can choose one or more enabled languages)
- Format (CSV, not-editable)
- Separator (it can be selected among the symbols ";", "," and "|")
- Only active (to export only localizations of active elements)

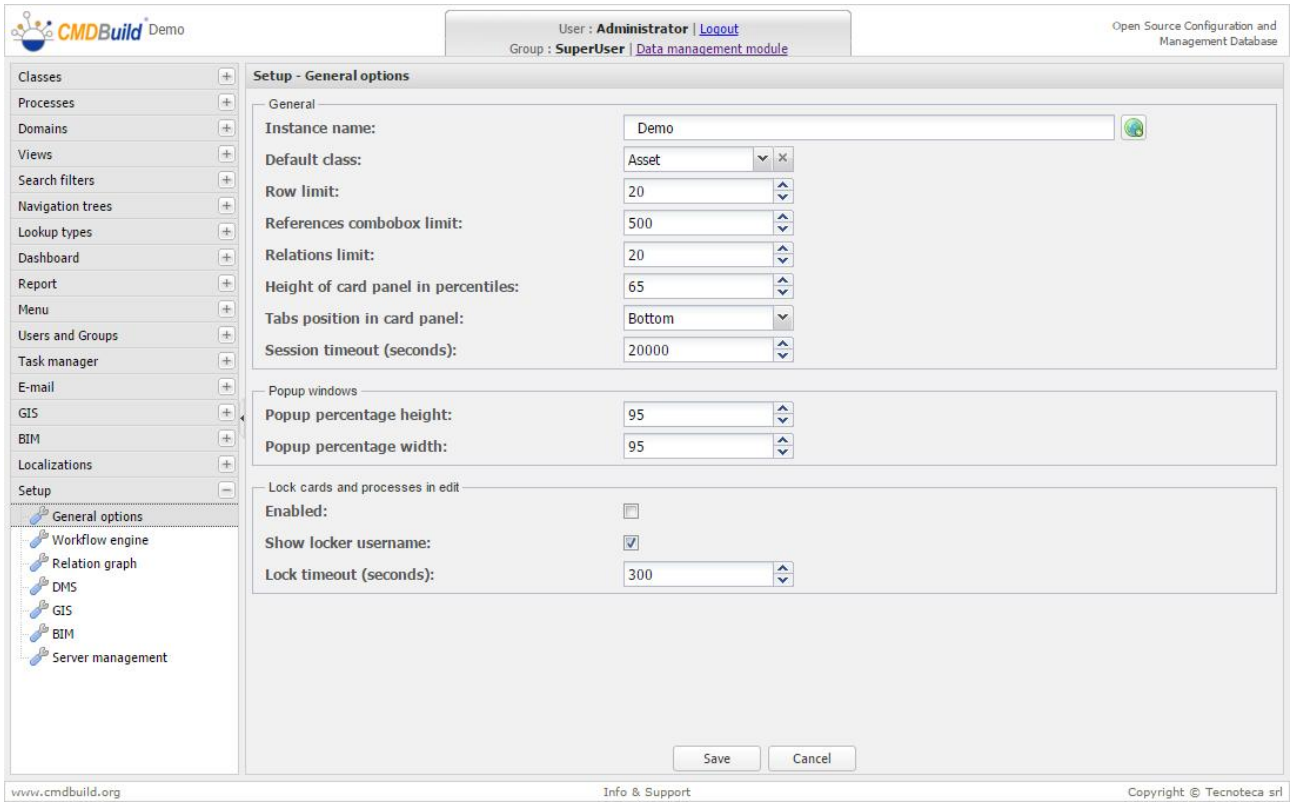
Here's a sample of interface screenshot.



Configuration

Setup page

This page contains main application settings.



In the "General" tab you can set:

- “Instance name” - a name (string) to recognize an instance (such as "Test" to distinguish production and test installations)
- "Default class" - the class loaded on startup
- “Row limit”, it shows the number of rows shown in the top grid on every class or process
- “References and lookup combobox limit” - indicates the maximum number of options shown in a combobox; if exceeded, the options are displayed in a popup window with pagination and filters
- “Relation limit”, indicates the maximum number of relations shown separately; if exceeded, the relations are grouped togethe”
- "Height of card panel in percentiles", indicates the space management between grid and data sheet areas (50 by default, set for example to 65 to give priority to the data sheet area)
- "Sub-panel tab position in card's panel" allows you to specify whether sub-panels titles should

appear on the top or bottom

- “Session timeout (seconds)”

In the "Popup windows" section you can set:

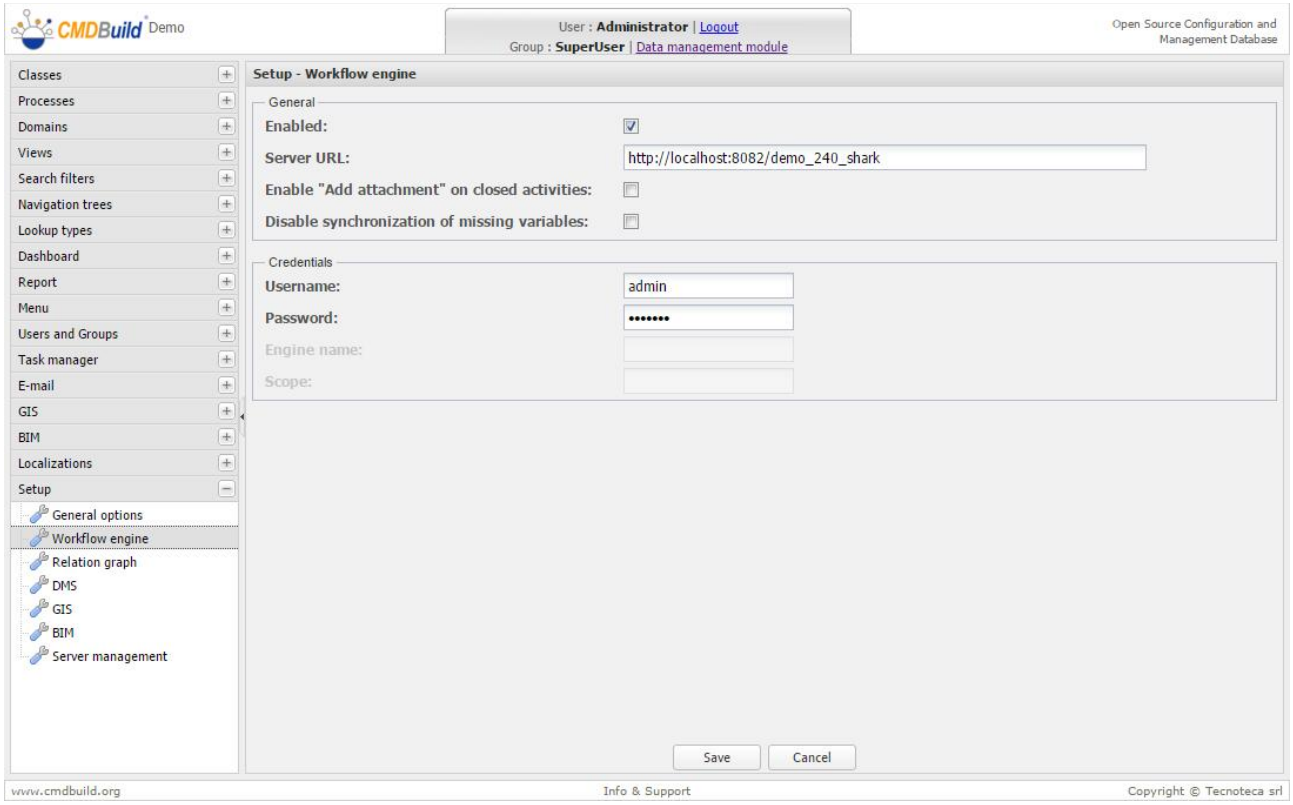
- “Popup percentage height” - the percentage height of the popup window
- Popup percentage width - the percentage width of the popup window

In the “Lock cards and processes in edit” section you can set:

- “Enabled”, to enable (default) or disabled the lock management of those data cards and processes in edit
- "Show the name of the user who locked the card", 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”, to specify the time reserved to an operator before releasing the card in edit

Workflow page

The page contains the workflow configuration options.



In the "General" tab you can set:

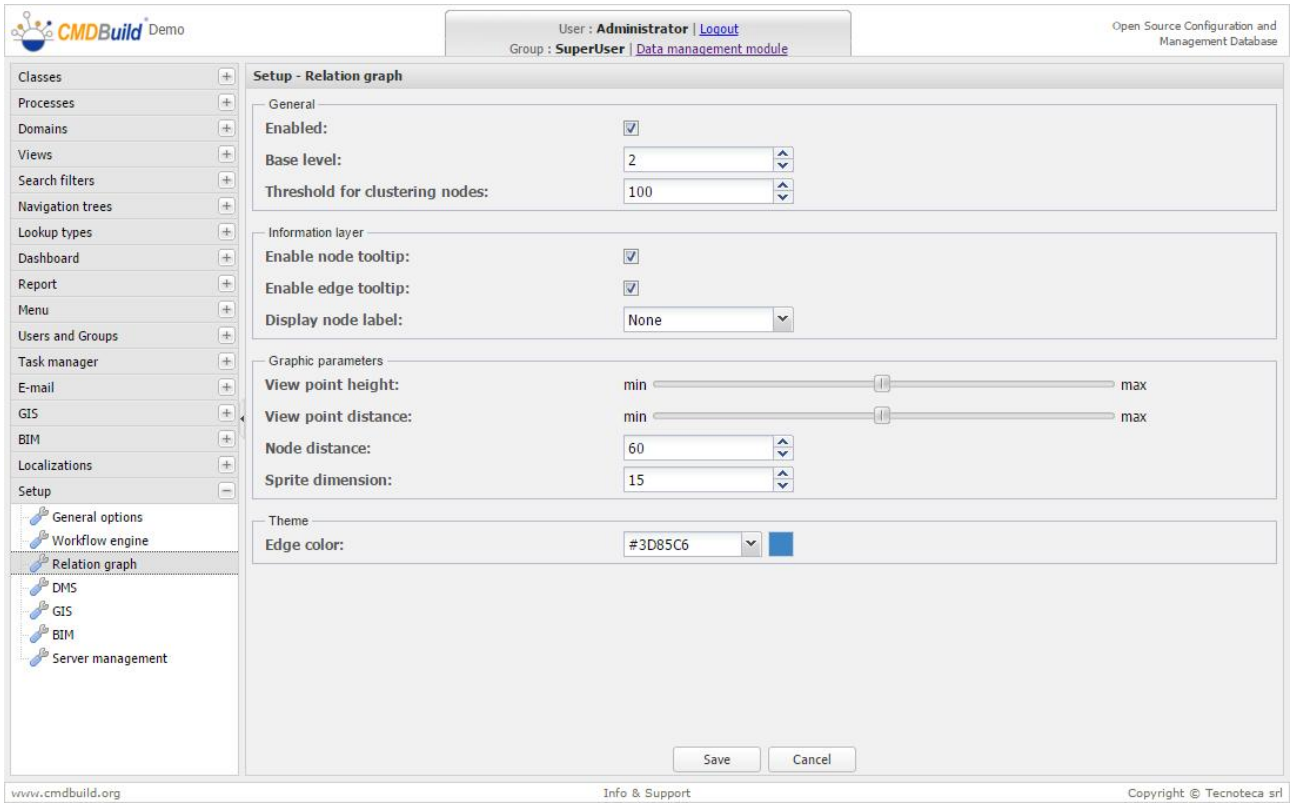
- "Enabled" - enable or disable the workflow
- "Server URL" - workflow service url
- “Enable 'Add attachment' when the activities are closed”
- “Disable sync of missing variables”, useful to manage irregular cases where the XPDL descriptor of a workflow refers to variables that are not defined in the CMDBuild process definition

In the “Credentials” section you can set:

- “Username”
- “Password”
- “Engine name”
- “Scope”

Relations graph page

The page contains the relation graph settings.



You can set:

- "Enabled" - enable or disable the graph visualization
- "Default level" - the default number of levels automatically expanded
- "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)
- “Enable tooltip nodes”
- “Enable tooltip archi”
- “View tooltip nodes”, with possibility of choosing between "None", "All", "Selected”
- “Viewpoint high”, in the 3D chart
- “Viewpoint distance”, in the 3D chart
- “Nodes distance”, useful to view closer or further nodes in the chart
- “Sprite dimensions”, useful to state the dimension of the icons representing the different kinds of nodes in the chart
- “Archi colours”

Some above-mentioned configuration settings can be modified within the current work session

through the function "Setup" available in the visualization interface of the Relations Graph.

DMS Page

CMDBuild uses Alfresco DMS as repository, interacting through the protocols SOAP and FTP.

Thanks to the contribution of the University of Bologna, the system has been then extended to other document repositories, integrating the support of the standard protocol CMIS (Content Management Interoperability Services).

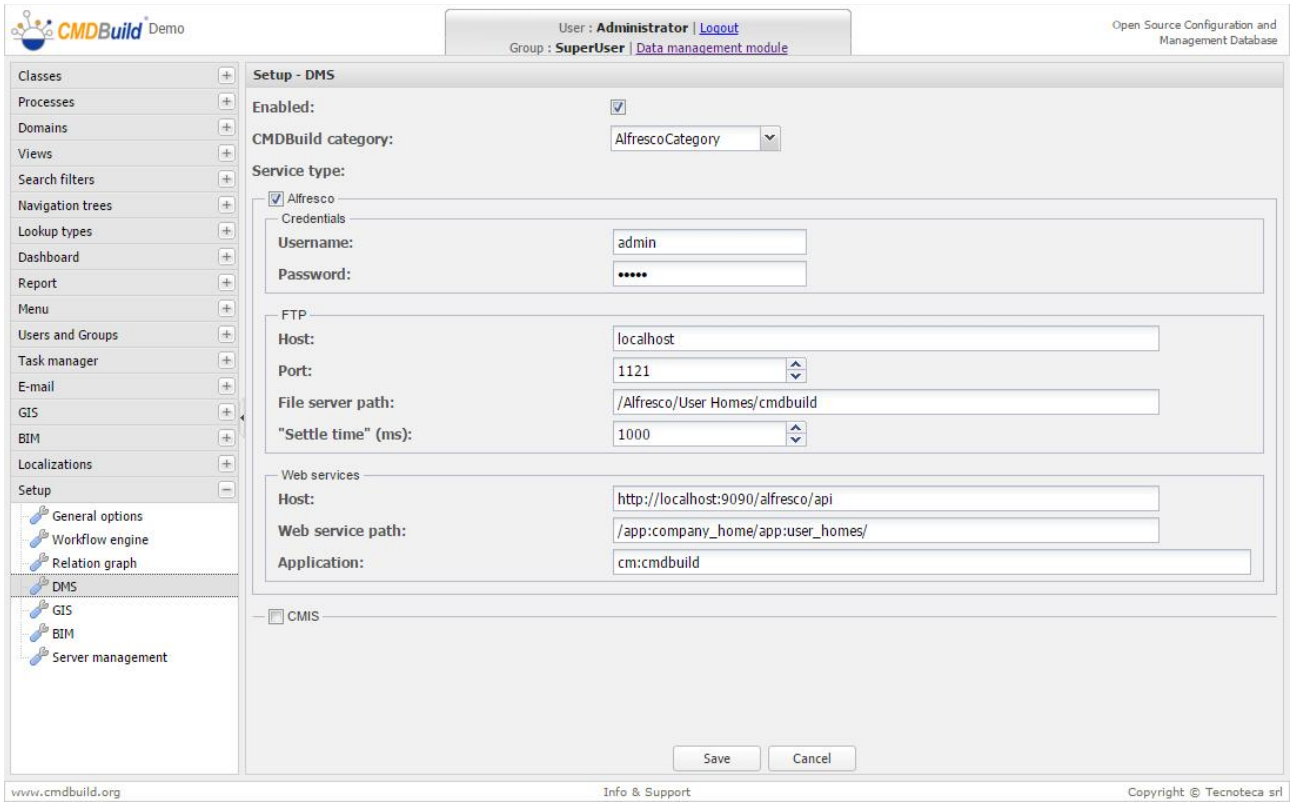
So, you can choose to work with the native interface of Alfresco (for the version 3.4) or to work with the CMIS protocol.

In both cases you need:

- "Enabled" - enable or disable the interface
- "CMDBuild category" - the Lookup list used to classify the documents stored in the DMS

In case you opt for working with the native interface of Alfresco, you need then to specify further settings, divided in four blocks:

- Alfresco
- FTP
- Web services



In the “Alfresco” section you can set:

- "Username" - Alfresco username
- "Password" - Alfresco password

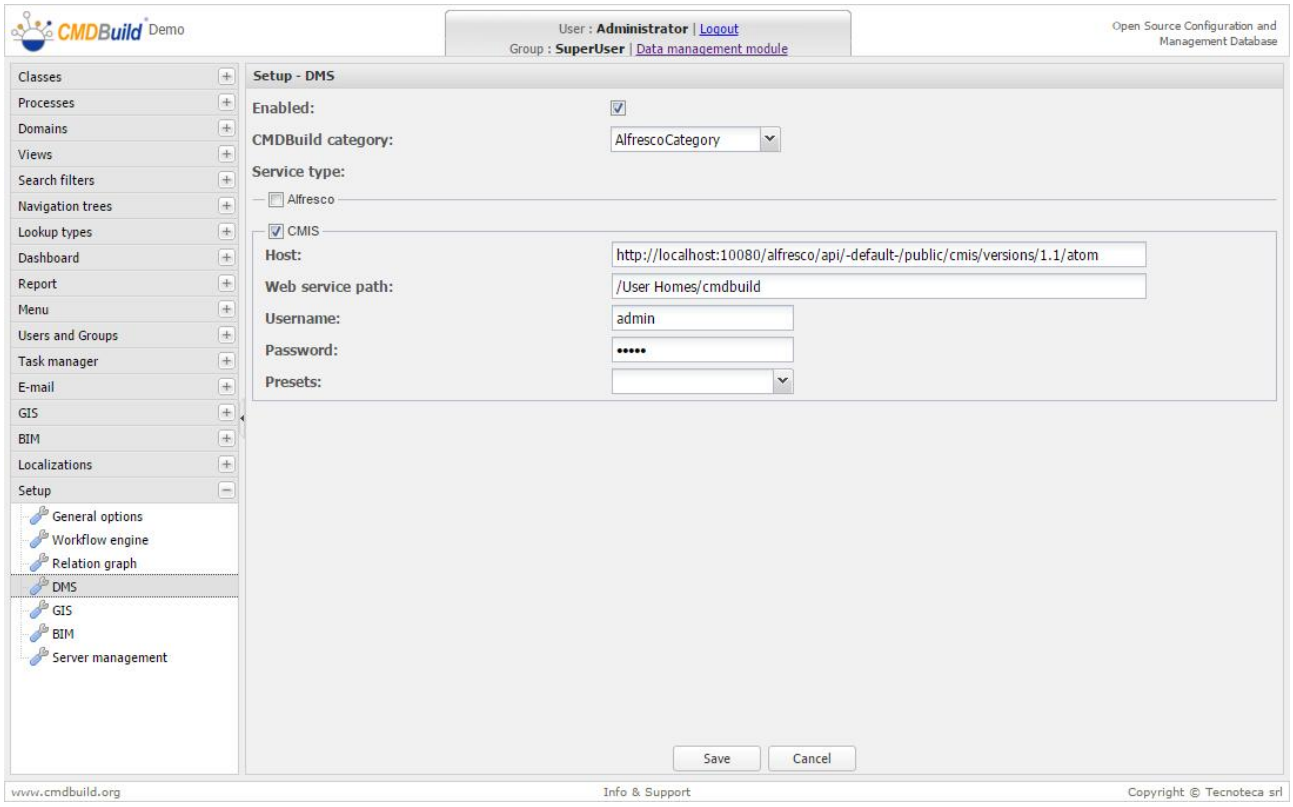
In the “FTP” section you can set:

- "Host" - Alfresco FTP server host
- "Port" - Alfresco FTP server port
- “File server path”
- "Operations delay " - delay to ensure that the upload is concluded before calling a grid refresh

In the “Webservice” section you can set:

- "Host" - Alfresco webservice host
- "Port" - Alfresco webservice port
- “Application”

On the other side, if you opt for working with the CMIS protocol, the interface will be as follows.



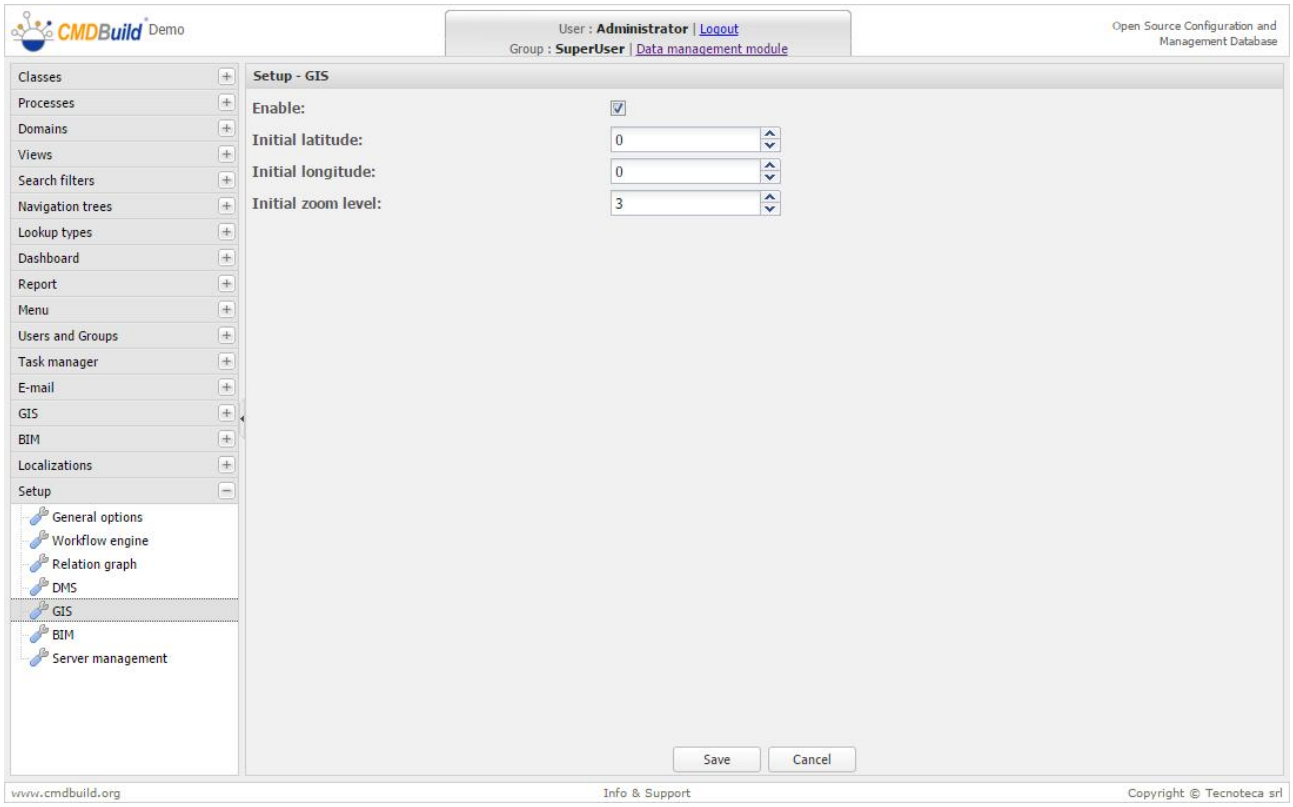
The required settings are as follows:

- "Host" - Alfresco DMS server host

- “Username” DMS username
- “Password” to access the DMS
- “Preset”, to select the "aspect" system to use

GIS page

The page contains GIS settings

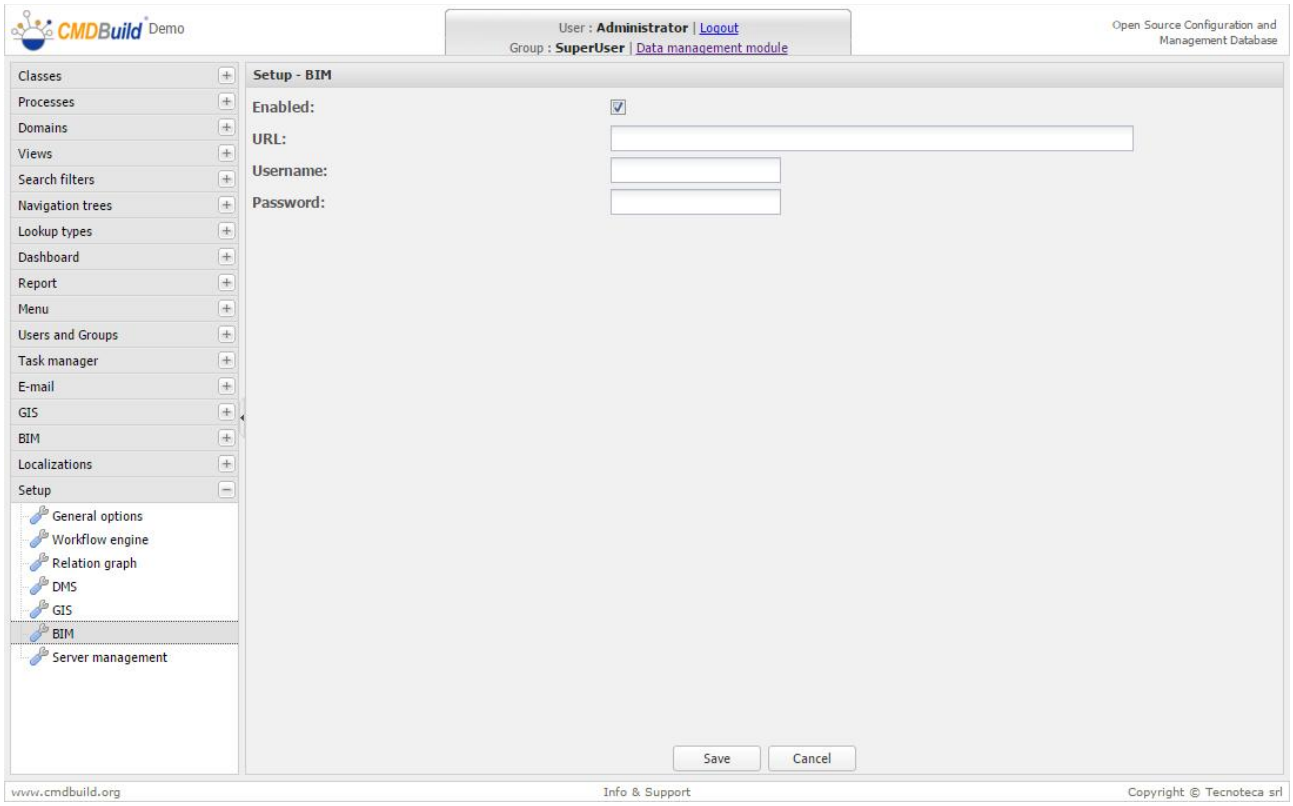


You can set:

- "Enable" - the database must be configured to support the PostGIS extension
- "Initial latitude" - map latitude at startup
- "Initial longitude" - map longitude at startup
- "Initial zoom level" - zoom level at startup

BIM page

It includes settings used in the system for the management of 3D BIM models (standard IFC), and in particular the connection account to the BiMserver repository used by CMDBuild.



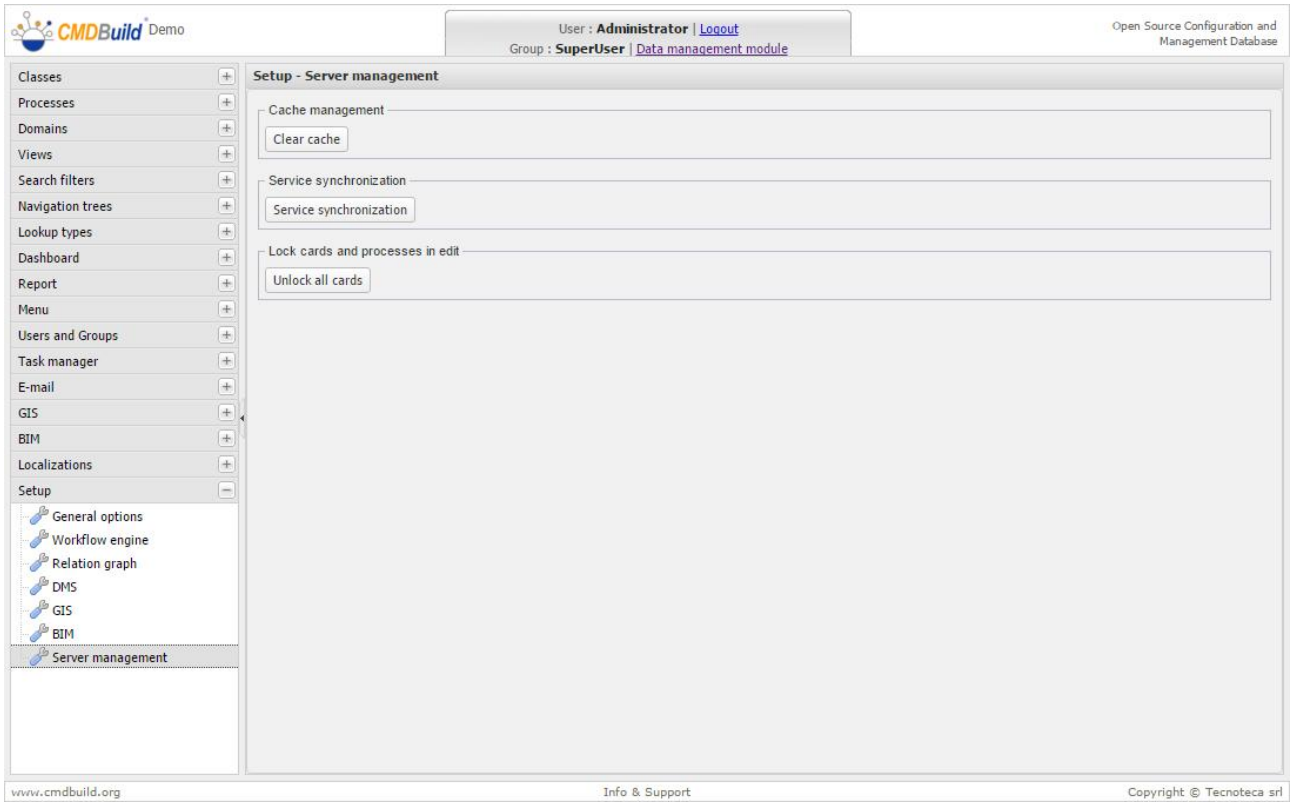
You can set:

- “Enabled” (it requires a configured BimServer instance)
- “URL” of the BiMserver instance
- “Username” of the BiMserver instance
- "Password” of the BiMserver instance

Server management page

Contains two utility functions:

- "Clear cache" - clear CMDBuild cache; useful when elements of the Postgres database are modified outside the application (database views, system functions, etc.) and you don't want to restart Tomcat to refresh
- "Delete inconsistent processes" - this function resets Shark tables; useful when processes have been manually deleted from the database.
- "Unlock all cards", to unlock the locked cards. CMDBuild allows the operator to use the card in edit (this is useful to anticipate the lock release in case an operator with an open card in edit breaks the operation)



APPENDIX: CQL language

CMDBuild provides the possibility of defining some filter conditions evaluated by the system, in order to give the user less choice in some specific cases provided by the application.

Such filter conditions should be expressed with CQL language (CMDBuild Query Language) and can be used to:

- filter some cards in the selection list presented in order to set a Reference attribute
- filter some cards in the “linkCards” widget
- set some selections in the “manageEmail” widget (addresses, authorization to send e-mails, filtration of some system parameters)

How to use CMDBuild

Hereunder you will find the three main directions for use, comparing the SQL syntax with the CQL one.

Simple filter

The simple filter allows to filter on a constant value:

SQL:

SELECT * FROM “ClassName” WHERE “Attribute”=Value

CQL:

from ClassName where Attribute=Value

EXAMPLE:

Select the e-mail from the group “GroupName” in the extended attribute manageEmail.

In the definition of extended attribute the parameter ToAddresses is defined as follows:

ToAddresses={cql:QueryCQLname.Email}

so you can define the following CQL query:

QueryCQLname=”select Email from Role where Code='GroupName'”

Simple filter with server-side variable

This mode allows to perform the filter on a server-side variable:

SQL:

SELECT * FROM “ClassName” WHERE “Attribute”=ServerVariable

CQL:

from ClassName where Attribute={server:VariableName}

NOTES:

In order to identify the variable VariableName (which can be displayed in the form) we used the simple form:

{server:VariableName}

For attributes such LookUp or Reference you have to specify, with the bullet list, whether you want the Id or the Description:

{server:VariableName.Id}

{server:VariableName.Description}

EXAMPLE:

Select the e-mail of the selected user through the attribute “Requester”, reference type, in the extended attribute manageEmail.

In the definition of extended attribute the parameter ToAddresses is defined as follows:

ToAddresses={cql:QueryCQLname.Email}

so you can define the following CQL query:

nomeQueryCQL="select Email from Employees where Id={server:Requester.Id} & Status='A'"

Simple filter with client-side variable

This mode allows to filter on a client-side variable, i.e. completed on the form but not sent to the server.

As you can see, if the field in the form is not set, the simple query with client-side variable reports an error since the value of the side-client variable has not been already defined.

SQL:

SELECT * FROM “ClassName” WHERE “Attribute”=ClientVariable

CQL:

from ClassName where Attribute={client:VariableName}

NOTES:

In order to identify the variable VariableName (which can be displayed in the form) we use the simple form:

{client:VariableName}

For attributes such LookUp or Reference you have to specify, with the bullet list, whether you want the Id or the Description:

{client:VariableName.Id}

{client:VariableName.Description}

EXAMPLE:

The most common example is similar to the one of the server-side variable (replacing client to the server).

Examples of usage

Here below there are some further examples of CQL use.

It filters the floors of the selected building (filter on Reference)

CQL:

```
from Floor where [BuildingFloor].objects(Id=0{client.Building.Id})
```

NOTES:

[BuildingFloor] is the domain between the class Building and the class Floor (type 1:N).

It filters the active processes (filter on Reference)

CQL:

```
from workOrder where FlowStatus='Started'
```

It recovers the e-mail of the user that starts a Ticket (server-side)

CQL:

```
ToAddresses={cql:queryRequester.Email}
queryRequester="select Email from Employees where Id={server:Requester.Id} & Status='A'"
```

NOTES:

[Requester] is the Reference type attribute of the Ticketing workflow which contains the indication of who started the Ticket.

It recovers the e-mail of the user that starts a Ticket (client-side)

CQL:

```
ToAddresses={cql:queryRequester.Email}
queryRequester="select Email from Employees where Id={client:Requester.Id} & Status='A'"
```

NOTES:

[Requester] is the Reference type attribute of the Ticketing workflow which contains the indication of who started the Ticket.

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 (or the DMS Alfresco until the version 3 through its native webservice).

The management of the attachments supports the versioning of those files that have been uploaded a few times, with automatic numbering.

WORKFLOW STEP

"Activity" means one of the steps of which the process consists.

An activity has a name, an executor, a type, possible attributes and methods with statements (CMDBuild API) to be executed.

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.

CMDBuild allows you to create new attributes (in classes and domains) or edit existing ones.

For example, in "supplier" class the attributes are: name, address, phone number, etc..

Each attribute corresponds, in the Management Module, to a form field and to a column in the database.

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).

See also: GIS

CI

We define CI (Configuration Item) each item that provides IT service to the user and has a sufficient detail level for its technical management.

CI examples include: server, workstation, software, operating system, printer, etc.

See also: Configuration

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 computer, a software, a service provider, etc.

CMDBuild allows the administrator - with the Administration Module - to define new classes or delete / edit existing ones.

Classes are represented by cards and, in the database, by tables automatically created at the definition time.

See also: Card, Attribute

CONFIGURATION

The configuration management process is designed to keep updated and available to other processes the items (CI) information, their relations and their history.

It is one of the major ITIL processes managed by the application.

See also: CI, ITIL

DASHBOARD

In CMDBuild, a dashboard corresponds to a collection of different charts, in this way you can immediately hold in evidence some key parameters (KPI) related to a particular management aspect of the IT 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.

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 (by the same operator or by operators of other user groups, which get the permission to use them by the system Administrator)

See also: Class, View

GIS

A GIS is a system able to produce, manage and analyse 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 (cable lines) and polygons (floors, rooms, etc.).

See also: BIM

GUI FRAMEWORK

It is a user interface you can completely customise. It is advised to supply a simplified access to the application. It can be issued onto any webportals and can be used with CMDBuild through the standard REST webservice.

See also: Mobile, Webservice

ITIL

"Best practices" system that established a "standard de facto"; it is a nonproprietary system for the management of IT services, following a process-oriented schema (Information Technology Infrastructure Library).

ITIL processes include: Service Support, Incident Management, Problem Management, Change Management, Configuration Management and Release 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).

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.

With the Administration Module it is possible to define new "LookUp" tables according to organization needs.

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: GUI Framework, Webservice

PROCESS

The term "process" (or workflow) refers to a sequence of steps that realize an action.

Each process will take place on specific assets and will be performed by specific users.

A process is activated by starting a new process (filling related form) and ends when the last

workflow step is executed.

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 database.

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.

CMDBuild users run reports by using the Management Module; reports definitions are stored in the database.

See also: Class, Domain, Database

CARD

The term "card" refers to an element stored in a class.

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 to define attributes shared between classes. From the abstract class you can derive real classes that contain data and include both shared attributes (specified in the superclass) and specific subclass attributes.

For example, you can define the superclass "Computer" with some basic attributes (RAM, HD, etc.) and then define derived subclasses "Desktop", "Notebook", "Server", each one with some specific attributes.

See also: Class, Attribute

ATTRIBUTE TYPE

Each attribute has a data type that represents attribute information and management.

The attribute type is defined using the Administration Module and can be modified within some limitations, depending on the data already stored in the system.

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 not only includes the whole content of a CMDB class, it is a group of cards defined in a logical way.

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.

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 insert additional information, and then display the output of the selected function.