Version **2.5** 



# » Workflow Manual

November 2017 Author Tecnoteca srl www.tecnoteca.com



www.cmdbuild.org

No part of this document may be reproduced, in whole or in part, without the express written permission of Tecnoteca s.r.l.

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.



CMDBuild ® is released under AGPL open source license (http://www.gnu.org/licenses/agpl-3.0.html)

CMDBuild ® is a registered trademark of Tecnoteca Srl.

Everytime the CMDBuild® logo is used, the official maintainer "Tecnoteca srl" must be mentioned; in addition, there must be a link to the official website:

http://www.cmdbuild.org.

CMDBuild ® logo:

- cannot be modified (color, proportion, shape, font) in any way, and cannot be integrated into other logos
- cannot be used as a corporate logo, nor the company that uses it may appear as author / owner / maintainer of the project
- cannot be removed from the application, and in particular from the header at the top of each page

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

# Contents

Introduction	4
CMDBuild modules	4
Available documentation	5
Description of the workflow system	6
General Information	
Purposes	6
Used tools	
Terminology	
Refactoring 2.0	
Implementation method	
Workflows as special classes	
Building the workflow Defining a new process	
Initiation and progress of a process	
Interaction of the workflow with external tools	
General Information Start of a process from an intranet portal via CMDBuild GUI Framework	
Example of configuration of a new process	
General Information Description of the RfC process used as example	
Phase 1 – Items creation in CMDBuild	
Phase 2 – Configuration of the flow with TWE.	
Phase 3 – Importation of the XPDL file in CMDBuild	26
Phase 4 – Implementation of the process from CMDBuild	27
Widgets prompted to use in the user activities of the workflow	
Widget list	
API prompted to use in the automatic activities of the workflow	44
General Information	
Key words	
Management of CMDBuild items	44
Access methods to CMDBuild.	
Methods for types conversion	
Appendix: Documentation to use TWS 2.3	
Foreword	
Automatic methods used in the workflow	
Template automatic methods usable in the workflow	
APPENDIX: Glossary	69

# 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 describes the workflow process included in the CMDBuild application, through which you can configure (Administration module) and run (Management Module) processes for the management of collaborative activities.

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

- system overview ("Overview Manual")
- system administration ("Administrator Manual")
- system usage ("User Manual")
- installation and system management ("Technical Manual")
- webservice details and configuration ("Webservice Manual")
- connectors to sync data through external systems ("ConnectorsManual")

# Description of the workflow system

# **General Information**

In order to support ITIL methodological indications, CMDBuild is able not only to manage the update of the asset inventory and their functional relations, but also to enable the definition and control of the processes for IT service management.

A process includes an activity sequence, carried out by operators and/or computer applications, every application represents an operation that has to be carried out within the process, related, in this case, to the IT asset management with quality criteria.

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.

In the first part of this document you wil find general concepts and basic mechanisms implemented in the system with the CMDBuild 2.0 refactoring.

In the second part there is a sample of simplified workflow, described in its configuration steps.

In the third part, you will find the technical tools available for the configuration of a workflow: widgets definition, description of API functions which can be used in the scripts for the definition of automatisms, performed in the workflow.

In appendix there is the specified technical documentation of the workflow system used until CMDBuild 1.5, whose compatibility is maintained also in CMDBuild 2.0; it will be discarded as soon as possible.

# Purposes

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

ITIL processes, which can be configured in CMDBuild, include: Incident Management, Problem Management, Change Management, Configuration Management, Service Catalogue Management, etc.

Other workflow types concern asset movement, entry of new staff, activation of new work projects, ect.

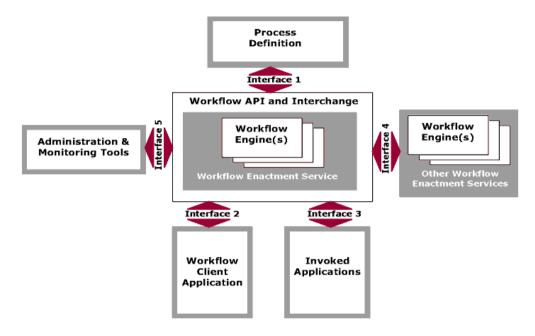
# Used tools

The application chosen for the workflow management uses the following tools:

- XPDL 2.0 (<u>http://www.wfmc.org/xpdl.html</u>) as definition language (standardized from WfMC, WorkFlow Management Coalition according to the model as follows)
- open source TWS Together Workflow Server 4.4 engine (<u>http://www.together.at/prod/workflow/tws</u>), an
  extensible framework for a complete and standard implementation of the specific WfMC
  (http://www.wfmc.org/) and OMG, using XPDL as a native language
- the graphical editor TWE Together Workflow Editor 4.4 (<u>http://www.together.at/prod/workflow/twe</u>) for the workflow design and for the definition of

#### integration mechanisms with CMDBuild

The following schema shows the workflow management according to the model standardized with the WfMC.



# Terminology

The following "vocabulary" includes the following terms:

- process: sequence of steps that realize an action
- activity: 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 related to a special class defined by the Administration Module under the heading "Processes"; the class includes all attributes of the scheduled activities
- each "process instance" corresponds to a card of the "process" class (current activity), combined with the list of its versions (ended activities)
- each activity instance corresponds to a card of the "process" class (current activity) or to a historicized version (ended activity)

Each process has a name, one or more participants, some variables and a sequence of activities and transitions which carry out it.

The process status can be:

- "active", i.e. it is still in an intermediate activity
- "complete", i.e. it ends its activity
- "aborted", i.e. unnaturally terminated
- "suspeded", i.e. maintained only for retrocompatibility with workflow system until CMDBuild

#### 1.5

Each activity can be distinguished by:

- a name
- a performer, which necessarily corresponds to a "user group" and optionally to an operator
- a type: process start, process ending, activity performed by an operator, activity automatically carried out by the system
- any attributes coming from CMDBuild or inside the workflow, which will be set during its implementation
- any widgets (visual controls of some predefined typologies) that will be activated during its implementation
- a script (in the Beanshell, Groovy or Javascript languages), provided in the automatic activities, through which the operations between an user activity and the following can be carried out

# Refactoring 2.0

With the 2.0 release we revised the workflow system, with upgrade to Together Workflow Server 4.4, 2.0 XPDL standard adoption, full support in CMDBuild to the native parallelism in the flow and important performance improvements.

In order to simplify the writing we decided to provide a different definition modality of the automatic activities, supporting the scriptwriting and excluding the use of "tools", available in the previous version of CMDBuild.

The scripts can be written in the BeanShell, Groovy or Javascript language and can use API functions provided for the definition of automatisms (manipulation of process variables, card creation and relations in CMDB, e-mail sending, report creation, etc).

The adoption of the new workflow system implies the lost of the retrocompatibility with workflows developed up to the present day.

In order to grant longer period for the migration of the old workflows to the new adopted solutions we decided to maintain in CMDBuild 2.0 the possibility to work - alternatively - both with Together Workflow Server 2.3 (the version used until CMDBuild 1.5, based on XPDL 1.0) and with the new version Together Workflow Server 4.4 (based on XPDL 2.0).

Then the adopted solution allows:

- new CMDBuild users to work with the new Together Workflow Server 4.4 and with the new functionalities developed in the version 2.0 (native parallelism, automatisms configured through scripts)
- old users to split the migration into two steps:
  - 1. to activate the 2.0 version immediately to make use of the new dashboards and other implemented functionalities, maintaining Together Workflow Server 2.3 active (with improved performances)
  - 2. to commute to Together Workflow Server 4.4 just after the test of the new environment on the test instance, when the conversion tool is available.

The support tool to the workflows migration developed with the previous CMDBuild versions will be then released.

It is advisable to migrate in a short time, since the double CMDBuild compatibility with Together

Workflow Server 2.3 (XPDL 1.0) and Together Workflow Server 4.4 (XPDL 2.0) will be maintained for a limited period.

# Implementation method

#### Workflows as special classes

The mechanisms for the workflow management are implemented in CMDBuild through concepts and procedures entirely consistent with the mechanisms already in the system for the management of the cards.

The workflow management includes:

- "special" Process classes; each corresponds to a type of workflow
- attributes, corresponding to the information presented (for read or write) in the forms which manage the realization of each single step of the process
- relations with other process instances or standard cards involved in the process
- users' groups, that will be able to perform every activity, coinciding with CMDBuild users' groups
- special tools for customizing the behavior of the workflow (widgets and scripts written with proper APIs)

Within the same homogeneity criteria between "normal" and "process" classes, we adopted the following technical tactics:

- the new "confidential" superclass called "Activity" contains some attributes shared with specific workflows, whose workflows are underclasses
- the "history" mechanism was used to draw the progress reports of a process
- the "relations" mechanism has been kept to create automatic or manual links between a card and a process instance, or between two process instances

## Building the workflow

The tools usable through the workflow visual editor are of utmost importance in enabling the design of complex processes, and include:

- the choice of those attributes which can be placed on each form corresponding to a user activity
- the choice of those widgets (visual controls) which can be placed on each form corresponding to a user activity (viewing or creating or editing cards, viewing or creating relations, single or multiple selection of cards, upload of the attached files, implementation of reports)
- flow-control mechanisms, among them parallel activities and subprocesses
- scripting language (BeanShell, Groovy or Javascript) for the definition of those automatisms which must be carried out between a user activity and the following
- API functions which can be called in the scripts

If you are interested in the documentation of further mechanisms used in the workflows, developed for CMDBuild versions until 1.5 (and supported in CMDBuild 2.0 if you use Together Workflow Server 2.3), see the documentation in Appendix (dedicated to the presentation of basic tools and to the mechanism which defines the custom tool through proper templates).

## Defining a new process

To create a new "Process" class, you should follow the next logic sequence of passages:

- analysis of the new process which has to be implemented, in order to single out:
  - list of the users' groups involved in the process
  - workflow: user activities, automatic activities, transition conditions, etc
  - descriptive attributes of the process in its user activities and the related typologies (strings, integer, etc) and the presentation mode (read-only, reading and writing, possible compulsoriness)
  - predefined lists of values required for the creation of "Lookup" attributes
  - domains required to deal correlations between the new process and other classes or other pre-existing processes (which might also be used to create the "Reference" attributes)
  - widgets to configure in every user activity
  - widgets to configure in every automatic process activity
- creation of the new process class, which will be defined in the "Processes" section of the CMDBuild Administration Module, complete of:
  - specific attributes identified in the previous step
  - domains identified in the previous step
- creation of missing users' groups, that should be added through the Administration Module
- through the Administration Module (from the "XPDL" TAB available for each "Process" class) export of the new process template, which includes:
  - process name
  - list of process attributes, that will be placed in the various user activities
  - list of "actors" (users) that interact with the process (the "System" role is automatically created to position system activities)
- design of detail flow of the workflow using the TWE external editor, which will help the completion of the template exported by CMDBuild
- save, using the special functions of TWE external editor of the XML file (to be exact XPDL 2.0) corresponding to the designed process
- import of the process schema in CMDBuild, using the special "XPDL" TAB, available under the heading "Processes" in the Administration Module

Once concluded the operations described above, the new process can be used in the Management Module, (Menu "Processes" or headings like "process" in the Navigation Menu), thus the process can be executed using the workflow engine Together Workflow Server 4.4.

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

#### Implementation method

#### Workflow Manual

CMDBuild Dem	0		User : Adminis Group : SuperUser   D	s <b>trator</b>   <u>Logout</u> ata management m	odule	Open Source Configuration and Management Database
Classes	+	Processes - Request for o	change			
Processes	-	🔘 Add process  🔒 Print s	schema 🔻			
Request for change		Properties Attributes	5 Domains Task manager			
			move process 🚔 Print process 🕆			
			nove process i jai mint process			
		Base properties				Ê
		Name:	RequestForChange			
		Description:	Request for change			
		Inherits from:	Attività	~		
		Superclass:				
		Active:	1			
		User stoppable:				
Dom ains	+					
Views	+			Save	Cancel	
Search fiters	+	Upload XPDL				
Navigation trees	+	XPDL file:	-	Browse	Upload	
nangaaanacee		XPDI THE:				
Lookup types	+	XPDL file:			,	
	+++++++++++++++++++++++++++++++++++++++	Download XPDL template			,	
Lookup types	+		1	~		
Lookup types Dashboard Report Menu	+	Download XPDL template				
Lookup types Dashboard Report	+	Download XPDL template				
Lookup types Dashboard Report Menu	+	Download XPDL template				
Lookup types Dæhboard Report Menu Users and Groups	+	Download XPDL template				
Lookup types Dashboard Report Menu Users and Groups Task manager	+	Download XPDL template				
Lookup types Dashboard Report Menu Users and Groups Task manager E-mail	+	Download XPDL template				
Lookup types Dashboard Report Menu Users and Groups Task manager E-mail GIS	+ + + + + + + + + +	Download XPDL template				

## Initiation and progress of a process

In the Management Module, CMDBuild can perform, through the support of the TWS Together Workflow Server engine, the processes designed with TWE Together Workflow Editor and then imported through the Administration Module.

In order to keep the greatest coherence with the CMDBuild functionalities, dedicated to the management of the items cards in the system, the user interface of the Management Module was designed so that it is consistent with the management of the normal data "classes":

- there is a special menu item "Processes" consistent with the "data sheets" (otherwise "process" elements can be inserted in the Navigation menu with "data sheets" elements or reports and dashboards)
- the process management draws on the standard managements which already exist for the normal cards: "List", "Card", "Details" "Notes", "Relations", "History", "Attachments"
- in the "List" TAB of a specific process, the user can see the activities instances, in which he/she is involved (since he/she attends that activity or previous activities of that process) with:
  - filters by status (started, completed, suspended)
  - data area with tabular display of the information (process name, activity name, request description, process status and further attributes defined as "display base" in the Administration Module), which you can click on in order to access to the management card of that activity
  - possible evidences of parallel activities for that process instance
  - buttons to create a new activity or to make that choice

- in the "Card" TAB you can visualize or fill in the attributes provided for that process activity instance (write or read-only access can be set up through the TWE editor) otherwise you can carry out further operations through the proper widgets (visual controls) configured with the TWE editor
- in the "Notes" TAB you can visualize or insert notes about the activity instance
- in the "Relations" TAB you can visualize or insert relations between the activity instance and the instances of other classes ("cards")
- in the "History" TAB you can visualize the previous versions of that activity instance (instances already carried out)

The list of activities is displayed high up in the following exemplifying form, while you can carry out an activity filling in the card at the bottom.

CMDBuild Demo			ſ			dministrator   L rUser   <u>Administr</u>			Open 1	Source Configuration and Management Database
Navigation		List - Request for c	hange							= - <b>-</b> e
🗷 🛄 Dashboard		🔘 Start Request for	change	Open		~				
Basic archives		Request number			Start date		Status	Category	Final result	Requester
Purchases     Locations		± 0			18/03/2016 22	2:31:01	Analysis requested	Create new ERP u		Wilson Barbara
Assets		1			18/03/2016 22	2:32:12	Registered			Davis Michael
Asset		<b>□</b> 2			18/03/2016 22	2:33:56	Analysis requested	External software i	L	Miler Linda
Computer		Specialist: RFC	cost ana	alysis						
PC		Specialist: RFC								
Notebook		Specialist: RFC								
Server		M Page 1	of 1		2	Q	≚ 🏙 Search filter 📸	Clear filter 🛛 🔒 Print	•	1-3of3
Monitor		Activity Note	Relati	ions His	story E-ma	il Attachme	nts			
Printer		CEdit activity 🗙		i.	million and a second			1 Court	alist RFC risk analysis	ę
NetworkDevice	0	Edit activity		ocess pub	Kelaton graph			Speck	aist RPC risk analysis	9
Rack				r: Miler Lin					*	Attachments
UPS		Re	queste	r: Miler Lin	ua					
License		Des	cription	n: I need ti	he new version	of Autodesk A	utoCAD			1
🗷 🧾 Report				Theed o			all cho			8
G Work flow		C	ategon	: External	software insta	lation				
Request for change			2.3	5						
			Priority							
		* Risk analysi	s resul	t B I	U Font Siz	ze 🔹 🔺	▝╩▝▐Ë 著 著 ■		🖻 🛍 🝼 📿 🛽	
Classes	-			Makes	sure there is er	nough memory	RAM			
	++++									
Pro cesses Views										
Dashboard									-	
Report	+ + +	4	_			111			•	
Utilty	+				Sa	ve Adva	ance Cancel			
www.cmdbuild.org	Said I					Info & Support			Con	yright © Tecnoteca s

Since workflows are peculiar classes, you can find the control buttons also in the upper right of the workflow management form to display full screen the upper or lower side of the form.

# Interaction of the workflow with external tools

## **General Information**

In some cases it may be required that a process (like a new request of HelpDesk) is started by a user who is not expert enough at using CMDBuild (such as the user of the item or of the IT service).

This can be solved using the CMDBuild GUI Framework, described as follows.

## Start of a process from an intranet portal via CMDBuild GUI Framework

The GUI Framework is a javascript / Jquery development environment, used to implement a simplified user interface. Thanks to it, non-expert users can interact with the CMDBuild application.

The GUI Framework includes the following main features:

- it can be activated in portals based on different technologies
- it allows an (almost) unlimited freedom when projecting the graphic layout, defined through an XML descriptor and with the possibility of intervening on the CSS
- it grants a quick configuration thanks to predefined functions (communication, authentication logics, etc.) and to native graphic solutions (forms, grids, upload buttons and other widgets)
- it adapts to workflow advancement forms designed through the visual editor TWE
- it interacts with CMDBuild through the REST webservice
- it is able to gather data from the database of other applications, allowing the management of mix solutions

An sample of implementation based on the CMDBuild GUI Framework is the Self-Service portal, which is part of the preconfigured version CMDBuild READY2USE.

The CMDBuild Ready2Use Self-Service portal allows non-technician users to interact with the IT employees in order to point out their needs and to keep up to date on the resolution activities.

Every user can access to the portal upon local authentication or authentication connected to the Active Directory repository of the company.

The home page of the portal includes:

- a complete menu, on the left
- a quick access to the main features, up on the right
- the most recent IT news
- the advancement situation of the last forwarded requests

CMDBuild suggests an implementation of the functioning portal as portlet in the open source Liferay portal. The used CMDBuild GUI Framework can be activated on portals based on different technologies, since it is developed in javascript / JQuery environment. So, you can ask for custom implementations of the self-service portal, which can work on various portals.

The CMDBuild Ready2Use Self-Service portal implements the following features:

- publication of IT news
- request for technical information

- opening of a damage notice
- request for an IT service, selected from the services catalogue
- · consultation of the advancement of your own requests
- approval of authorization requests
- FAQ
- summary of received notification emails
- connected user profile
- · list of assets and services assigned to the connected user
- useful links

🔾 Add 🔻 🥜 Manage 🔻 📔 Edit G	controls			🕈 Go to 🔻 🛛 👤 admin admin (Sign Out)
	IT Self	-Service Portal		
: Navigation	🀚 Ask a question		•	o
> Home	Requester Anders	on Aaron		
> IT News	Request type Information			
> Ask a question	Area *		•	
> Submit an incident	Short description *			
> Submit a service request				
> My requests				
> Pending approvals				
> Email notifications	Extended description			
Knowledge Base				
My profile				
> My items / services				
> Useful links				
You are now logged into the IT Self Service Portal. Please select an option from the memu. If you have any problems using this support system, please email support(Brygfirm, com The ICT Helpdesk is open from 8.30 am to 18.30 pm Monday to Friday.		Send Cancel	I	
vww.cmdbuild.org - Copyright © Tecnoteca srl				Powered By Liferay

The CMDBuild GUI Framework is not the only one possible option.

You can even implement external web interfaces from the beginning, using your favourite development language and interacting with CMDBuild through its REST and SOAP webservices.

But this solution is less efficient if compared to the re-use of the available GUI Framework.

# Example of configuration of a new process

## **General Information**

The chosen process that describes the various passages necessary for its configuration is a simplified Request for Change (or RfC) process.

It is an extremely simplified process, modelled only for educational purposes; it was suggested for configuration modalities, not for a real use in the production.

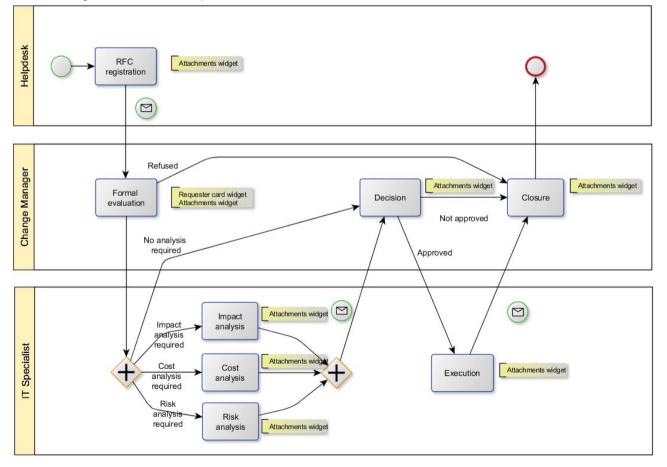
The sample process, complete with the definition in CMDBuild and the XPDL flow designed with TWE, is available on the demo database supplied with CMDBuild.

# Description of the RfC process used as example

The actors of the process are the users' groups:

- · Helpdesk, which carries out the initial registration of the request received by a user
- Change Manager, responsible for the changes to the IT assets of the company
- IT expert, involved for the production of analysis documents and for the change execution

Here's a logic schema of the process:



The process includes the following operations:

- RfC recording
- evaluation of the request's formal aspects:
  - direct closing if the RfC is not acceptable
  - shift to the decisional step, if analysis activities are not requested
  - execution request of one or more analysis typologies, among impact, cost, risk analysis
- execution of the requested analysis typologies (impact, cost, risk analysis)
- decision by the Change Manager, which might be closed if the RfC is not approved
- RfC execution by an IT expert, if the RfC is approved
- final closing

## Phase 1 – Items creation in CMDBuild

Through the Administration Module, under the heading Menu Processes, the process RequestForChange is created in order to manage the workflow:

CMDBuild Demo			User : Administrator   Logout Group : SuperUser   Data management module	Open Source Configuration and Management Database
Classes	+	Processes - Request for d	nange	
Processes	-	🔘 Add process  🔒 Print s	chema 🔹	
Request for change		Properties Attributes	Domains Task manager	
		Modify process X Ren	ove process 🚔 Print process 🔹	
		Base properties		
		Name:	RequestForChange	
		Description:	Request for change	
		Inherits from:	Attività 👻	
		Superclass:		
		Active:		
		User stoppable:		
Dom ains	+	1	Save Cancel	
Views	+	1	Save	
Search fiters	+	Upload XPDL		E
Navigation trees	+	XPDL file:	Browse Upload	
Lookup types Dashboard				
Report	+	Download XPDL template		
Menu	+	Version:	1 Cownload template	
Users and Groups	+			
Task manager	+			
E-mail	+			
GIS	+			
BIM	+			
Localizations	+++++++++++++++++++++++++++++++++++++++			
Setup	+			

Some attributes provided in the process are Lookup, so they require the preventive definition of the related lists, as you can see in the following screenshots.

#### Lookup RFC category (linked to the "Category" attribute of the process)

Class List  Processes Domains Codup Types Cable color	Add Lookup Type     Property Lookup List	Description Form atting PC External software installation Internet access Modify UP address	Parent description	Active V
Domains	Property Lookup List Odd Lookup Code FPC ISE ARI MIR	Description Form atting PC External software installation Internet access	Parent description	V
Lockup Types Cable color Cable color Cable color Country EmailStatus Employee level Employee qualification	ARI MIR	Description Form atting PC External software installation Internet access	Parent description	V
Cable color Cable color Country EmailStatus Employee level Employee qualification	Code FPC ISE ARI MIR	Description Form atting PC External software installation Internet access	Parent description	V
Cable color Country EmailStatus Employee level Employee qualification	Gode IPC ISE ARI MIR	Form atting PC External software installation Internet access	Parent description	V
Country EmailStatus Employee level Employee qualification	FPC ISE ARI MIR	Form atting PC External software installation Internet access	Parent description	V
EmailStatus Employee level Employee qualification	ISE ARI MIR	External software installation Internet access		
Employee level	ARI MIR	Internet access		
	MIR			
Employee state		Modify IP address		$\checkmark$
	NI EDD			
Employee type	NO_ER	Greate new ERP user		$\nabla$
- ElowStatus	NU_CRM	Greate new CRM user		V
Invoice type	8			
Elicense category	2			
E Monitor type	4			
Paper size	Page 1 of	1 > > 2		1-6 of 6
Printer type				
Printer usage	🥒 🖉 Modify Lookup 🛛 💢 Disa	able Lookup		
RAID	Transfer .			
RFC category	Code:			
E RFC decision	Description:			
E RFC final result	Parent description:			
RFC formal evaluation				=
Dashboard +				
Report +	te l			
Menu +				
Users and Groups +	e			
GIS +		(12)		Ŧ
Setup +	E		Save Cancel	

Lookup RFC decision (linked to the "Decision" attribute of the process)

CMDBuild Demo				istrator   <u>Logout</u> Data management module	Open Source Configuration and Management Database
dass List	+	Lookup Management			
Processes	+	Add Lookup Type			
Domains	+	Property Lookup List			
Lookup Types	-				
Cable color	-	Add Lookup			
Country		Code	Description	Parent description	Active
Email Status		APPROVED	Approved		V
Employee level	100	NOT_APPROVED	Not approved		
Employee qualification Employee state Employee state Employee type FlowS tatus Invoice type Incices category Monitor type	н				
Network device type Paper size		A Page 1 of	1 > > 2		1 - 2 of 2
Printer type Printer usage		🥜 Modify Lookup 🛛 💥 Dise	able Lookup		
RAID		Code:			•
E RFC decision		Description:			
RFC final result	*	Parent description:		~	
Dashboard	+	Notes:			-
Report	+				
Menu	+				
Users and Groups	+				
GIS	+	Activo	100		*
Setup	+			Save Cancel	
www.cmdbuild.org			Cred	ts	Copyright © Tecnoteca srl

#### Lookup RFC final result (linked to the "FinalResult" attribute of the process)

CMDBuild Demo			User : Administrator   <u>Logout</u> Group : SuperUser   <u>Data management mode</u>	ile	Open Source Configuration and Management Database
dass List	+	Lookup Management			
Processes	+	Add Lookup Type			
Domains	+	Property Lookup List			
Lookup Types	-				
Table color	-	Add Lookup			
- E Country		Code	Description	Parent description	Active
Email Status		POSITIVE	Positive		
Employee level	1.57	NEGATIVE	Negative		
<ul> <li>Employee qualification</li> <li>Employee state</li> <li>Employee type</li> <li>FlowStatus</li> <li>Twoke type</li> <li>License category</li> <li>Monitor type</li> </ul>		(			
Network device type Paper size		14 4 Page 1 of	1 → ⋈ &		1 - 2 of 2
📰 Printer type 📰 Printer usage	_	🖉 Modify Lookup 🛛 💥 Disa	ble Lookup		
RAID RFC category RFC decision		Code: Description:			Â
RFC final result		Parent description:			
RFC formal evaluation	*	Parent description:	×		=
Dashboard	+	Notes:			
Report	+				
Menu	+				
Users and Groups	+				
GIS	+	Activo	120		-
Setup	+		Save	Cancel	
www.cmdbuild.org			Credits		Copyright © Tecnoteca srl

CMDBuild Demo				r : Administrator   <u>Logout</u> erUser   <u>Data m anagement module</u>	Open Source Configuration and Management Database
dass List	+	Lookup Management			
Processes	+	Add Lookup Type			
Domains	+	Property Lookup List			
Lookup Types	-				
FlowStatus	*	O Add Lookup			
Invoice type		Code	Description	Parent description	Active
E License category		ACCEPTED	Accepted		
E Monitor type		REJECTED	Rejected		
E Network device type					
Paper size					
Printer usage					
RAID					
RFC category					
E RFC decision					
RFC final result		N 4 Page 1 of	1 > > 2		1 - 2 of 2
RFC formal evaluation RFC priority					
	11	🥔 Modify Lookup 🛛 💢 Disa	ible Lookup		
RFC status	10		ble Lookup		
ERFC status	11	Modify Lookup 🔀 Disa	ble Lookup		
RFC status Room usage type Screen size Supplier type	Ш		ble Lookup		^
<ul> <li>RFC status</li> <li>Room usage type</li> <li>Screen size</li> <li>Supplier type</li> <li>Technical reference role</li> </ul>	8	Code: Description:	ble Lookup		Î
<ul> <li>RFC status</li> <li>Room usage type</li> <li>Screen size</li> <li>Supplier type</li> <li>Technical reference role</li> <li>Workplace type</li> </ul>	-	Code: Description: Parent description:	ble Lookup		
RFC status     RFC status     Room usage type     Screen size     Supplier type     Technical reference role     Workplace type     Dashboard	+	Code: Description:	ble Lookup		E
RFC status     Room usage type     Screen size     Supplier type     Technical reference role     Workplace type     Dashboard Report	+	Code: Description: Parent description:	ble Lookup	×	
RFC status RFC status Room usage type SCreen size Supplier type Technical reference role Workplace type Dashboard Report Menu	+ + +	Code: Description: Parent description:	ble Lookup	×	н
RFC status     Room usage type     Soreen size     Supplier type     Technical reference role     Workplace type Dashboard Report Menu Users and Groups	+++++++++++++++++++++++++++++++++++++++	Code: Description: Parent description: Notes:		M	Ш
RFC status RFC status Room usage type SCreen size Supplier type Technical reference role Workplace type Dashboard Report Menu	+ + +	Code: Description: Parent description:	ble Lookup	Save Cancel	н

#### Lookup RFC formal evaluation (linked to the "FormalEvaluation" attribute of the process)

#### Lookup RFC priority (linked to the "RFCPriority" attribute of the process)

CMDBuild Demo				User : Administrator   <u>Logout</u> SuperUser   <u>Data m<i>a</i>nagementmodule</u>		Open Source Configuration and Management Database
dass List	+	Lookup Management				
Processes	+	Add Lookup Type				
Domains	+	Property Lookup List				
Lookup Types	-					
FlowS tatus	*	Add Lookup				
Invoice type		Code	Description		Parent description	Active
E License category		н	High			V
- E Monitor type		MID	Medium			
🔄 Network device type		LOW	Low			
E Paper size						
Printer type						
Printer usage						
RFC category						
RFC decision						
RFC final result						
RFC formal evaluation		Page 1 of	1 🕨 🕅 🤁			1 - 3 of 3
E RFC priority		🥔 Modify Lookup  💥 Disa	able Lookup			
E RFC status						
E Room usage type		Code:				<u>^</u>
📰 Supplier type		Description:				
E Technical reference role		Parent description:				
E Workplace type	*			~		=
Dashboard	+	Notes:				
Report	+					
Menu	+					
Users and Groups	+					
GIS	+	Activo	100			•
Setup	+			Save	Cancel	
www.cmdbuild.org				Credits		Copyright © Tecnoteca srl

Lookup RFC status (linked to the "RFCStatus" attribute of the process)

CMDBuild Demo					Administrator   <u>Logout</u> Jser   <u>Data management module</u>		Open Source Configuration and Management Database
dass List	+	Lookup Mana	agement				
Processes	+	O Add Looku	up Type				
Domains	+	Property	Lookup List				
Lookup Types	-						
Ellips de tipe	*	O Add Look	up				
Invoice type		Code		Description		Parent description	Active
E License category		REC_RFC		Registered			V
E Monitor type		REQ_DOC		Documentation requested			V
📃 Network device type		PRE_DOC		Documentation predisposition			V
Paper size		REQ_EXE		Execution requested			
E Printer type		IN_EXE		Implementation			V
Printer usage		OUT_EXE		Performed			V
RAID		CLOSED		Closed			N. C.
RFC category RFC decision RFC final result RFC final result		14 4 Pa	age 1 of	1 → ⋈ &			1 - 7 of 7
E RFC priority	11	Nodify Loo	okup 💢 Disa	ble Lookup			
E Room usage type E Screen size		Code: Description					Â
Technical reference role							
Workplace type	*	Parent des	scription:		~		-
Dashboard	+	Notes:					=
Report	+						
Menu	+						
Users and Groups	÷						
GIS	+	Activo		100			*
Setup	+				Save	Cancel	
www.cmdbuild.org					Credits		Copyright © Tecnoteca srl

The following "domains" are created in order to define in the process the "Requestor" attributes as foreign keys on the class "Employee" and the relations with the Change Manager and IT experts, who respectively assesses and execute the RfC:

CMDBuild Demo					Grou	User : Administ up : SuperUser   <u>Dat</u>	rator   <u>Logout</u> a managementmodule			Open Source Conf Manageme	iguration an ant Databas
dass List	+	Manage proces	ises								
Processes	=	O Add process	🚔 Print Sc	hema 🔹							
Request for change		Properties	Attributes	Domains	XPDL	Scheduling					
		Add Domain	🧷 Modify I	Domain 🔀 [	Delete Dom	rain				🔽 Indu	de inheri te
		Name		nain descriptio		escription direct	Description inverse	Origin	Destination	Cardinality	M
		RFCAnalyst	RFC	Analyst	A	analysed by	Analyse	Request for change	Employee	N:N	C
		RFCExecutor	RFC	Executor	E	executed by	Per form	Request for change	Employee	N: 1	17
		RFCRequester	RFC	Requester	R	lequested by	Requests	Request for change	Employee	N: 1	E
Dmains	-										
	+										
ookup Types	+ +										
ookup Types ashboard	* * + +										
ookup Types Dashboard Report	+ + + +										
odkup Types Jashboard Report Vienu	+ + + + +										
.ookup Types Dashboard Report Henu Jsers and Groups	+ + + + + +										
Domains Lookup Types Dashboard Report Menu Likers and Groups GIS Setup	+ + + + + + +										

At this point the attributes of the process can be created:

CMDBuild	)emo				iser : Administrator   Logo iperUser   <u>Data managem</u>					Open Source Configuration a Management Databa
dass List	+	Man ag e processes								
Processes	=	🕥 Add process 🛛 🚔 P	rint Schema 👻							
Request for change		Properties Attri	outes Domains 1	(PDL Scheduling	3					
		🔾 Add Attribute	Set sorting							Include inherite
		Name	Description	Type	Editor type	Display in list	Unique	Mandatory	Active	Editing mode
		Code	Nome Attività	STRING						Editable
		Description	Description	STRING						Editable
		FlowStatus	Process Status	LOOKUP						Editable
		RequestNumber	Request number	INTEGER						Editable
		StartDate	Start date	TIMESTAMP		<b>V</b>				Editable
		RFCStatus	Status	LOOKUP		<b>V</b>				Editable
		RF@escription	Description	TEXT	HTML	v				Editable
		Category	Category	LOOKUP	111116	<b>v</b>				Editable
		FormalEvaluation	Formal evaluation	LOOKUP						Editable
		ImpactAnalysisRequest	Impact analysis requ	BOOLEAN						Editable
		<sup>1</sup> CostAnalysisRequest	Cost analysis request	BOOLEAN						Editable
		RiskAnalysisRequest	Risk analysis request	BOOLEAN						Editable
		ImpactAnalysisResult	Impact analysis result	TEXT	HTML					Editable
		CostAnalysisResult	Cost analysis result	TEXT	HTML					Editable
		RiskAnalysisResult	Risk analysis result	TEXT	HTML	(m)				Editable
		Decision	Decision	LOOKUP						Editable
		PlannedActions	Planned actions	TEXT	HTML					Editable
		ExecutionStartDate	Execution start date	TIMESTAMP		<b>V</b>				Editable
omains	+	ActionsPerformed	Actions performed	TEXT	HTML					Editable
ookup Types	+	ExecutionEndDate	Execution end date	TIMESTAMP		<b>V</b>				Editable
ashboard	÷	FinalResult	Final result	LOOKUP						Editable
eport	+	EndDate	End date	TIMESTAMP						Editable
enu	+	Requester	Requester	REFERENCE		<b>V</b>				Editable
enu sers and Groups	+	RFCPriority	Priority	LOOKUP						Editable
IS	+	Modify Attribute	Poloto Attribute							
etup	+		Delete Attribute		Sa	ve Cancel				
www.cmdbuild.org					Credits					pyright © Tecnoteca srl

Last thing you can create the users' groups involved in the workflow:

CMDBuild Dem	no		User : Administrator   Logout Group : SuperUser   Data managementmodule	Open Source Configuration and Management Database
Class List	+	Group management		
Processes	+	🙆 Add Group		
Domains	+	Properties Permissions	Users UI configuration	
Lookup Types	+			
Dashboard	+	🥜 Modify Group 🔀 Disable Gr	pup	
Report	+	Group Name:	ChangeManager	
Menu	+	Description:	Change manager	
Users and Groups	-			
😑 🗁 Groups		Email:		
🛃 Change manager		Is administrator:		
A Helpdesk		Starting page at:	▼ ×	
Specialist	(	Active:		
GIS	+			
Setup	+		Save Cancel	
www.cmdbuild.org			Credits	Copyright © Tecnoteca srl

At this point you can export the XPDL schema produced by CMDBuild, used with the visual editor

TWE to design the detail flow of the process itself:

CMDBuild Den	10			Grou	User : Adn p : SuperUser	ninistrator   <u>Loc</u>   <u>Data manage</u>	Open bource coming	
Class List	+	Manage processes						
Processes	-	🔾 Add process 🔒 P	Print Schema 🔹					
Request for change		Properties Attri		ns XPDL	Scheduling			
		Upload XPDL						
		XPDL File:				Browse		
		Sketch:				Browse		
		Download XPDL temp Version:	late		Y		È stato scelto di aprire  È stato scelto di aprire  RequestForChange.xpdl  che è un: XPDL Together Workflow Editor file da: http://localhost.8080  Che cosa deve fare Firefox con questo file?  Aprirlo con Java(TM) Platform SE binary (predefinita)  Salva file	•
Domains	+							
Lookup Types							Da ora in avanti esegui questa azione per tutti i file di questo tipo	D.
Dashboard	+ + + + + + + + + + + + + + + + + + + +							
Report	+						OK Annul	lla
Menu	+							
Users and Groups	+							
GIS	+					0		
Setup	+						Download template	

The XPDL file will include only the general data available at the moment:

- process name
- · list of unreserved process attributes present in the process management class
- list of the roles defined in the system

Those data will be the starting point of the activities carried out through the TWE editor, in which all aspects related to the specific process flow will be enriched.

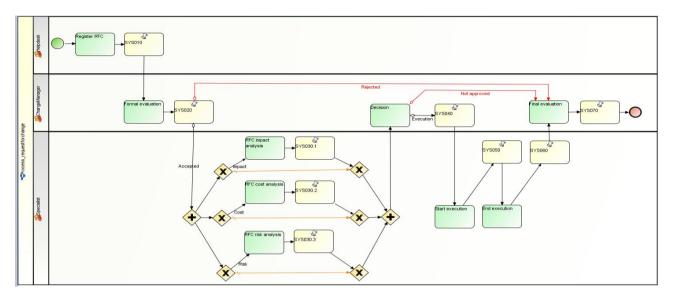
#### Phase 2 – Configuration of the flow with TWE

Through the TWE editor it's possible to perform the following operations:

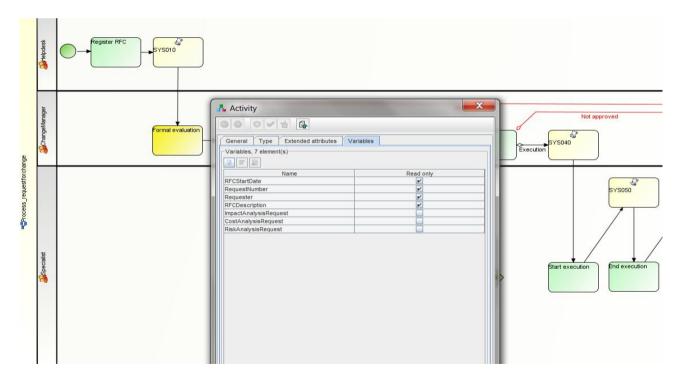
- flow design by placing the activities of the various provided typologies (process starting and ending, user activities, automatic activities, routing activities for the parallelism management) and their connection according to the provided transition typologies
- completion of user activities, specifying what process attributes will be shown in the form related to that activity (by indicating if read-only or read/write) and what widgets will be made available in the same form (by indicating the parameters provided for each one)
- completion of automatic activities, writing the script which implements the automatisms required in that activity (using the API available for that aim)
- completion of transitions among activities, specifying the criteria for the flow to cover a transition or another, when the choice is binding

Here are some descriptive screenshots of the above mentioned activities.

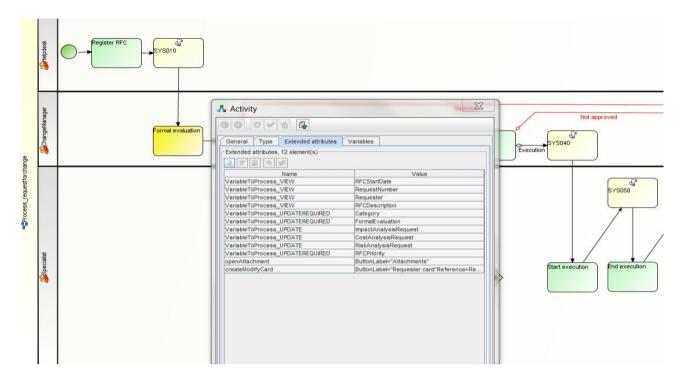
#### General flow design:



<u>User activities - "Variables" TAB</u>, used to choose the attributes which must appear in the form (with possible indication of read-only modality):



<u>User activities - "Extended Attributes" TAB</u>, used to indicate the compulsory attributes ("UPDATEREQUIRED") and to request the input in the form of one or more widgets (in the sample

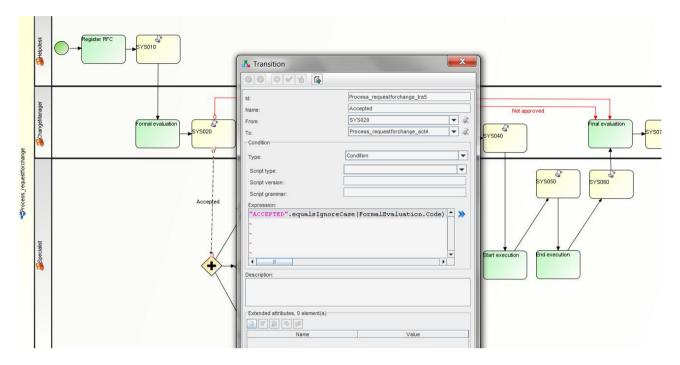


#### openAttachments for the attachments and createModifyCard to consult the requester card)

<u>Automatic activity - "Type" TAB</u>, used to write the script that implements the provided automatisms (in the sample, the activity SYS010 carries out the automatic structuring of the system date, the automatic attribution of a univocal progressive number, the building of a significant description, the structuring of a new state reached by the process).

Register RFC	
🔥 Activity	
60 0 1 1 1 G	
General Type Extended attributes	
Туре	Task script
- Script Script type: Script version: Script grammar. Expression:	text/java 🔽
<pre>RPCStartDate = Calendar.getInstance(); RequestNumber = Integer.parseInt(cmdb.callFunction("cmwf_getRFCNumber").execute().get("RFCNumber") requesterCard = cmdb.cardFrom(Requester); Description = "RFC n. " + RequestNumber.toString() + " created by " + requesterCard.getDescription RFCStatus = cmdb.selectLookupByCode("RFC status", "REC_RFC"); ************************************</pre>	

<u>Transition</u>, used to link two activities, conditionally or not (in the example it is provided a condition related to the formal acceptance of the RfC)



# Phase 3 – Importation of the XPDL file in CMDBuild

When the configuration of the process in TWE is complete, you will load in CMDBuild the related XPDL file.

The process flow can be then modified several times, only exporting the last version from CMDBuild, editing it with TWE and importing it again in CMDBuild. You have to consider that the new version will be used when new processes are started, while each current process will go on with the XPDL version valid when they first started.

CMDBuild Dem	0			Group			r   <u>Logout</u> anagement m odule	Open Source Configuration and Management Database
Class List	+	Manage processo	es					
Processes	=	O Add process	😂 Print Sch	nema 🔹				
Request for change		Properties /	Attributes	Domains	XPDL	Sched	ling	
		Upload XPDL						
		XPDL File:		RequestFor	Chap on yo	al	Result	
				Requestror	change.xp	ui	Browse	
		Sketch:					Browse	
		Download XPDL	template					
		Version:		9		*		
Dom ains	+							
Lookup T ypes	+							
Dashboard	+							
Report	+							
Menu	+							
Users and Groups	+ + + + + +							
GIS	+							
Setup	+						Downbad template	
www.cmdbuild.org					c	redits		Copyright © Tecnoteca srl

#### Phase 4 – Implementation of the process from CMDBuild

The workflow imported in CMDBuid is available to be used by the provided operators groups.

In the example, the management workflow of the RfC will be started by an operator of the Helpdesk group, valued by an operator of the Change Manager group, analysed and carried out by an operator of the IT expert group. You have to consider that the operators of the SuperUser group can "personify" any other group defined in CMDBuild.

From the RfC process management, the RfC are presented as open (or in the state selected on the upper list: open, suspended, complete, aborted, all).

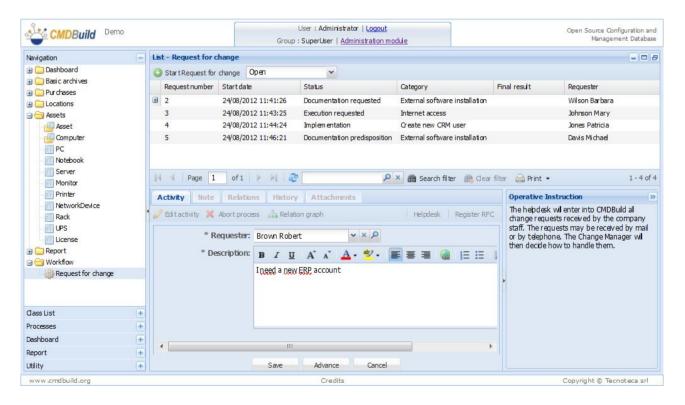
Through the button "Start Request for Change" the Helpdesk can register a new request.

#### Example of configuration of a new process

#### Workflow Manual

CMDBuild Demo			Group	User : Administrator   Logout : SuperUser   Administration mo	idule			Configuration an agement Databas
Navigation -	List - Request fo	r change						
🗄 🧰 Dashboard	Start Request	for change	Open	~				
Basic archives	Request numb	er Start dat	e	Status	Category	Final result	Requester	
) 🚞 Purchases ) 🧰 Locations	± 2	24/08/20	)12 11: <mark>41:2</mark> 6	Documentation requested	External software install	ation	Wilson Barb	ara
Assets	3	24/08/20	12 11:43:25	Execution requested	Internet access		Johnson Ma	y
Asset	4	24/08/20	012 11:44:24	Implem entation	Create new CRM user		Jones Patric	a
Computer	5	24/08/20	012 11:46:21	Documentation predisposition	External software install	ation	Davis Micha	э
PC Notebook								
Server	14 4 Page	1 of 1	IN N R	م ا	× 🏦 Search filter 👘	dear filter 🔒 Print 🔹		1-40
- Monitor Printer	( )(	Law						
NetworkDevice	Activity No	e Relati	ons History	/ Attachments				
Rack	🕈 🥜 Editactivity 🚦	Abort pro	cess 🚠 Relati	on graph			Helpdesk Re	gister RFC
UPS	*	Poquector	: Brown Robe	rt v×P				
License								
) 🧰 Report ) 😋 Workflow	*	Description	BIU	A A 🗛 🖓 🖌 📕		II 😰		
Request for change			I need a new	ERP account				gister RFC
								1
lass List +								
rocesses +			-					
ashboard +								
eport +								
tlity +				Save Adv	vance Cancel			
	J [							

Previous to filling in the form, the operator can refer to the operative instructions associable with every user activity (which can be formulated with TWE, filling in the field "Description" in the "General" TAB of the activity).



#### Workflow Manual

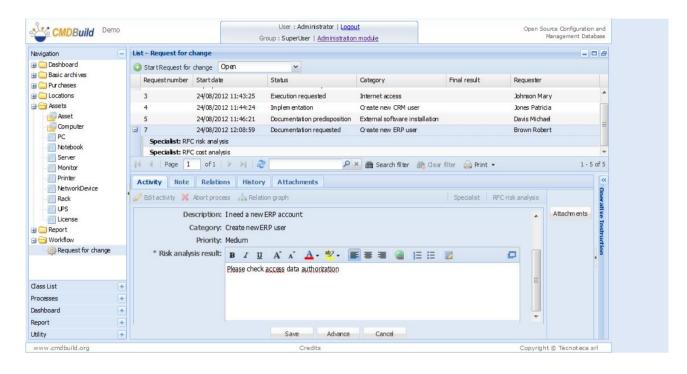
By validating the progress at the following step, the activity is taken by the Change Manager, that - in our simplified example - will fill in the following information:

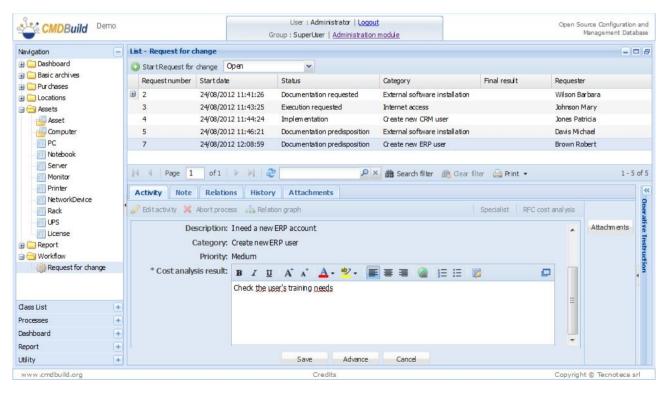
CMDBuild Demo			User : Administrator   L Group : SuperUser   <u>Administr</u>			Ope	n Source Configuration Management Datab	
Navigation	- List - Reques	t for change					_	3 8
🗄 🦲 Dashboard	📀 Start Requ	est for change Open	~					
Basic archives	Request nu	umber Start date	Status	Category	Final result	Requester		
Purchases     Locations	<b>⊞</b> 2	24/08/2012 11:41:	26 Documentation requested	External software installation	1	Wilson Bart	ara	
	3	24/08/2012 11:43:	25 Execution requested	Internet access		Johnson M	ary	
Asset	4	24/08/2012 11:44:	24 Implementation	Create new CRM user		Jones Patri	cia	
- Computer	5	24/08/2012 11:46:	21 Documentation predisposition	External software installation	ı	Davis Micha	el	
PC	7	24/08/2012 12:08:	59 Registered			Brown Rob	ert	
No tebo ok								
Server	- Harris		~					
Monitor Printer	IN A Pag	e 1 of 1 🕨 🕅	P	🎽 🏙 Search filter 🛛 🛗 Clear filt	er 📋 Print 👻		1 - 5 0	of 5
NetworkDevice	Activity	Note Relations Hi	story Attachments					<
Rack	P Edit activit	y 💢 Abort process 🚠	Delating graph		ChangeManager Fo	m al evaluation		8
UPS	Cure ac uvic	y 🙈 Abort process nam	Relation graph		Cialgenaragei ro	inia evaluation		<b>Onerative Instruction</b>
License		Start date: 24/08/20	12 12:08:59				Attachments	ive
🕀 🧰 Report	Red	quest number: 7					Requester card	Ins
🖃 🚍 Work flow		Requester: Brown Ro	hart					1
Request for change								tion
		Description: I need a	and the second					6
		* Category: Create n	ew ERP user 🗸 🗙					4
	* Forn	al evaluation: Accepted	• × ×					
	Impact and	ilysis request: 🦳						
	- Craham	ilysis request: 🔽						
Class List								
Processes		alysis request: 🔽						
Dashboard	+	* Priority: Medium	<b>₩</b> ×					
Report	+							
Utility	+		Save Advar	ce Cancel				

In the example we provide at this step the possible use of the enclosed loading widgets and those for the reference of the complete requester card:

1	CMDBuild Demo	User : Administrator   Logout Group : SuperUser   Administration module		Ope	n Source Configuratio Management Dat	
Navig	Requester card		×		-	08
	🕓 Add card 🝷		<u>*</u>			
	Number: 05			uester		
	Surname: Brown			on Barb		
86	Name: Robert			ison Ma es Patric		
	Type: External consultant			s Micha		
-	Qualification: Clerk			vn Rob		
	Level: Silver					
	Email: robert.brown@gmail.com				1.5	of 5
-	Office: Office 03 - Legal Department					
-	Phone: 65432		=			≪ 0n
	Mobile: 24555556			ition		Det
	Fax				Attachments	ative.
•	State: Suspended				Requester card	Ins
	ource outpended				the designed of the second	truc
						Idion
						4
dass	4	111				
Proce						
Dashb		Save Close				
Report						
Utility	+	Save Advance Cancel				
www	.cmdbuild.org	Credits		Copyri	ight © Tecnoteca s	srl

The Change Manager demanded in our example two typologies of analysis, so the workflow moves to IT experts, that, in parallel (using one of the new functionalities implemented in CMDBuild 2.0), can carry out their analysis (respectively risk and cost analysis) and transfer the results.





The Change Manager currently provides the results of the requested discussions and can take his/her decision.

CMDBuild Demo		G	User : Administrator   <u>Logou</u> roup : SuperUser   <u>Administration</u>	Contract of the second se			urce Configuration and Management Database
Navigation –	List - Request for change						
🗄 🧰 Dashboard	🗿 Start Request for change	Open	~				
Basic archives	Request number Start	date	Status	Category	Final result	Requeste	r.
🛛 🦳 Pur chases 🗑 🦳 Locations		/2012 11:41:26	Documentation requested	External software installat	ion	Wilson Ba	rbara
Assets	3 24/08	/2012 11:43:25	Execution requested	Internet access		Johnson M	1ary
Asset	4 24/08	/2012 11:44:24	Implem entation	Create new CRM user		Jones Pat	ricia
Computer	5 24/08	/2012 11:46:21	Documentation predisposition	External software installat	ion	Davis Mic	hael
PC	7 24/08	/2012 12:08:59	Documentation predisposition	Greate new ERP user		Brown Ro	bert
Monitor NetworkDevice Rack License Workflow Workflow Request for change	Editactivity 💥 Abort p Reques Descripti Catego	ations History process Relation ter: Brown Robert on: Ineed a new pry: Create new El ity: Medium	Attachments on graph t ERP account	🚰 📸 Search filter 🛛 🚵 🛛	ear filter 🔐 Print •	Decision	Attachments
Jass List 🛛 🕂	Cost analysis res		er's training needs				
Processes +	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		access data authorization				
Dashboard 🛛 🛨	Same and	on: Approved	¥ X				
Report +						<b>T</b>	
Jtility +			Save Advance	Cancel			
www.cmdbuild.org			Credits			Convrint	t © Tecnoteca srl

If the decision is positive, according to the flow designed with TWE, the IT experts are asked to carry out the RfC activity. At the beginning the operation makes the request with indication of the activities which must be carried out; at the end it registers the activities already carried out.

CMDBuild Demo			User : Administrator   <u>Logo</u> Group : SuperUser   <u>Administratio</u>	and the second se			urce Configuration and Management Database
Navigation 📃	List - Request for o	change					
🕀 🧰 Dashboard	3 Start Request for	change Open	~				
Basic archives	Request number	Start date	Status	Category	Final result	Requester	r
Contractions	⊞ 2	24/08/2012 11:41:26	Documentation requested	External software installation	1	Wilson Ba	rbara
Assets	3	24/08/2012 11:43:25	Execution requested	Internet access		Johnson N	1ary
Asset	4	24/08/2012 11:44:24	Implem entation	Create new CRM user		Jones Pat	ricia
Computer	5	24/08/2012 11:46:21	Documentation predisposition	External software installation	1	Davis Mid	nael
PC	7	24/08/2012 12:08:59	Execution requested	Create new ERP user		Brown Ro	bert
Server     Monitor     Printer     Rack     UPS     License     Workflow	Costanaly	sis result: Check the u	y Attachments	🗙 🏙 Search filter 🛛 🚓 Gear		rtexecution	1 - 5 of
🙀 Request for change	* Planne	dactions: B I U	Ι Α΄ Α΄ <u>Α</u> · ½· [		2	0	dior
		Create the I	new account with the Administra	ator Jool			1
Jass List 🔶 🕂							
rocesses +						-	
Dashboard +							
Report +							
Jtility +			Save Advance	Cancel			

#### Workflow Manual

#### Example of configuration of a new process

CMDBuild Demo			Gro	User : Administrator   Logou oup : SuperUser   Administration	A STATE OF			urce Configuration and Management Database
Navigation	List - Request for	change						
🕀 🧰 Dashboard	🔘 Start Request for	change Oper	ı	~				
Basic archives     Durchases	Request number	Start date		Status	Category	Final result	Requester	r
	± 2	24/08/2012 1	1:41:26	Documentation requested	External software installation		Wilson Ba	rbara
Assets	3	24/08/2012 1	1:43:25	Execution requested	Internet access		Johnson N	1ary
Asset	4	24/08/2012 1	1:44:24	Implem entation	Greate new CRM user		Jones Pat	ricia
Computer	5	24/08/2012 1	1:46:21	Documentation predisposition	External software installation		Davis Mid	hael
PC	7	24/08/2012 12	2:08:59	Implem entation	Create new ERP user		Brown Ro	bert
Notebook Server								
Monitor Printer	N Page 1	of1 🕨	N 2		🗙 🏦 Search filter 🛛 🛗 Gear	filter 🔐 Print 🔹		1 - 5 of 5
NetworkDevice	Activity Note	Relations	History	Attachments				~
Rack	🥜 Editactivity 💥		and a constraint	2 1		Specialist End e	xecution	Attachments of Task
License								Attachm ents
🕀 🧰 Report		tart date: 24/						Ins
😑 🗁 Workflow	Planne	d actions: Cre	eate the new	vaccount with the Administrat	or Tool			T I
Request for change	* Actions p	erformed: B	IU	A A 🗛 · 🥙 • 🔳				, iii
		Jb.	le new acco	unt has been successfuly, crea	ted			0
dass List								
Processes							TH .	
Dashboard +								
Report +		_						
Utility +				Save Advance	Cancel			
www.cmdbuild.org				Credits			Copyrigh	t © Tecnoteca srl

As last operation, the Change Manager closes the RfC stating a positive result.

CMDBuild Demo		G	User : Administrator   <u>Logou</u> roup : SuperUser   <u>Administration</u>	and the second se		Open Source Configuration and Management Database
Navigation	List - Request for cha	nge				- 0 8
🕀 🧰 Dashboard	Start Request for characteristics	ange Open	~			
🕀 🧰 Basic archives		tart date	Status	Category	Final result	Requester
Pur chases     Locations	± 2 2	4/08/2012 11:41:26	Documentation requested	External software installa	ation	Wilson Barbara
Generations     Generations		4/08/2012 11:43:25	Execution requested	Internet access		Johnson Mary
Asset		4/08/2012 11:44:24	Implem entation	Create new CRM user		Jones Patricia
Computer	5 2	4/08/2012 11:46:21	Documentation predisposition	External software installa	ation	Davis Michael
PC	7 2	4/08/2012 12:08:59	Performed	Create new ERP user		Brown Robert
Notebook						
Server		1				
Monitor	Page 1	of1 🕨 🕅 🧟	۲ ( <b>Q</b> )	🍯 🋗 Search filter 🛛 📇 🖸	Jear filter 🛛 🔒 Print 🔹	1 - 5 of 5
Printer NetworkDevice	Activity Note	Relations History	Attachments			~
Rack					1	8
UPS	Gractivity 🔏 Ab	ort process 👘 Relati	on graph		ChangeManager An	nal evaluation
License	Cat	egory: Create new E	P.D. usor			Attachments
🕀 🧰 Report			KF USEI			Ins
🖃 🔂 Workflow		riority: Medium				
Request for change	Impact analysis					- E
	Cost analysis	result: Check the us	er's training needs			
	and the second second second	result: Please check	access data authorization			
dass List +	- Busine Perin	ormed: The new acco	ount has been successfuly creat	ted		=
Processes +	Execution end	d date: 24/08/2012	12:26:26			
Dashboard 🕴	* Final	result: Postive	¥ X			
Report +						
Utility +			Save Advance	Cancel		
www.cmdbuild.org			Credits			Copyright © Tecnoteca srl

At this stage, the RfC we worked on (number 7) will not appear in the list of the open RfC.

CMDBuild Demo				G	User : Administrator   <u>Log</u> Group : SuperUser   <u>Administratio</u>	and a start of the		Open Source Cont Managem	figuration and ent Database
lavigation -	Lis	t - Request for o	change						
🛛 🧰 Dashboard	0	Start Request for	change	Open	~				
Basic archives		Request number	Start da	ate	Status	Category	Final result	Requester	
) 🚞 Purchases ) 🧰 Locations	Đ	2	24/08/2	012 11:41:26	Documentation requested	External software installa	ton	Wilson Barbara	
		3	24/08/2	012 11:43:25	Execution requested	Internet access		Johnson Mary	
Asset		4	24/08/2	012 11:44:24	Implem entation	Create new CRM user		Jones Patricia	
Computer		5	24/08/2	012 11:46:21	Documentation predisposition	External software installa	ton	Davis Michael	
Monitor Printer NetworkDevice Rack UPS License	•	Ctivity Note	Rela	tions Histor	y Attachments	X 🏙 Search filter 🛛 🟦 C			1 - 4 of
) 🧰 Report									7
Workflow									
Jass List + rocesses + bashboard +									
eport +	1								
eport + tility +					Save	Advance Cancel			

But it can be refered with all its information in the list of the completed RfC (the list can be selected in the upper part of the form)

CMDBuild Demo			User : Administrato Group : SuperUser   <u>Admi</u>				Configuration and agement Database
Navigation	List - Request for	change					8
🕀 🧰 Dashboard	🗿 Start Request for	r change Completed	*				
Basic archives     Durchases	Request number	Start date	Status	Category	Final result	Requester	
Locations	6	24/08/2012 11:48:37	Documentation requested	Not applicable	Negative	Smith James	
🖃 🔂 Assets	7	24/08/2012 12:08:59	Performed	Create new ERP user	Positive	Brown Robert	
Asset							
Computer							
PC No tebook							
Server	14 4 Page 1	of 1 🕨 🕅 🧯	٩ ٧	🗙 🋗 Search filter 🛛 🛗 dear filter	🚔 Print 🔹		1 - 2 of 2
Monitor	Activity Note	Relations Histo	ry Attachments				«
Printer	Landerson and Landerson						
NetworkDevice	🥖 Edit activity 💢	Abort process 🔒 Rel	ation graph				On erative Instruction
Rack	Nome	Attività:					tiv e
License			ated by Brown Robert date Fri A	ug 24 12:08:59 CEST 2012			Ind
🕀 🦲 Report		s Status: Completato		ug 2112.00.00 02012012			ta la
🖃 🔄 Work flow		number: 7					= tion
Request for change			12:00-50				
	50	art date: 24/08/2012 : Status: Performed	12:08:39				
	_						
		scription: Ineed a new					
		ategory: Create new I	BRP user				
dass List +		aluation: Accepted					
Processes +	Impact analysis						
Dashboard +	Cost analysis						
Report +	Risk a nalysis	request: Yes					-
Utility +			Save	Advance Cancel			
www.cmdbuild.org			Credits			Copyright © T	'ecnoteca srl

#### Workflow Manual

CMDBuild Demo			User : Administrator Group : SuperUser   <u>Admin</u>				Configuration and agement Database	
lavigation	List - Request for a	dhange						
🗉 🧰 Dashboard	😳 Start Request for	r change Completed	*					
Basic archives	Request number	Start date	Status	Category	Final result	Requester		
	6	24/08/2012 11:48:37	Documentation requested	Not applicable	Negative	Smith James		
Asse ts	7	24/08/2012 12:08:59	Performed	Create new ERP user	Positive	Brown Robert		
Asset Computer PC No tebook Server	14 4 Page 1	of 1 🕨 🕅 🧔	: ۵	🖌 🃸 Search filter 🛛 📸 Dear filter	🔒 Print 🔹		1 - 2 of	
Monitor	Activity Note	Relations Histo	ry Attachments				10	
Printer								
NetworkDevice Rack	🧭 Edit activity 💢	Edit activity X Abort process and Relation graph Impact analysis result:     Cost analysis result: Check the user's training needs     Risk analysis result: Please check access data authorization     Derektion: Approved						
UPS	Impact analys							
License	Cost analys							
Report	Risk analys	is result: Please check	access data authorization					
🔁 Workflow		Decision: Approved						
Request for change			ew account with the Administrato	r Tool				
		art date: 24/08/2012						
			count has been successfully create	d				
		and date: 24/08/2012		iu iii				
			12.20.20					
ass List		al result: Positive					=	
		ind date: 24/08/2012						
ashboard	ree .	quester: Brown Robe	rt					
	6	Priority: Medium					-	
	E.		Save	Advance Cancel				
1 - 1 <sup>0</sup>								

In addition to the basic information, you can refer to the relations configured with that RfC process instance (Relations TAB).

CMDBuild Demo		Gr	User : Administrato oup : SuperUser   <u>Admir</u>	and the second second	odule				ce Configuration magement Data	
Navigation	- List - Request for char	nge							-	- 8
🕀 🧰 Dashboard	Start Request for characteristics	ange Completed	~							
Basic archives     Durchases	Request number St	tart date	Status		Category		Final result	Requester		
	6 24	4/08/2012 11:48:37	Documentation request	ted I	Not applicable		Negative	Smith Jame	s	
G G Assets	7 24	4/08/2012 12:08:59	Per formed	(	Create new ER	RP user	Positive	Brown Robe	rt	
Asset Computer		Ē	10							
- Notebook	N Page 1	of1 🕨 🕅		2 × Q	Search filt	ter 🛗 Clear filt	er 🚔 Print 🔹		1 - 2	2 of 2
Server	Activity Note	Relations History	Attachments							~
Monitor										8
NetworkDevice	Relation graph			line and						erat
Rack	, dass		Begin date	Code		Description				ive
UPS	Employee	. item)	24/08/2012 12:08:59	05		Brown Robert			\$ <i>0</i> 9	Operative Instruction
Request for change										4
dass List	+									
Processes	+									
Dashboard	÷									
Report	<b>+</b>									
	+									
www.cmdbuild.org			Credits					Copyright (	© Tecnoteca :	srl

You can also refer to the sequence complete with progress activities of the process (History TAB).

List - Request for cl		sup respereder preminerer	on module		Managem	ent Databas	
	- List - Request for change						
3 Start Request for	change Completed	*					
Request number	Start date	Status	Category	Final result	Requester		
6	24/08/2012 11:48:37	Documentation requested	Not applicable	Negative	Smith James		
7	24/08/2012 12:08:59	Performed	Greate new ERP user	Positive	Brown Robert		
14 4 Page 1	of1 🕨 🕅 😂	م	🔀 🃸 Search filter  🏨 (	Jear filter 🔒 Print 🔹		1 - 2 a	
Activity Note	Relations History	Attachments					
	1	1 House and the second se	Attributes	Activity name	Activity performer	•	
	:33	system	1				
			1	Final evaluation	ChangeManager		
			1	End execution	Specialist		
······································	:01 24/08/2012 12:2	5:02 admin	1	Start execution	Specialist		
₫ 24/08/2012 12:19	:47 24/08/2012 12:1	9:47 admin	1	Decision	ChangeManager	=	
······································	:48 24/08/2012 12:1	8:49 admin	1	RFC cost analysis			
☑ 24/08/2012 12:16	:45 24/08/2012 12:10	5:46 admin	1	RFC risk analysis,	Specialist		
₫ 24/08/2012 12:14	18 24/08/2012 12:14	4:20 admin	1	Formal evaluation	ChangeManager		
	CANANA AND AND AND AND AND AND AND AND AN	9:00 admin	×	Register RFC	Helpdesk		
Final result Risk analys	: <b>is request</b> : false						
Formal eva Risk analys	luation: is result:						
	7  Activity Page 1  Activity Note Begin date →  24/08/2012 12:27  24/08/2012 12:26  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  24/08/2012 12:16  Activity Note Comparison of the temperature of the temperature of tempe	Page         of1           Activity         Note         Relations         History           Begin date         End date         End date         24/08/2012 12:27:33           2/4/08/2012 12:27:32         2/4/08/2012 12:27:32         2/4/08/2012 12:27:32         2/4/08/2012 12:22:13           2/4/08/2012 12:27:32         2/4/08/2012 12:22:13         2/4/08/2012 12:25:01         2/4/08/2012 12:22:13           2/4/08/2012 12:25:01         2/4/08/2012 12:22:13         2/4/08/2012 12:22:13         2/4/08/2012 12:22:13           2/4/08/2012 12:15:145         2/4/08/2012 12:12:14         2/4/08/2012 12:12:14           2/4/08/2012 12:12:16:45         2/4/08/2012 12:12:14	7       24/08/2012 12:08:59       Performed         Activity       Note       Relations       History       Attachments         Begin date       End date       User       User         24/08/2012 12:27:33       system       9         24/08/2012 12:27:32       24/08/2012 12:27:33       admin         24/08/2012 12:26:26       24/08/2012 12:26:27       admin         24/08/2012 12:25:01       24/08/2012 12:25:02       admin         24/08/2012 12:19:47       24/08/2012 12:19:47       admin         24/08/2012 12:19:47       24/08/2012 12:19:48       24/08/2012 12:19:40       admin         24/08/2012 12:19:47       24/08/2012 12:19:49       admin       34/08/2012 12:109:00 <td>7       24/08/2012 12:08:59       Performed       Create new ERP user         Note       Page 1       of 1       Page 1       of 1         Activity       Note       Relations       History       Attachments         Begin date       End date       User       Attributes         24/08/2012 12:27:33       system       Image: Comparison of the comp</td> <td>7     24/08/2012 12:08:59     Performed     Create new ERP user     Positive       Note     Page 1     of 1     page 1<!--</td--><td>7       24/08/2012 12:08:59       Performed       Oreate new ERP user       Positive       Brown Robert         Image: Control of the state of the sta</td></td>	7       24/08/2012 12:08:59       Performed       Create new ERP user         Note       Page 1       of 1       Page 1       of 1         Activity       Note       Relations       History       Attachments         Begin date       End date       User       Attributes         24/08/2012 12:27:33       system       Image: Comparison of the comp	7     24/08/2012 12:08:59     Performed     Create new ERP user     Positive       Note     Page 1     of 1     page 1 </td <td>7       24/08/2012 12:08:59       Performed       Oreate new ERP user       Positive       Brown Robert         Image: Control of the state of the sta</td>	7       24/08/2012 12:08:59       Performed       Oreate new ERP user       Positive       Brown Robert         Image: Control of the state of the sta	

Uploading the attachments during the process (using the proper widget), you can refer to the possible available documents (Attachments TAB).

CMDBuild Demo			G		iinistrator   <u>Logout</u> r   <u>Administration</u>	17742		Open Source Co Manage	onfiguration and ment Database
Navigation 🗧	List - Request for o	change							
🗄 🧰 Dashboard	3 Start Request for	change	Completed	~					
∃ 🛄 Basic archives ∃ 🧰 Purchases	Request number	Start dat	e	Status		Category	Final result	Requester	
Locations	6	24/08/20	12 11:48:37	Documentatio	n requested	Not applicable	Negative	Smith James	
🗉 🔂 Assets	7	24/08/20	12 12:08:59	Per formed		Greate new ERP user	Positive	Brown Robert	
Computer PC Notebook Server Monitor Printer NetworkDevice Rack UPS License License Report Report Request for change	Activity Note Activity Note Add attachment Begin date	of 1 Relati		Attachme		a Search filter 🗼 Ce	ar filter 🔒 Print 🔹 Description		1 - 2 of 2 Operative Instruction
Class List + Processes + Dashboard +									
Utility +									

# Widgets prompted to use in the user activities of the workflow

# Widget list

CMDBuild makes some widgets available (visual controls), placed in the right part of the form, which manage the progress of the process through the provided activity.

Graphically, such controls are designed with buttons with the specified label during the definition step.

About the configuration, they are defined as "Extended attributes" (provided in the XPDL standard) using the TWE editor.

In this document, the data types are both original (integer, string, date, float, boolean) and complex types added in the workflows of CMDBuild (lookup = id + type + description, lookups = lookup array, reference = id + idclass + description, references = reference array).

Visual control	Description	Parameters	Notes
manageRelation	It shows the card list (which can be selected) in relation to the specified card according to the specified domain	Input: DomainName string ClassName string Objld integer ButtonLabel string EnabledFunctions character array Required integer IsDirect string or Input: DomainName string ObjRef reference ButtonLabel string EnabledFunctions character array Required integer Output: CheckArray references	EnabledFunctions is an array of boolean values which enables different functionalities according to the following positional method: - link element - add and link element - add and link element - activate selection check - activate selection radio button - modify relation - disconnect element - delete element - delete element The parameter Required = 1 must be indicated only if the selection of at least one element is compulsory IsDirect can take the values "true" or "false"
linkCards	It shows the paginated list - which can be selected - of all cards belonging to a class, with possible display on a geographical map	Input: ClassName string ButtonLabel string SingleSelect integer NoSelect integer Required integer Filter string DefaultSelection string	The parameter SingleSelect = 1 must be indicatedonly if the selection of one single row is allowed (radio-button rather than checkbox) The parameter NoSelect = 1 desables the selection of rows (neither radio button nor checkbox) The parameter Required = 1 forces the selection of one row at least

	AllowCardEditing integer DisableGridFilterTog	The Filter parameter is a CQL expression (CMDBuild query language) Example: Filter = "from Person where Id =
	gler <i>boolean</i>	{client:Customer.Id}" The optional parameter DefaultSelection
	Map <i>string</i> StartMapWithLatitude <i>integer</i>	specifies the CQL query used for the automatic selection when opening the widget
	de integer	The optional parameter AllowCardEditing = 1 adds an icon to edit the card
	integer Metadata string MetadataOutput	The optional parameter DisableGridFilterToggler = "true" hides the button "Disable filter"
	string	The optional Map parameter enables the map visualization (if it is set = 'enabled')
	CheckArray references	The parameters related to the initial presentation of the map are optional
	[metadataOutput] text	The Metadata variable accepts as unique value (waiting for future extensions) the 'point:POINT' string.
		The MetadataOutput variable accepts as unique value the '_metadataOutput' string that represents the name of the output variable.
		They both are necessary to manage the selection of a single point on an already existing polygonal. The point coordinates will be given back in the metadataOutput variable in the WGS84 format.
		A possible example: point:POINT(5847010.6684071 1438393.2786558)
It shows the specified card in the change (if Objld is specified), otherwise it allows the creation of a	Input: ClassName <i>string</i> ButtonLabel <i>string</i> ReadOnly integer or	Example: ClassName='User' ObjId=client:Requester ButtonLabel = 'Create or modify User' Requester
new card in the specified class	Input: Reference reference ButtonLabel string ReadOnly integer	Note: the prefix "client:" is required to access to a variable before the workflow is proceeded to the following step
	or	ReadOnly=1 shows the card read-only
	Input: ClassName string ObjId integer ButtonLabel string ReadOnly integer	
	specified card in the change (if Objld is specified), otherwise it allows the creation of a new card in the	integer DisableGridFilterTog gler booleanMap string StartMapWithLatitude integer StartMapWithLongitu de integer StartMapWithZoom integer MetadataOutput stringOutput: CheckArray references [metadataOutput]] textIt shows the specified card in the change (if Objid is specified), otherwise it allows the creation of a new card in the specified classIt shows the specified classIt shows the specified classIt shows the specified classIt shows the specified classInput: ClassName string ReadOnly integerInput: ClassName string ReadOnly integerInput: ClassName string ReadOnly integerInput: ClassName string ReadOnly integer

		<u>Output</u> : Reference <i>reference</i>	
createReport		Input: ReportType string ReportCode string ButtonLabel string ForcePDF integer ForceCSV integer Parameter-1 Parameter-2  Parameter-n <u>Output</u> : ReportURL string	ReportType can currently only take the 'custom' value ReportCode coincides with the report "Code" attribute in the schedule "Report" ForcePDF forces the output in PDF format ForceCSV forces the output in CSV format Parameter-1 Parameter-n they represent launch parameters provided by the report
manageEmail	It allows to product through template or write free e-mails which will be sent during the development of the process.	Input: ButtonLabel string ToAddresses string CCAddresses string Subject string Content string Assignments string ReadOnly boolean	Visualizing e-mails, the electronic mailbox will be checked for possible new e-mails The parameters ToAddresses, CcAddresses, Subject and Content are "string template" which can include "tags" for the "substitution" of variables (for further information see the next paragraph) It is required the configuration of parameters for the e-mail sending in the file <i>workflow.conf</i> .
openNote	It visualizes the page which includes the HTML editor to insert notes	<u>Input:</u> ButtonLabel <i>string</i>	It can't be used in the first process activity
openAttachment	It visualizes the page provided for the uploading of the file which has to be enclosed to the current process	Input: ButtonLabel <i>string</i>	It can't be used in the first process activity
calendar	It displays the calendar with the selected dates	Input: ButtonLabel string ClassName string Filter string EventStartDate date EventEndDate date EventTitle string	From the class ClassName you can collect the dates you want to display in the calendar, with possible filter (it is optional but it takes the precedence on the ClassName). The parameter EventEndDate is optional. EventTitle indicates the attribute that draws the text and writes it on the calendar for every date
presetFromCard	It populates the current activity with those data recovered by a selected card.	Input: ButtonLabel string ClassName string Filter string AttributeMapping string	ClassName, the name of the class, as an alternative to Filter, which is on the contrary a CQL expression. AttributeMapping is a string in the form of 'a1=c1,a2=c2' that shows how to chart activity attributes with the card ones. The comma separates the assignments.

webService	It displays the result of a call to Web Service (at the moment SOAP only) as a grid. You can select some rows of this grid to obtain their XML serialization as widget output.	Input: ButtonLabel string EndPoint string Method string NameSpacePrefix string NodesToUseAsRows string NodesToUseAsColu mns string SingleSelect='true' Mandatory='true' ReadOnly='true' String parameters OutputSeparator string Output: Output string variable	EndPoint=Service URL Method=Method name NameSpacePrefix=namespace prefix' (optional) NameSpaceURI='namespace URI' (optional) NodesToUseAsRows= Names of elements (separated by commas without spaces) of the answer to display in the grid NodesToUseAsColumns=Names of the elements (separated by commas without spaces) of the answer to use as grid columns. Call parameters (optional) = possible parameters provided for in the Web Service. Output variable(optional) that will be optimized through the XML serialization related to the selected nodes If it is string type, then the separator has also to be specified. OutputSeparator (optional)= character to separate the results. If it is missing, they will be given back as string array.
startWorkflow	It allows to start a workflow according two modalities: 1) configuration read by widgets 2) configuration read by a "support" table	1) <u>Input</u> : ButtonLabel <i>string</i> WorkflowCode <i>string</i> or 2) <u>Input:</u> ButtonLabel <i>string</i> FilterType <i>string</i> Filter <i>string</i> <u>Output</u> : processRef <i>ReferenceType</i>	<ol> <li>WorkflowCode name of the starting process</li> <li>FilterType supports at the moment just "cql"</li> <li>Filter the cql filter to select a series of card from a CMDBuild table.</li> <li>The result of the filter should be the same as the name list of the processes that should be started from the widget itself.</li> </ol>
grid	It allows to manage a row grid (by adding, removing and/or modifying the rows )	Input: ClassName string ButtonLabel string CardSeparator string AttributeSeparator <i>string</i> KeyValueSeparator	ClassName is the name of the class in which you want to work CardSeparator separator among the various inserted cards (default ";") AttributeSeparator separator among the attributes of the same card (default "&")

			1
		string	KeyValueSeparator separator between an
		PresetsType="functio	attribute and its value (default "==")
		n"	The output variable will be a single string
		Presets string	containing the serialization of the inserted
			data, separated by the above-mentioned
		<u>Output</u> :	characters
		Output string	
		variable	PresetType and Preset are necessary to upload in advance some values into the grid; these values will be then edited directly by the user. If you want to use a variable as input or the output of another
			grid, you simply have to specify the key Preset=InputString where InputString is a formatted string like the grid output.
			If you want to upload in advance the grid starting from a function (having as column
			names the same reference class fields) you have specify PresetsType="function" and
			Presets="wf_function_name" where "wf_function_name" is the name of a stored
			procedure in the database defined according to criteria used to create
			dashboards. Any parameter has to be specified in
			succession in the form of:
			Param1="value1" (function input
			parameter)
			Param2="value2" (function input parameter)
			There is also the possibility to upload in advance the values into the grid through the proper widget button called "Import from CSV". The file must follow the norms defined for the CSV file importation into CMDBuild. You will also have the possibility to specify the separator and the way to import the data (Replace or Add)
customForm	It allows to manage a form or a row grid	<u>Input</u> : ButtonLabel <i>string</i>	The structure of the custom form can be defined starting from:
	(by adding,	ModelType "[form	form - JSON item array
	removing and/or	class function]"	class - attributes of a class
	modifying the rows )	Layout <i>"[grid</i>  form]" DataType [raw_json	function - function input parameters
		raw_text function]	The layout can be a form (as if it is a
		ReadOnly <i>"[true</i>   <i>false]"</i>	CMDBuild card) or a row series.
		Required <i>"[true false]"</i> AddDisabled <i>"[true</i> ]	The data of the widget can be initialized starting form:
		false]"	raw_json - JSON item array
		DeleteDisabled "[true]	raw text – well-structured strings of text
		<i>false]"</i> ImportDisabled <i>"[true</i> ]	function – output values of a function
		false]"	Data can be serialized as type of text (see

		ModifyDisabled "[true  false]" SerializationType "[json text]" KeyValueSeparator string AttributesSeparator string RowsSeparator string Output: Output string variable	the widget grid) or as type of json.
navigationTree	It allows to select one or more data cards through an interface that is based on a preconfigured navigation tree (subset of domain graph)	Input: NavigationTreeName string ButtonLabel string <u>Output</u> : CheckArray <i>references</i>	NavigationTreeName represents the name of that tree you want to display
adminStart	By a process with more start activities distinct for each group, it singles out the activity for the administer user		No input nor output parameters It is an "extended attribute", not a widget (it doesn't have a user interface), but it is described in this section, since it is configured like widgets.

#### Further information for the use of "string template" in the tool manageEmail

The tool *manageEmail* allows to write e-mails which will be sent during the development of the process. Visualizing e-mails, the electronic mailbox will be checked for possible new e-mails to visualize the grid.

Input parameters	<ul> <li>string ButtonLabel</li> <li>one or more blocks for the e-mails definition         <ul> <li>string template ToAddresses: recipient's addresses</li> <li>string template CcAddresses: carbon copy addresses</li> <li>string template Subject: e-mail subject</li> <li>string template Content: e-mail body (HTML)</li> <li>string template Condition: javascript expression whose evaluation defines if the e-mail is generated or not</li> <li>other optional parameters which include queries or javascript expressions</li> <li>flag ReadOnly: read-only email</li> </ul> </li> </ul>
output parameters	none

The only-read *flag* is seen as a boolean value; a boolean value (of the process), a positive integer value or a non empty string are considered *true* 

In the *template strings* the variables, written in the form {namespace:localname}, are interpreted in a different way depending on the namespace (if omitted, it defaults to "server").

client:name client:name.ld client:name.Description	Form's <i>name</i> variable; for attributes such as LookUp or Reference you have to specify, with the bullet list, whether you want the <i>Id</i> or the <i>Description</i>
server:name	Process <i>name</i> variable in the previous step
<b>xa</b> :name	Variable <i>name</i> of the extended attribute definition, extended as template excluding the variables with namespace <i>js</i> and <i>cql</i>
<b>user</b> :id <b>user</b> :name	ID and name of the connected user
group:id group:name	ID and name of the connected group
js:name	Variable <i>name</i> of the extended attribute definition interpreted as a template and evalued as a javascript code
<b>cql</b> :name.field	Variable <i>name</i> of the extended attribute definition interpreted as a template and evalued carrying out a CQL query, whose field is identified by <i>field</i>

The definition blocks of the e-mails can be written in two ways:

```
ToAddresses="..."
CcAddresses="..."
Subject="..."
Content="..."
```

or (if you want to specify more than one e-mail):

```
ToAddresses1="..."
CcAddresses1="..."
Subject1="..."
Content1="..."
ToAddresses2="..."
CcAddresses2="..."
Subject2="..."
Content2="..."
```

#### Example 1

```
ToAddresses="foo@example.com"
Subject="{cql:QueryRequester.Description} - {client:Request}"
QueryRequester="select Description,Email,Office from Employee where Id = {cql:SillyQuery.Id}"
SillyQuery="select Id from Employee where Id={client:Requester}"
```

Address: The recipient's address is statically completed with the string foo@example.com

### Body: Message Body Empty

Subject:

- The variable QueryRequester selects an Employee card which includes the fields Description, Email and Office; the extracted values are available using for example the syntax {cql:QueryRequester.Description}, which will be replaced with the field Description extracted from the variable QueryRequester
- Inside QueryRequester, {cql:SillyQuery.Id} will be replaced with the Id field of the card returned from the SillyQuery (indeed nested queries are supported), replaced before with {client:Requester} with the value taken in the form
- {client:Request} of will be completed with the form value

#### Example 2

Content="The requester, {js:JoinJS}, belonging to the office {cql:QueryRequester.Office\_value} requests:<br /><br />{server:Request}" JoinJS="{js:FirstJS}+"#"+{js:SecondJS}" FirstJS="{cql:QueryRequester.Description}.slice(0,{xa:SplitLength})" SecondJS="{cql:QueryRequester.Description}.slice({xa:SplitLength})" SplitLength=2 QueryRequester="select Description,Email,Office from Employee where Id = {Requester}"

This is an example of more complexity.

In the body there are three variables which must by replaced:

- {js:JoinJS} values the extended attribute variable like a javascript expression, splitting with # the variables FirstJS and SecondJS, always valued through javascript
- {js:FirstJS} and {js:SecondJS} include both a variable taken from a field of CQL query QueryRequester and a static variable taken from the ones of the extended attribute
- {cql:QueryRequester...} includes a reference to a server side variable called Requester
- {cql:QueryRequester.Office\_value} uses the Office reference description instead of its ID (that would be just Office)
- {server:Request} takes a server side variable (as Requester), but it also states the namespace

# API prompted to use in the automatic activities of the workflow

In CMDBuild there are some APIs (Application Programming Interface) which can be used in the automatic activities of the workflow for the script writing; so it is possible to implement custom behaviors (manipulation of process variables, card creation and relations in CMDB, e-mail sending, report creation, etc).

• The condition to send e-mails is always verified since {xa:SplitLength} is constant and the javascript expression is always true.

#### **General Information**

#### Key words

Process
ProcessId: int
Id of the current process
ProcessClass: String
Class name of the current process
ProcessCode: String
univocal ProcessInstanceId of the current process

#### Performer

<u>CurrentUser: ReferenceType</u> reference to the User that performed the last activity of the current process <u>CurrentGroup: ReferenceType</u>

reference to the Role that performed the last activity of the current process

API	
<u>cmdb</u>	
	it identifies the native functions in CMDBuild

#### Management of CMDBuild items

They concern the CMDBuild specific data; for other data (integer, string, date, float) you can use all manipulation methods offered by the Java language.

#### ReferenceType

Methods	
<u>getId(): int</u>	

it returns the Reference id

#### getDescription(): String

it returns the Reference description

#### LookupType

#### Methods

getId(): int

it returns theLookup id

#### getType(): String

it returns the type of Lookup

#### getDescription(): String

it returns the Lookup description

#### getCode(): String

it returns the Lookup code

#### CardDescriptor

Methods
getClassName(): String
it returns the Class name for a CardDescriptor variable
getId(): int
it returns the Id name for a CardDescriptor variable
equals(CardDescriptor cardDescriptor): boolean
it compares the CardDescriptor variable with the specified one

#### Card

Methods
getCode(): String
it returns the Code for a Card variable
getDescription(): String
it returns the Description for a Card variable
has(String name): boolean
it controls the presence of the specified attribute in the Card variable
hasAttribute(String name): boolean
it controls the presence of the specified attribute in the Card variable
get(String name): Object
it returns the specified attribute value of the Card variable
getAttributeNames(): Set <string></string>
it returnsthe attributes list of the Card variable
getAttributes(): Map <string, object=""></string,>

it returns the attributes list and their values of the Card variable. The returned values respect the CMDBuild types (ReferenceType, LookupType, Date, Integer, ...)

#### Attachments

Methods						
fetch(): Iterable <attachmentdescriptor></attachmentdescriptor>						
it returns the attachments list of the Card or of the instantiated process						
upload(Attachment attachments):void						
it attaches the documents to the card or to the instantiated process						
upload(String name, String description, String category, String url):void						
it creates an attachment with name, description and category specified starting from the file with						
the specified URL and attaches it to the card or to the instantiated process						
selectByName(String names): SelectedAttachments						
it returns the attachments of the card or of the instantiated process with the specified name						
selectAll(): SelectedAttachments						
it returns all attachments of the card or of the instantiated process						

#### AttachmentDescriptor

#### Methods

getName(): String it returns the name of the attachment

getDescription(): String it returns the attachment description

getCategory(): String it returns the attachment category

#### Attachment

Methods	
getUrl(): String	
it returns the URL of the file	

#### DownloadedReport

Methods					
getUrl(): String					
it returns the local URL where the report has been saved					
equals(DownloadedReport downloadedReport): boolean					
it compares the DownloadedReport variable with the specified one					

#### Access methods to CMDBuild

#### NewCard

## Builders <a href="https://www.ard.com/articles.com/articl

it creates a new Card created in the specified Class of CMDBuild

#### Modifiers

withCode(String value): NewCard

it adds the Code to the new card created in CMDBuild

withDescription(String value): NewCard

it adds the Description to the new card created in CMDBuild

with(String name, Object value): NewCard

it adds the value specified for the specified attribute to the new card created in CMDBuild

withAttribute(String name, Object value): NewCard

it adds the value specified for the specified attribute to the new card created in CMDBuild

#### Actions

```
create(): CardDescriptor
```

it creates the new card in CMDBuild setting the attributes previously defined

#### Example:

```
/*
* Creation of a new card in the "Employee" class having
* the following attributes:
*
  "Code"
            =
                     "T1000"
* "Name"
              =
                   "James"
* "Surname" =
                    "Hetfield"
*/
cdNewEmployee = cmdb.newCard("Employee")
.withCode("T1000")
.with("Name", "James")
.withAttribute("Surname", "Hetfield")
.create();
```

#### ExistingCard

#### Builders

existingCard(String className, int id): ExistingCard
it creates a Card existing in the specified Class having the specified Id to query CMDBuild
existingCard(CardDescriptor cardDescriptor): ExistingCard

#### it creates an existing Card indicated by the specified CardDescriptor to query CMDBuild

Modifiers
withCode(String value): ExistingCard it sets the Code for the Card requested to CMDBuild
withDescription(String value): ExistingCard it sets the Description for the Card requested to CMDBuild
<u>with(String name, Object value): ExistingCard</u> it sets the specified attribute with the specified value for the Card requested to CMDBuild
withAttribute(String name, Object value): ExistingCard it sets the specified attribute with the specified value for the Card requested to CMDBuild
withAttachment(String url, String name, String category, String description): ExistingCard it attaches a file (pointed out through a server local url) to the selected card by setting the file name, its category and its description
attachments(): ExistingCard it allows you to access the attachments of the selected card
selectAll(): ExistingCard
<u>selectByName(String name1, String name2,):ExistingCard</u> it allows you to select all documents of the selected card

#### Actions

update()

it updates the Card in CMDBuild by setting the attributes previously indicated with the specified values

delete()

it deletes (logic delete) the Card from CMDBuild

If the "attachments" modifier has been used, it will delete only the selected files

fetch(): Card

it requests the Card to CMDBuild with the attributes previously indicated. If no modifier has been used, it requests the whole Card (with all attributes)

fetch(): Iterable<AttachmentDescriptor>

If the "attachments" modifier has been used, the method returns the list of the card attachments

upload(Attachment attachment, Attachment attachment2,.,)

to be used in the presence of the "attachments" modifier: it attaches one or more files to the card

upload(Attachment attachment, String description, String category, String url)

to be used in the presence of the "attachments" modifier: it attaches to the card a single file with specified description and category

#### download(): Iterable<Attachment>

If the "attachments" modifier has been used, the method returns the selected attachments of the card

#### copyTo()

If the "attachments" modifier has been used, the method copies a selected attachment of the card into a specified destination

#### moveTo()

If the "attachments" modifier has been used, the method moves a selected card attachment into a specified destination

#### Examples:

```
/*
* It modifies the card previously created in the class "Employee"
 * by setting the following attributes:
 * "Phone"
               =
                      "754-3010"
 * "Email"
                      "j.hetfield@somemail.com"
               =
 */
cmdb.existingCard(cdNewEmplyee)
.with("Phone", "754-3010")
.withAttribute("Email", "j.hetfield@somemail.com")
.update();
/*
* (Logic) delete of the card previously created in the class
* "Emplyoee"
*/
cmdb.existingCard(cdNewEmplyee)
.delete();
/*
* Delete of the card attachment that was previuosly
* created in the "Employee" class
 */
Iterable <AttachmentDescriptor> attachments =
cmdb.existingCard(cdNewEmplyee)
.attachments()
.fetch();
```

```
/*
 * Delete of the card attachment that was previuosly
 * created in the "Employee" class
 */
cmdb.existingCard(cdNewEmplyee)
.attachments()
.selectByName(String[]{"attachment-name"})
.delete();
```

#### NewProcessInstance

#### Builders

<u>newProcessInstance(String className): NewProcessInstance</u> it creates a new process instance created in CMDBuild for the specified process

#### Modifiers

withDescription(String value): NewProcessInstance

it adds the Description to the new card created in CMDBuild

with(String name, Object value): NewProcessInstance

it adds the value specified for the specified attribute to the new process created in CMDBuild

withAttribute(String name, Object value): NewProcessInstance

it adds the value specified for the specified attribute to the new process created in CMDBuild

#### Actions

start(): ProcessInstanceDescriptor

it creates the new process in CMDBuild setting the attributes previously defined, and does not advance

startAndAdvance(): ProcessInstanceDescriptor

it creates the new process in CMDBuid setting the attributes previously defined, and advances at the following step

#### Example:

```
/*
 * Creation of a new card in the "RequestForChange" class
 * having the following attributes
 * "Requester" = "James Hetfield"
 * "RFCExtendedDescription" = "My printer is broken"
 */
pidNewRequestForChange =
cmdb.newProcessInstance("RequestForChange")
```

```
.with("Requester", "James Hetfield")
.withAttribute("RFCExtendedDescription", "My printer is broken")
.startAndAdvance();
```

#### ExistingProcessInstance

#### **Builders**

existingProcessInstance(String processClassName, int processId): ExistingProcessInstance it creates a process instance existing in the specified process class with the specified Id

#### Modifiers

with(String name, Object value): ExistingProcessInstance

it sets the specified attribute with the specified value for the process instance

withAttribute (String name, Object value): ExistingProcessInstance

it sets the specified attribute with the specified value for the process instance

withDescription(String value): ExistingProcessInstance

it sets the specified attribute with the specified value for the process instance

attachments(): Attachments

it allows you to access the attachments of the process instance

Actions
abort(): void
it aborts the process instance
advance(): void
it advances a process instance
resume(): void
it resumes the hanging process instance
suspend(): void
it suspends the open process instance
update(): void
it updates the process instance

#### Example:

```
/*
* Update of the process instance in the class "Request
* for change" with Id = pid by editing the requester and
* advancing the process at the following step
*/
```

```
cmdb.existingProcessInstance("RequestForChange", pid)
.with("Requester", cdNewEmployee.getId())
.advance();
```

#### NewRelation

#### Builders

<u>newRelation(String domainName): ExistingProcessInstance</u> it creates a new relation added in the specified Domain of CMDBuild

#### Modifiers

withCard1(String className, int cardId): NewRelation					
it sets the card in the source side of the relation					
withCard2(String className, int cardId): NewRelation					
it sets the card in the target side of the relation					

#### Actions

#### create()

it creates the new relation in CMDBuild among the Cards indicated in the specified Domain

#### Example:

```
/*
 * Creation of a new relation in the "AssetAssignee" domain
 * between a card of the selected "Asset" class,
 * through the "Item" Reference attribute, and
 * the card previously created in the "Employee" class
 */
cmdb.newRelation("AssetAssignee")
.withCard1("Employee", cdNewEmployee.getId())
.withCard2("Asset", Item.getId())
.create();
```

#### ExistingRelation

#### Builders

existingRelation(String domainName): ExistingRelation it creates an existing relation in the specified Domain of CMDBuild

#### Modifiers

withCard1(String className, int cardId): ExistingRelation

it sets IdClass and I'ObjId of the Card from the source side of the relation

withCard2(String className, int cardId): ExistingRelation

it sets IdClass and I'ObjId of the Card from the target side of the relation

#### Actions

#### delete()

it deletes (logic delete) the relation existing in CMDBuild among the Cards indicated in the specified Domain

#### Example:

```
/*
 * Delete the relation on the "AssetAssignee" domain
 * among the cards previously indicated
 */
cmdb.existingRelation("AssetAssignee")
.withCard1("Employee", cdNewEmployee.getId())
.withCard2("Asset", Item.getId())
.delete();
```

#### QueryClass

#### Builders

<u>queryClass(String className): QueryClass</u> it creates a query that queries the class specified in CMDBuild

Modifiers
withCode(String value): QueryClass it sets the Card Code for the filter used to query CMDBuild
withDescription(String value): QueryClass
it sets the Card Description for the filter used to query CMDBuild
with(String name, Object value): QueryClass
it sets the value for the specified attribute of the Card for the filter used to query
CMDBuild
withAttribute(String name, Object value): QueryClass

it sets the value for the specified attribute of the Card for the filter used to query

CMDBuild

#### Actions

fetch(): List<Card>

it performs the search query on the specified Class of CMDBuild and returns the list of those Cards that respect the filter previously set

#### Example:

```
/*
 * List of the cards of the "Employee" class having
 * the "State" attribute set to 'Active'
 */
Employees = cmdb.queryClass("Employee")
.with("State", "Active")
.fetch();
```

#### CallFunction

#### Builders

```
<u>callFunction(String functionName): CallFunction</u>
it creates a call to a stored procedure previously defined in PostgreSQL
```

#### Modifiers

<u>with(String name, Object value): CallFunction</u> it sets the value of the input parameter specified for the stored procedure

#### Actions

<u>execute(): Map<String, String></u>

it performs the stored procedure and returns the list of the output parameters with the related values

#### Example:

```
/*
 * Call of the stored PostgreSQL procedure
 * "cmwf_getImpact"(IN "DeliveryDate" date, IN "Cost" integer,
 * OUT "Impact" character varying)
 * that computes the impact level (attribute of
 * "Impact" process) of an activity on a scale of "High",
 * "Medium" and "Low", given in input the expected delivery
 * date (process attribute "ExpectedDeliveryDate") and
 * the price (attribute "ManHoursCost") expressed in hour/employee
 */
spResultSet = cmdb.callFunction("cmwf_getImpact")
.with("DeliveryDate", ExpectedDeliveryDate.getTime())
.with("Cost", ManHoursCost)
.execute();
Impact = spResultSet.get("Impact")
```

<u>Note</u>: SQL functions - which should be called - must be defined according to CMDBuild standards. For their definitio,n see the Administrator Manual, section Cart TAB, paragraph "Definition of the data source (PostgreSQL function)".

#### QueryRelations

Builders
queryRelations(CardDescriptor cardDescriptor): ActiveQueryRelations
it creates a query to ask CMDBuild the Cards related to the specified one
queryRelations(String className, int id): ActiveQueryRelations
it creates a query to ask CMDBuild the Cards related to that specified by className
and id

#### Modifiers

withDomain(String domainName): ActiveQueryRelations it sets the Domain to perform the query

#### Actions

fetch(): List<CardDescriptor>

it performs the query on CMDBuild using the parameters previously defined, it returns the list of the linked Cards

#### Example:

```
/*
 * List of "Assets" linked to the "Employee" card indicated
 * by the CardDescriptor cdNewEmployee previously created,
 * through the relation on the domain "AssetAssignee"
 */
assets = cmdb.queryRelation(cdNewEmployee)
.withDomain("AssetAssignee")
.fetch();
```

#### CreateReport

#### Builders

<u>createReport</u>(String title, String format): CreateReport it creates the Report in the specified format (pdf, csv) with the specified Title

#### Modifiers

<u>with</u>(String name, Object value): CreateReport it sets the input parameter value specified for the Report

#### Actions

```
<u>download()</u>: DownloadedReport
it generates the indicated Report using the parameters previously defined
```

#### Example:

```
/*
 * It generated the Report "DismissedAssets" which shows the list
 * of the abandoned Assets
 */
newReport = cmdb.createReport("Assigned assets to")
.download();
```

#### NewMail

#### Builders

newMail(): NewMail it creates a new e-mail to send

#### Modifiers

withFrom(String from): NewMail
it sets the sender of the e-mail to send
withTo(String to): NewMail
it sets the recipients of the e-mail to send
withCc(String cc): NewMail
it sets the carbon copy recipients of the e-mail to send
withBcc(String bcc): NewMail
it sets the blind carbon copy recipients of the e-mail to send
withSubject(String subject): NewMail
it sets the subject of the e-mail to send
withContent(String content): NewMail
it sets the text of the e-mail to send
withContentType(String contentType): NewMail
it sets the content MimeType of the e-mail to send, the allowed values are "text/html" or
"text/plain". If not otherwise specified, the default value is "text/plain"
withAttachment(URL url): NewMail
it sets the url of a document to enclose to the e-mail
withAsynchronousSend(bool boolean): NewMail
it sends the e-mail asynchronously in spite of the script; in this way any timeout
problem will be avoided, but you will not be able to intervene in case of error by sending the e-
mail

Actions							
<u>send()</u>							

it performs the e-mail sending using the previously defined statements

#### Example:

```
/*
 * Send a new email
 */
cmdb.newMail()
.withFrom("fromaddress@somemail.com")
.withTo("toaddress@somemail.com")
.withCc("ccaddress@somemail.com")
.withSubject("Mail subject")
.withContent("Mail content")
.send();
```

#### NewMailQueue

Builders					
newMailQueue(): NewMailQueue					
it creates a new email queue					
Methods					
newMail(): QueueableNewMail					
it adds a new email to the queue					
sendAll(): void					

it sends all emails from the queue

```
/*
 * Send a new email
 */
cmdb.newMailQueue()
.newMail()
.withFrom("fromaddress@somemail.com")
.withTo("toaddress@somemail.com")
.withCc("ccaddress@somemail.com")
.withSubject("Mail subject")
.withContent("Mail content")
.add()
```

.sendAll();

#### Methods for types conversion

#### ReferenceType

Methods
<u>referenceTypeFrom(Card card): ReferenceType</u> it returns the ReferenceType item related to the specified Card
referenceTypeFrom(CardDescriptor cardDescriptor): ReferenceType it retuns the ReferenceType item related to the specified CardDescriptor
referenceTypeFrom(int id): ReferenceType it returns the ReferenceType item related to the card with the specified Id

#### Example:

```
/*
 * Set the "Requester" process attribute Reference
 * type, given the "cdNewEmployee" CardDescriptor
 * previously created
 */
```

Requester = cmdb.referenceTypeFrom(cdNewEmployee);

#### LookupType

#### Methods

<u>selectLookupById(int id): LookupType</u> it returns the LookupType item with the specified Id
<u>selectLookupByCode(String type, String code): LookupType</u> it returns the LookupType item with specified Type and Code
selectLookupByDescription(String type, String description): LookupType it returns the LookupType item with specified Type and Description

#### Example:

#### CardDescriptor

Methods
cardDescriptorFrom(ReferenceType reference): CardDescriptor
it returns the CardDescriptor of the specified card through the specified ReferenceType
item

#### Example:

```
/*
 * Get the CardDescriptor related to the "Requester"
 * process attribute Reference type
 */
cdSelectedEmployee = cmdb.cardDescriptorFrom(Requester);
```

#### Card

#### Methods

```
<u>cardFrom(ReferenceType reference): Card</u>
it returns the Card item of the specified card through the specified ReferenceType item
```

#### Example:

```
/*
 * Get the complete Card related to the "Requester"
 * process attribute Reference type
 */
selectedEmployee = cmdb.cardFrom(Requester);
```

## **Appendix: Documentation to use TWS 2.3**

#### Foreword

In appendix you will find the specified technical documentation of the workflow system used until CMDBuild 1.5, whose compatibility is maintained also in CMDBuild 2.0; as soon as possible it will be discarded.

We must remember that in CMDBuild 2.0 there is the possibility to work - alternatively - both with Together Workflow Server 2.3 (the version used until CMDBuild 1.5, based on XPDL 1.0) and with the new version Together Workflow Server 4.4 (based on XPDL 2.0).

It is advisable to migrate in a short time, since that double compatibility will be maintained for a limited period.

#### Automatic methods used in the workflow

In order to use Together Workflow Server 2.3 (passed by the system based on Together Workflow Server 4.4), CMDBuild provides some methods ("tools"), which can be used inside the "tool activities" (automatic activities) in order to perform the various operation typologies:

- methods for the manipulation of the variables: conversion among data typologies, strings connection, etc.
- methods for the flow control: iterator, process suspension, process reboot
- access methods to CMDB: create a new card, read or change attribute, create relation, etc.
- external methods: sending e-mails, reading system time, etc.

#### Methods for the manipulation of the variables

ΤοοΙ	Description	Input parameters	Output parameters	Notes
addDays	It adds to the specified date the indicated number of days	InputDate <i>date</i> days <i>integer</i>	OutputDate <i>date</i>	
boolToString	It converts a boolean variable to a string	InputBool <i>boolean</i>	OutputString string	
boolCopy	It copies the value of a boolean variable into another boolean one	From <i>boolean</i>	To boolean	
clearIterator	It resets the iterator	RefArray references HasNext boolean CurrentIndex integer	RefArray references HasNext boolean CurrentIndex integer	RefArray is set to null, CurrentIndex is set to 0 and HasNext to false
clearLookup	It resets the value	Lookup <i>lookup</i>		The value of the "Id"

	of a Lookup variable			attribute is set to -1
clearReference	It resets the value of a Reference variable	Ref reference		The value of the "Id" attribute is set to -1
concat concat3 / concat4 / / concat8	It concatenates two or more strings	InputString1 <i>string</i> InputString2 <i>string</i>  InputStringn <i>string</i>	OutputString string	
createReferenceObj	It creates a reference variable and initializes it	ClassName <i>string</i> ObjId <i>integer</i> Description <i>string</i>	OutRef <i>reference</i>	The variable is initialized with the values read by the attributes "ClassName", "ObjId" and "Description" of the specified card
dateToString	It converts a date variable into a string	InputDate <i>date</i>	OutputString string	
floatToString	It converts a float variable into a string	InputFloat <i>float</i>	OutputString string	
floatCopy	It copies the value of a float variable into another float variable	From <i>float</i>	To float	
getReferenceId	It extracts the "Id" attribute from a reference variable	Ref reference	CardId <i>integer</i>	
getReferenceClassId	It extracts the "ClassId" attribute from a reference variable	Ref reference	ClassId integer	
getLookupDescription	It extracts the "Description" attribute from a lookup variable	Lookup <i>lookup</i>	Description <i>string</i>	
getLookupld	It extracts the "Id" attribute from a lookup variable	Lookup <i>lookup</i>	ld Integer	
getLookupCode	It extracts the "Code" attribute from a lookup variable	Lookup <i>lookup</i>	Code String	
getReferenceDescri ption	It extracts the "Description" attribute from a reference variable	Ref reference	Description string	
getReferenceFrom Array	It extracts the specified	RefArray references	OutRef <i>reference</i>	If the array is null or the index is higher than its

	Reference from the specified array	Index integer		dimension, it returns "null"
intToString	It converts an integer variable into a string	InputInt <i>integer</i>	OutputString string	
intCopy	It copies the value of an integer variable into another integer variable	From <i>integer</i>	To integer	
lookupToString	It converts the "Id" field of the lookup variable into a string	InputLookup <i>lookup</i>	OutputString string	
nextInt	It increases the specified integer variable	InputInt <i>integer</i>	InputInt <i>integer</i>	
referenceToString	It converts the "Id" field of the reference variable into a string	InputReference <i>reference</i>	OutputString string	
stringToDate	It converts a string variable into a date	InputString string	OutputDate date	It accepts as input formats YY/MM/dd or YY/mm/dd HH:mm:ss
stringCopy	It copies the value of a string variable into another string variable	From string	To string	
dateCopy	It copies the value of a date variable into another date variable	From date	To date	
stringToBool	It converts a string variable into a boolean value	From string	To boolean	It accepts as input the true or false strings
stringToInt	It converts a string variable into an integer	From string	To integer	In input it accepts the representation of an integer number in the shape of string
stringToFloat	It converts a string variable into a float	From string	To float	In input it accepts the representation of a float in the shape of string

#### Methods for the flow control

ΤοοΙ	Description	Input parameters	Output parameters	Notes
nextRef	It increases the	RefArray	HasNext	RefArray is a reference

	iterator on a reference array	<i>references</i> CurrentIndex <i>integer</i>	<i>boolean</i> CurrentIndex <i>integer</i> CurrentValue <i>reference</i>	array, CurrentValue is the reference corresponding to the current index
resetIterator	It resets the iterators	RefArray references	HasNext <i>boolean</i> CurrentIndex <i>integer</i>	CurrentIndex is set to 0, HasNext is true if the array is not empty
resumeProcess	It reboots the specified process	ProcessInstanceld string Complete integer		The status of the specified process must be "Suspended" If "Complete" takes on the value 1, the process steps forward
suspendProcess	It suspends the specified process	ProcessInstanceId string		The constant "CURRENT" can be used to indicate the current process
				The process is suspended immediately before the following manual activity
voidApp	Null tool			

#### Access methods to CMDB

ΤοοΙ	Description	Input parameters	Output parameters	Notes
createCard	It creates a new card and returns the "Id"	ClassName <i>string</i> CardCode <i>string</i> CardDescription <i>string</i>	Cardld <i>integer</i>	The method sets only the basic attributes "Code" and "Description" In order to set the other ones, you have to use the updateAttribute tool or define a createCard metatool
createCardRef	It creates a new card and returns the reference	ClassName <i>string</i> CardCode <i>string</i> CardDescription <i>string</i>	CardReference <i>reference</i>	The method sets only the basic attributes "Code" and "Description" In order to set the other ones, you have to use the updateAttribute tool or define a createCard metatool
createRelation	It creates a relation between two cards	DomainName string ClassName1 string ClassName2 string ObjId1 integer ObjId2 integer	Done <i>boolean</i>	

createRelation1Ref	It creates a relation between two cards, the first of them is specified by reference	DomainName string ObjReference1 reference ClassName2 string ObjId2 integer	Done <i>boolean</i>	
createRelation2Ref	It creates a relation between two cards, the second of them is indicated by reference	DomainName string ClassName1 string Objld1 integer ObjReference2 reference	Done <i>boolean</i>	
createRelationRefs	It creates a relation between two cards, both specified by reference	DomainName string ObjReference1 <i>reference</i> ObjReference2 <i>reference</i>	Done <i>boolean</i>	
deleteRelation	It removes a relation between two cards	DomainName string ClassName1 string ClassName2 string ObjId1 integer ObjId2 integer	Done <i>boolean</i>	
deleteRelationByR eference	It deletes a relation between two cards, both specified by reference	DomainName string ObjReference1 reference ObjReference2 reference	Done <i>boolean</i>	
selectAttribute	It reads an attribute of the specified card	ClassName <i>string</i> AttributeName <i>string</i> Objld <i>integer</i>	AttributeValue <i>string</i>	The returned value is always represented by a string
selectAttributeFrom Reference	It reads an attribute of the specified card, specified by reference	ObjReference <i>reference</i> AttributeName <i>string</i>	AttributeValue <i>string</i>	The returned value is always represented by a string
selectLookup	It reads the description of a Lookup entry, specified by type and "Id"	Type <i>string</i> LookupId <i>integer</i>	LookupDescription string	
selectLookupByld	It returns a Lookup entry, specified by "Id"	Lookupld <i>integer</i>	Lookup <i>lookup</i>	
selectLookupByTy peDesc	It returns a Lookup entry, specified by type and description	Type <i>string</i> Description <i>string</i>	Lookup <i>lookup</i>	
selectLookupByTy	It returns a Lookup	Type <i>string</i>	Lookup <i>lookup</i>	

peCode	entry, specified by type and code	Code string		
selectReferenceBy Code	It returns a reference item corresponding to the card specified by code	ClassName <i>string</i> Code <i>string</i>	OutRef reference	
selectReferenceBy CustomAttribute	It returns a reference item corresponding to the indicated card through a generic attribute	ClassName <i>string</i> AttributeName <i>string</i> AttributeValue <i>string</i>	OutRef <i>reference</i>	
selectReferenceBy Reference	It returns a reference item corresponding to a reference attribute existing in the card specified by reference	ObjReference reference AttributeName string	OutRef <i>reference</i>	
selectRelations	It returns on a specific domain an array of references corresponding to the cards related to the one given by the specified id and class	ClassName string Cardld integer DomainName string	RefArray <i>relations</i>	
selectRelationsByR eference	It returns on a specific domain an array of references corresponding to the cards related to the one given by the specified reference	ClassName string Cardld integer DomainName string	RefArray <i>relations</i>	
updateAttribute	It modifies a card	ClassName <i>string</i> AttributeName <i>string</i> Objld <i>integer</i> AttributeValue <i>string</i>	Done <i>boolean</i>	The method edits only the specified attribute In order to edit more attributes, you have to define an updateCard metatool
updateAttributeRef	It edits a card specified by reference	ObjRef <i>reference</i> AttributeName <i>string</i> AttributeValue <i>string</i>	Done <i>boolean</i>	

#### External methods

ΤοοΙ	Description	Input parameters	Output parameters	Notes
getCurrentTimestamp	It returns the system date and time		TheDate <i>date</i>	
getCurrentGroupR eference	It returns a reference variable corresponding to the current user group		GroupRef reference	The returned item corresponds to the current group card in the CMDB "Role" table
getCurrentUserRe ference	It returns a reference variable corresponding to the current user		UserRef <i>reference</i>	The returned item corresponds to the current user card in the CMDB "Role" table
getReportFullUrl	It returns the link to the report created with the extended attribute createReport	ReportUrl <i>string</i>	ReportUrl string	
sendMail	It sends an e-mail	FromAddresses string ToAddresses string CCAddresses string BCCAddresses string Subject string Content string UrlAttachments string MimeType string		The tool provides that Shark parameters related to the e- mail sending are correctly configured Parameters From, To and Attach can include more values concatenated with "," Parameters CCAddresses, BCCAddresses and UrlAttachments can be set by an empty string MimeType can take the values "text/html" or "text/plain"

#### Template automatic methods usable in the workflow

In order to use Together Workflow Server 2.3 (passed by the system based on Together Workflow Server 4.4), CMDBuild provides some templates of automatic methods (meta-tools), used for the definition of tools.

For the creation of new "tools", custom CMDBuild provides the following steps:

- creation of a new "Application" with TWE Together Workflow Editor 4.4 (you can access the list from the process features), with the proper button "Create new element"
- the completion of the "Application" definition by clicking on the new row added to the list and setting the following parameters:
  - Id = name for the new "tool"
  - Name = you can set the same value chosen in the previous field
  - Formal parameters = adding as many input and output parameters as provided in the tool (as you can see in the following table)
  - Extended attribute "ToolAgentClass"
  - <sup>•</sup> further specific extended attributes in the meta-tool (as you can see in the following table)

Template type	Description	Input parameters	Output parameters	Notes
createCard	Card creation in the CMDB	Attributes list set in the new card or ClassName <i>string</i> List of input parameters provided by the function	CardReference <i>reference</i>	It returns the id of the created card The second specification of the input parameters can be used if you exclude ClassName from the external attribute list (see the following table)
createReport	Running a report	List of input parameters provided by the report	ReportURL <i>string</i>	The returned URL can be used to enclose the report to an e-mail with the tool sendMail
executeFunction	Execute PostgreSQL functions	List of input parameters provided by the function	List of output parameters provided by the function	There must be at least one input parameter and one output parameter, even if they are fake
startProcess	Instance initiation of another process	Attributes list set during the process initiation	ProcessInstanceld string	It returns the process instance name (string type)
updateCard	Card update in the CMDB	ClassName <i>string</i> Objld <i>integer</i> Attributes list updated in the card or	Done <i>boolean</i>	

		ObjRef <i>reference</i> List of input parameters provided by the function		
updateProcess	Save or progress of another process instance	ProcessInstanceId <i>string</i> Attributes list to set	Done <i>boolean</i>	

For a better readability, the following table shows separately the ToolAgent indication and other possible attributes that must be specified as TWE in the metatool definition. All values of the attributes are string type.

Template type	Metatool attribute	Metatool value	
createCard	ToolAgentClass ClassName	org.cmdbuild.shark.toolagent.CreateCardToolAgent [Class Name]	
createReport	ToolAgentClass Type Code Format	org.cmdbuild.shark.toolagent.CreateReportToolAgent custom [Report code] pdf or csv	
executeFunction	ToolAgentClass Procedure or CursorProcedure	org.cmdbuild.shark.toolagent.ExecuteStoredProcedureToolAgent [PostgreSQL function name with return single value] [PostgreSQL function name with return multiple value]	
startProcess	ToolAgentClass ProcessClass Complete	org.cmdbuild.shark.toolagent.ProcessStartToolAgent [Class Name] 1 (to advance the process to the following activity) or 0 (to stop the process on the first activity)	
updateCard	ToolAgentClass	org.cmdbuild.shark.toolagent.UpdateAttributeToolAgent	
updateProcess	ToolAgentClass ProcessClass Complete	org.cmdbuild.shark.toolagent.ProcessUpdateToolAgent [Class Name] 1 (to advance the process to the following activity) or 0 (to stop the process on the first activity) If the process is "Suspended", the resumeProcess method must be carried out in advance.	

## **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.