

Version

2.5



» Connectors Manual

November 2017

Author Tecnoteca srl

www.tecnoteca.com

ENG

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 |
| Available documentation..... | 4 |
| Interoperability solutions..... | 6 |
| Feeding the CMDB..... | 6 |
| General characteristics of a connector with external systems..... | 6 |
| Available solution in the CMDBuild system..... | 7 |
| Wizard Connector..... | 8 |
| General Information..... | 8 |
| Operation modes..... | 8 |
| Configuration modes..... | 8 |
| License to use..... | 9 |
| Advanced Connector..... | 10 |
| General Information..... | 10 |
| Operation modes..... | 10 |
| Configuration modes..... | 12 |
| Execution..... | 12 |
| Limitations of use..... | 13 |
| Examples of connectors..... | 14 |
| General Information..... | 14 |
| OCS Inventory..... | 15 |
| Active Directory..... | 19 |
| VMware VCenter..... | 21 |
| LLDP Devices..... | 23 |
| Zabbix..... | 25 |
| Nagios – GroundWork – NetEye..... | 26 |
| Archi..... | 31 |
| APPENDIX: Glossary..... | 37 |

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.

Available documentation

This manual is aimed to illustrate the configuration of connectors through which it is possible to

synchronize information, managed in external applications and databases, in CMDBuild, in the different technical modes currently available.

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

- system overview ("Overview Manual")
- system usage ("User Manual")
- system administration ("Administrator Manual")
- installation and system management ("Technical Manual")
- workflow configuration ("Workflow Manual")
- webservice details and configuration ("Webservice Manual")

Interoperability solutions

Feeding the CMDB

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

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

Therefore the configuration of connectors through external systems becomes important in order to sync in CMDBuild (the central CMDB system) data that are mainly managed ("master") on other specialist applications, among these:

- systems of automatic inventory, for automatic comparison of the technical data of the assets and of the installed software and management of the observed differences (OCS Inventory or other products)
- control systems (virtualization, etc.) to extract information about the current configuration of the IT infrastructure
- tracking systems of information regarding the execution of server services and applications
- monitoring systems, to detect system failures and start workflows of Incident Management
- LDAP directory as a repository for the archive of the staff
- HR systems, as alternative solution to receive changes to the staff list
- ERP systems to receive administration data (sources, suppliers, etc.)

General characteristics of a connector with external systems

Generally speaking, a connector with external systems is allowed to:

- contact the system of interest (automatic inventory tool, LDAP directory, etc.), through one of the possible channels of communication (direct access to DB, webservice SOAP, REST webservice, other API, structured e-mails, csv files, etc)
- access to the specific information to be synchronized (list of assets, list of people, etc.), extracted by specific SQL query or SOAP calls or API, etc.
- apply the appropriate criteria of "mapping" between the source information and the corresponding information to be inserted in CMDBuild, considering:
 - key: source data <=> key: CMDBuild class
 - entity/ data source information (such as table/ database column or item/property returned from a SOAP method) <=> entity/target information (attribute/class of CMDBuild)
 - any application logic element to be applied within the framework of the "mapping" (e.g. application of rules of aggregation/disaggregation of information, performance of operations such as log or sending notifications, posting information on the external system, etc.)

A connector will then be limited to synchronize out-of-date data by comparison to the external "master" system, or it might submit the need for change in that information to an authorization

process of Change Management (suggested for differences that are classified as "critical").

In a same instance CMDBuild can be useful to enable multiple connectors with several external systems, each with its own processing and scheduling rules.

Avilable solution in the CMDBuild system

CMDBuild offers two different solutions to enable and schedule connectors to synchronize data with external systems:

- a first solution in CMDBuild, called Wizard Connector, fully configurable from the user interface through "wizard". It can be used in the simplest cases where the rules of "mapping" do not require a specific application logic
- a second solution, external to CMDBuild and called Advanced Connector, based on a framework realized in Java and programmable through the Groovy scripting language. It can be used in the most complex cases (different data sources, complex application logic, large amount of data)

In the previous versions of this manual there was another solution, called Basic Connector, which could be configured through the transformation language XSLT / XPATH and based on the access directed at external data sources. This solution is not supported by Tecnoteca anymore.

In any time you can give up the libraries and the base functions provided by the Advanced Connector and realize from the beginning interoperability solutions external to CMDBuild, which implement the desired logics by accessing its database through the public webservices REST and SOAP.

Wizard Connector

General Information

The version of the connector, based on the internal wizard, allows to resolve the simplest cases of data synchronization, in which the "mapping" rules do not require a specific application logic.

Since it is fully configurable from the user interface, it also has the advantage that it can be used immediately, without the need for complex activation procedures.

On the other hand the Wizard Connector has some limitations:

- it can handle only simple mapping rules
- it only accesses external data sources like relational database (PostgreSQL, MySQL, Oracle, SQLServer)
- each instance can synchronize only a CMDBuild class and detail top-level classes associated to the domains 1: N
- it does not contemplate the possibility of starting authorization processes of Change Management for the resolution of the variations observed on imported data

Operation modes

The Wizard Connector is implemented as a function that is internal to the CMDBuild application and it is based on the following criteria:

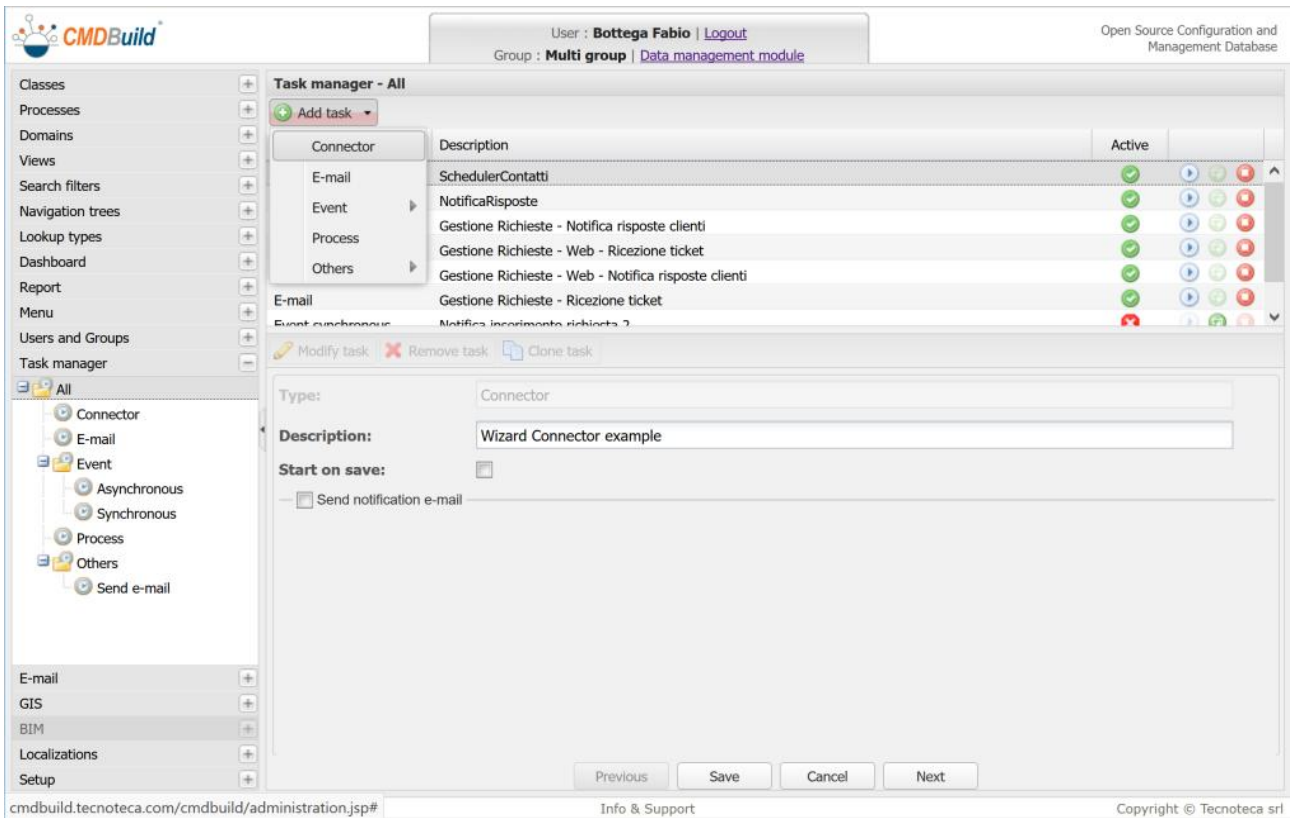
- it must have access to an external data source of the expected typology
- in the external data source a subset of information of interest of CMDBuild is provided, already organised in order to be directly "mapped" on classes/attributes of CMDBuild (tables or views or other data structures for data sources that are different from relation databases)
- CMDBuild administrator, from the user interface, configures the criteria of information "mapping"
- a task will run at defined intervals of time during the process of scheduling:
 - it will read the information from the configured views
 - it will compare the information with the corresponding information in CMDBuild
 - it will perform updating operations
 - at the end, it will send a notification e-mail or it executes a script, according to the required configuration modalitie

Configuration modes

The Wizard Connector is one of the different tasks managed in CMDBuild with the Task Manager.

It is configured through a proper wizard requiring all the necessary information to schedule and execute a task.

Here's the initial page of the wizard.



For detailed information about the connector configuration mode, you can refer to the Administrator Manual.

License to use

Since this is a UI features of CMDBuild, the same open source license AGPL 3.0 of the basic application must be applied.

Advanced Connector

General Information

The Advanced Connector is also composed by an application external to CMDBuild, even if it is "schedulable" both through the specific system service and through the Task Manager internal to CMDBuild.

It is based on a predefined Java framework that implements the core logics, useful for synchronization activities, and requires that the specific behavior of each instance of the connector is configured through the Groovy scripting language.

Even the Advanced Connector can be configured to interact with a management process (normally a Change Management workflow in ITIL) which submits all the potentially critical changes to the approval of a responsible user (usually the Change Manager).

It is the most complete solution among those available with CMDBuild, and it guarantees:

- the ability to work with any external application with an accessible database or webservices/API or producing text files/e-mails containing the information to be synchronized
- possibility of contextual synchronization of complex data structures
- good performance with large amounts of data
- unlimited customizations, since the connector is configurable through a standard programming language

The only limitation of the Advanced Connector is that it requires programming skills for configuring each different instance.

Operation modes

The Advanced Connector consists of a series of modules that are executed in sequence according to what has been defined in the configuration file:

```
{CONNECTOR_HOME}/conf/connector.properties
```

These modules can:

- perform input/output operations by reading from the foreign data source and writing on the CDMBuild database through its webservices
- send email notification with attached report
- process the information read in input in order to normalize the data or to make them compatible with the data structure of CMDBuild

The procedures that can be carried out by each module are described in the configuration files that are defined in the folder:

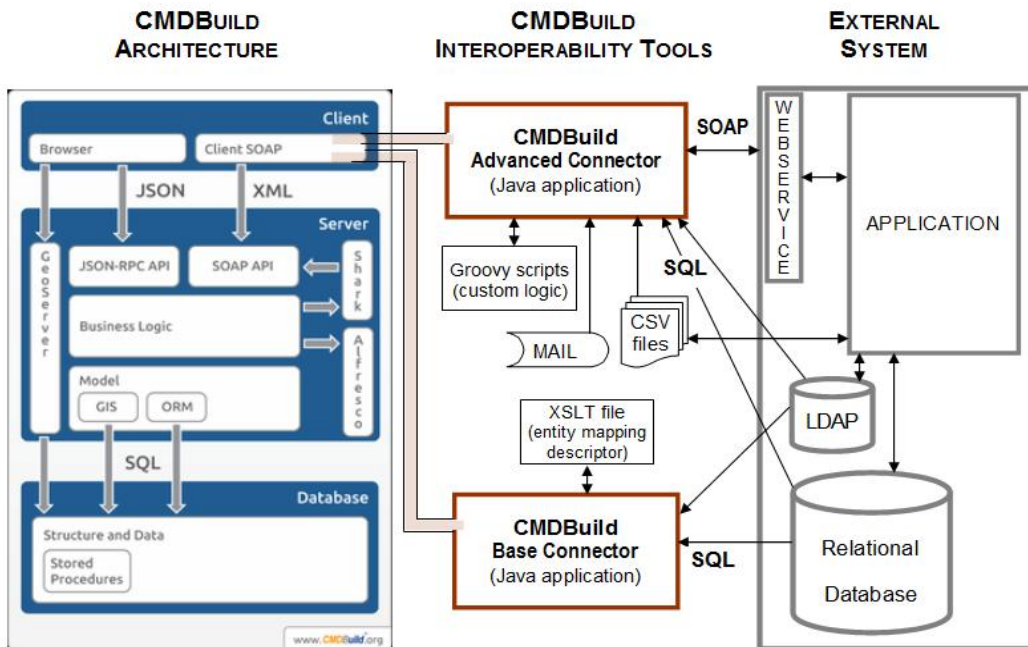
```
{CONNECTOR_HOME}/conf
```

Every configuration file consists of a script in the Groovy language that is suitably interpreted by the connector that will then perform the sequence of instructions defined in the script itself.

In detail the modules to be configured are the following:

1. module scheme: connector-schema.groovy
It defines the scheme of the data structure of CMDBuild on which the connector keeps the data read from the external source updated.
2. sql module: connector-sql.groovy
It is the input module that reads the information from the external data source and runs a first normalization of the read information. This module also performs a check on the read data, verifying any anomaly or inconsistency that requires the sending of a notification.
3. modify on remove module: connector-mor.groovy
It is the output module that simply sets the connector to modify also the information that is no longer present in the external database (DBO.Data) and that would normally be removed from the CMDBuild database (in this way the information is just set as inactive and not removed from the data structure).
4. mail module: connector-mail.groovy
It is the module that allows you to read a mailbox and to send email notifications as needed.
5. There is also an additional output module (implicit) that defines the end-point of the webservices of CMDBuild, used by the connector to write the information on CMDBuild, in the tables appropriately defined in the "schema" form.

A diagram of how the Advanced Connector and the Basic Connector interact with CMDBuild and with the external system/application with which the data synchronization is required is shown below.



Configuration modes

File system structure

The connector can be placed in a folder as required in the filesystem of the server, which from this point forward will be referred to with the name of `${CONNECTOR_HOME}`.

The default filesystem structure is shown below.

```
-- ${CONNECTOR_HOME}
|
|--bin
|   |--connector.sh
|
|--config
|   |--connector-extension.properties
|   |--connector.properties
|   |--connector-sql-context.xml
|   |--log4j.properties
|   |--template
|   |-- ...
|
|-- lib
|   |--...
```

The bin/connector.sh file

It is the script needed to run the application. It contains the configuration and default parameters needed for the performance.

The config/connector.properties file

The file describes these modules and their organization (chains):

```
modules.schema=
modules.in.source=
modules.in.cmdbuild=ws
modules.out=
module.ws.classname=org.cmdbuild.connector.ws.WsModule
```

At the very least a scheme module and a form of input and output module must be specified. By default the necessary module for the communication with CMDBuild via web service is already present.

The directory config/template

This directory contains the templates of the configuration files for all the available modules.

The module configuration files

For each added module, within the directory `${CONNECTOR_HOME}/config`, the files “connector-`${MODULE_NAME}.properties`” and “connector-`${MODULE_NAME}.groovy`”, where “`${MODULE_NAME}`” represents the name of the module as specified within the file.

Execution

Once the above-described files are edited, the connector can be started by running the script:

```
connector.sh
```

The execution of the connector can be programmed both via the Task Manager internal to CMDBuild and through the system scheduler (the “cron” service in Unix/Linux).

For the first case, please refer to the Administrator Manual.

For the second case, it is advisable to insert into the `/etc/cron.d/` folder an additional file, such as "connectors", to specify, in accordance with the cron syntax, when to execute the different connectors.

An example of scheduling is the following:

```
00 20 * * * root sh ${CONNECTOR_HOME}/bin/connectors.sh
```

Please remember to restart the cron service once the schedule is inserted.

Limitations of use

The Advanced Connector is made available with non-open source license, which allows only those who have signed with Tecnoteca srl a maintenance service for the CMDBuild application to use the service, and only until that service is active.

However, the Advanced Connector is provided with the source code of all its components to those who have signed with Tecnoteca srl a maintenance service for the CMDBuild application to use the service

Examples of connectors

General Information

Being CMDBuild a totally customizable system, it is not possible to create standard connectors that work on system whose configuration is not known.

Since the aim of the connector is to synchronize information between the data model of the external system and the data model of CMDBuild, the total freedom in the configuration of CMDBuild requires to proceed with the connector configuration.

Similarly any approval mechanism of the most critical Changes allowed by the logic of the connector must be adapted to the corresponding workflows configured in CMDBuild.

For this reason the three types of connectors, described in the previous pages, contain configuration options, even if with different degrees of difficulty (UI => editing of text files => writing of programming code).

In the subsequent pages some already implemented connectors are described, that must therefore be taken as sample implementations and not as ready-to-use tools.

The configuration criteria of the connector with OCS Inventory, that is also available as a downloadable file from the site of CMDBuild, are described in more detail.

OCS Inventory

General Information

The manual control and update of the changes in the inventory of IT assets require important resources that are not always available.

It is therefore suggested to automate the synchronization of that information through the use of an automatic inventory tool and, in particular, OCS Inventory (<http://www.ocsinventory-ng.org/en/>) is the most suggestable product, considering both its technical features and the open source license.

OCS Inventory allows you to:

- activate "agent" programs on the computers to be controlled (Windows, Linux or Mac OS)
- collect information about BIOS, processor, RAM, input devices, controller, peripherals, additional cards, network settings, operating system, software applications, etc.
- periodically send to the server that information in the form of XML files
- store that information in a MySQL relational database
- consult and modify that information from a proper management web application

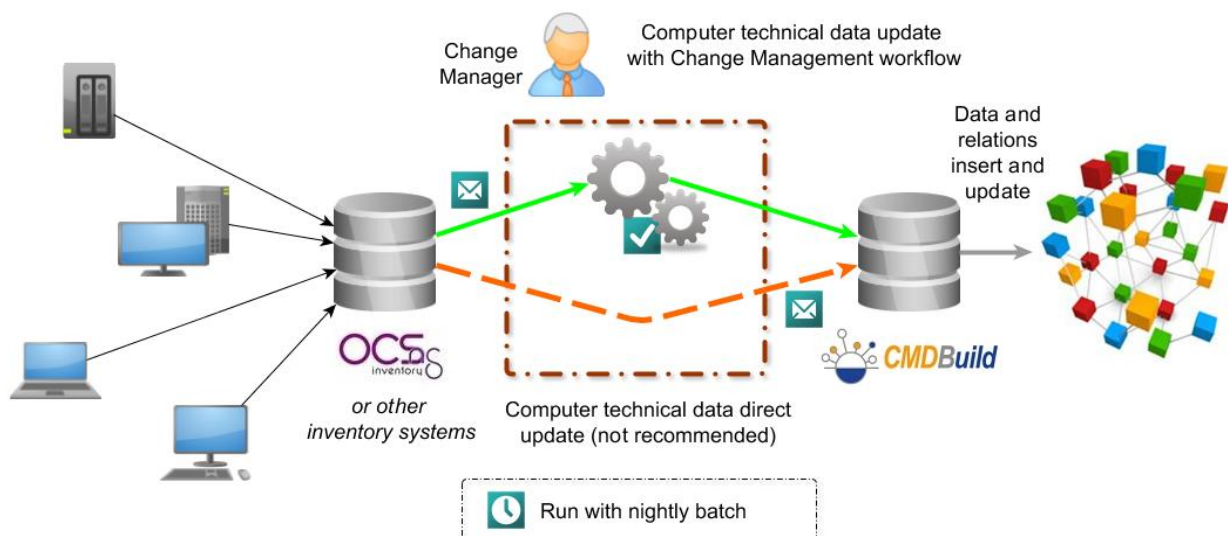
CMDBuild Connector

The aim of the connector is the synchronization of the computer files registered in CMDBuild with data available in the OCS Inventory database.

In some implementations (like the one included in CMDBuild READY2USE) the connector synchronizes also the instances of the "controlled" software (where you want to verify the licenses) detected by the OCS Inventory with the data available in the CMDBuild database.

At the end the connector can send a report with the list of the changes applied in the CMDB.

A connector scheme is shown below.



Available attributes and examples of "mapping"

The list of the entities and information gathered from OCS Inventory, among which those that will be "mapped" in CMDBuild, and possibly on which classes and attributes (last column), must be identified, is shown below.

| OCS Inventory | CMDBuild I/F view | Description | Example | CMDBuild | |
|---|---------------------|--|--|--|--|
| Hardware.Id AccountInfo.Tag -- Hardware.Name Hardware.Workgroup Hardware.UserDomain -- -- Hardware.OSName Hardware.OSVersion Hardware.OSComments Hardware.ProcessorT Hardware.ProcessorS Hardware.ProcessorN Hardware.Memory Hardware.WinProdKey | CMDBuild_device | Id Tag Serial Name Workgroup UserDomain Manufacturer ProductName OSName OSVersion OSComments ProcessorS ProcessorN Memory WinProdKey | Id Key Serial number Activity name Working Group Domain Producer Product name Operating system name Operating system version Service Pack Processor speed Number of processors RAM Memory Product Keyo | 2 lisa HUB5390NQK Kelly Tecnoteca Kelly Hewlett-Packard HP Compaq nc6120 (PN936AV) Microsoft Windows XP Professional 5.1.2600 1832 1 512 | |
| Bios.Hardware_Id Bios.SManufacturer Bios.SModel Bios.SSN Bios.Type Bios.BManufacturer Bios.BVersion Bios.BDate | CMDBuild_bios | Device_Id SManufacturer SModel SSN Type BManufacturer BVersion BDate | Reference machine MotherBoard Manufacturer MotherBoard Model Serial number M.B. Chassis type Producer Version Signing date | 2 VIA K7Upgrade-600 SYS-1234567890 Desktop American Megatrends Inc. AMIINT-10-SMBiosVersion: P1.50 N/A | |
| Controllers.Hardware_Id Controllers.Manufacturer Controllers.Name Controllers.Caption Controllers.Description Controllers.Version Controllers.Type | CMDBuild_controller | Device_Id Manufacturer Name Caption Description Version Type | Reference machine Producer Activity name Title Description Version Type | 2 VIA Technologies, Inc. Controller IDE VIA Bus Master Controller IDE VIA Bus Master Controller IDE VIA Bus Master N/A IDE Controller | |
| Drives.Hardware_Id Drives.Letter Drives.Type Drives.FileSystem Drives.Total Drives.Free Drives.NumFiles Drives.VolumN | CMDBuild_drive | Device_Id Letter Type FileSystem Total Free NumFiles VolumN | Reference machine Letter Type File System Total space Free space Number of files Volume name | 2 C:/ Hard Drive NTFS 39997 15708 0 | |
| Inputs.Hardware_Id Inputs.Type Inputs.Manufacturer Inputs.Caption Inputs.Description Inputs.Interface Inputs.PointType | CMDBuild_input | Device_Id Type Manufacturer Caption Description Interface PointType | Reference machine Type Producer Title Description Interface Pointer type | 2 Pointing (Standard System Devices USB Human Interface Device USB Human Interface Device USB N/A | |

| | | | | | |
|--|-------------------|--|---|--|--|
| Memories.Hardware_Id Memories.Caption Memories.Description Memories.Capacity Memories.Purpose Memories.Type Memories.Speed Memories.NumSlots | CMDBuild_memory | Device_Id Caption Description Capacity Purpose Type Speed NumSlots | Reference machine Title Description Dimension Purpose Type Speed Number of slots | 2 Physical memory DIMM1 (Other ECC) 512 Reserved Unknown N/A 1 | |
| Modems.Hardware_Id Modems.Name Modems.Model Modems.Description Modems.Type | CMDBuild_modem | Device_Id Name Model Description Type | Reference machine Activity name Model Description Type | | |
| Monitors.Hardware_Id Monitors.Manufacturer Monitors.Caption Monitors.Description Monitors.Type Monitors.Serial | CMDBuild_monitor | Device_Id Manufacturer Caption Description Type Serial | Reference machine Producer Title Description Type Serial number | 2 Unknown manufacturer code ACR AL1716 RGB color | |
| Networks.Hardware_Id Networks.Description Networks.Type Networks.TypeMIB Networks.Speed Networks.MCAddr Networks.Status Networks.IPAddress Networks.IPMask Networks.IPGateway -- Networks.IPSubnet Networks.IPDHCP | CMDBuild_network | Device_Id Description Type TypeMIB Speed MACAddr Status IPAddress IPMask IPGateway IPDNS IPSubnet IPDHCP | Reference machine Description Type MIBType Speed Macaddress State IP Address IP Mask IP Gateway IP DNS IP subnet IP DHCP server | 2 Fast Ethernet VIA compat. card Ethernet EthernetCsmacd 100 Mb/s 00:0B:6A:AE:DE:6B Up 192.168.2.199 255.255.255.0 192.168.2.1 192.168.2.100 192.168.2.0 255.255.255.255 | |
| Ports.Hardware_Id Ports.Type Ports.Name Ports.Caption Ports.Description | CMDBuild_port | Device_Id Type Name Caption Description | Reference machine Type Activity name Title Description | 2 Serial Communications port (COM1) Communications port (COM1) Communication port | |
| Slots.Hardware_Id Slots.Name Slots.Description Slots.Designation Slots.Purpose Slots.Status Slots.PShare | CMDBuild_slot | Device_Id Name Description Designation Purpose Status PShare | Reference machine Activity name Description Code Use State Sharing | 2 System Slot System Slot PCI1 OK 1 | |
| Softwares.Hardware_Id Softwares.Publisher Softwares.Name Softwares.Version Softwares.Folder Softwares.Comments Softwares.FileName Softwares.FileSize Softwares.Source | CMDBuild_software | Device_Id Publisher Name Version Folder Comments FileName FileSize Source | Reference machine Producer Activity name Version Folder Comment File name File size Source | 2 OpenOffice.org OpenOffice.org 2.0 2.0.9044 OpenOffice.org 2.0 (en-US)(Build:9044) N/A 0 1 | |

| | | | | | |
|--|------------------|---|---|---|--|
| Sounds.Hardware_Id Sounds.Manufacturer Sounds.Name Sounds.Description | CMDBuild_sound | Device_Id Manufacturer Name Description | Reference machine Producer Activity name Description | 2 Silicon Integrated Systems [SiS] AC'97 Multimedia audio controller rev a0 | |
| Storages.Hardware_Id Storages.Manufacturer Storages.Name Storages.Model Storages.Description Storages.Type Storages.DiskSize | CMDBuild_storage | Device_Id Manufacturer Name Model Description Type DiskSize | Reference machine Producer Activity name Model Description Type Disk size | 2 (standard disk unity) Generic Flash Disk USB Device //./PHYSICALDRIVE1 Disk unity Removable media other than floppy 117 | |
| Videos.Hardware_Id Videos.Name Videos.Chipset Videos.Memory Videos.Resolution -- | CMDBuild_video | Device_Id Name Chipset Memory Resolution RefreshRate | Reference machine Activity name Chipset Memory Resolution Refresh frequency | 2 RADEON 9200 PRO Family (Microsoft C.) RADEON 9200 PRO AGP (0x5960) 128 1280 x 1024 64 | |

It's important to carefully consider which attribute is to be used as computer's unique field: a number of internal asset (specify "one-off" with the appropriate flag "tag" during the OCS-Agent installation), the MAC Address of the network adapter (not necessarily invariant in case of substitution), etc.

Active Directory

General Information

Despite being a typology of information normally treated by dedicated systems, it is generally useful to have the staff stored in CMDBuild too, thus enabling to correlate, for example, computers to their assignees, activities of assistance to applicants, rooms to their occupants, etc.

It is therefore useful to activate a connector for the automatic synchronization of data between the master archive, which is very often managed through the Active Directory service, and CMDBuild.

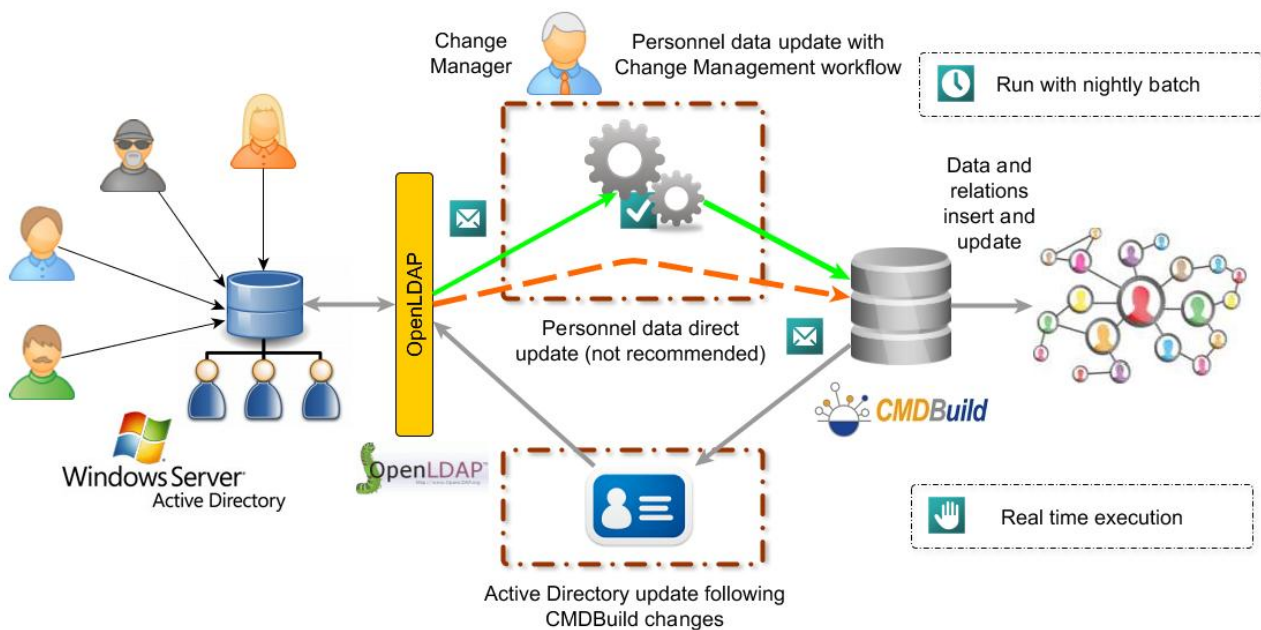
CMDBuild Connector

The aim of the connector is the synchronization of the employees registry registered in CMDBuild with data available in the LDAP database.

In some implementations the connector synchronizes from LDAP the list of the offices related to the employees (organizational chart).

At the end the connector can send a report with the list of the changes applied in the CMDB.

A connector scheme is shown below.



Available attributes and examples of "mapping"

BELOW YOU CAN FIND A SAMPLE OF MAPPING FROM THE LDAP CONNECTOR INCLUDED IN THE APPLICATION CMDBUILD READY2USE:

| ACTIVE DIRECTORY FIELD | CMDBUILD CLASS | CMDBUILD ATTRIBUTE | KEY |
|------------------------|------------------|--------------------|-----|
| uid | InternalEmployee | Code | yes |
| givenName | InternalEmployee | FirstName | |
| sn | InternalEmployee | LastName | |

| | | | |
|-----------------|------------------|-------|--|
| mail | InternalEmployee | Email | |
| telephonenumber | InternalEmployee | Phone | |
| 'Active' | InternalEmployee | State | |

VMware VCenter

General Information

VMware is one of the leading virtualization environments, while VCenter is the tool provided by VMware for the management of the infrastructure.

The connector described in this paragraph is intended to synchronize in CMDBuild the up-to-date inventory of the virtual servers used in the VMware Infrastructure, with:

- information regarding the assigned resources (operating system, RAM, CPU, etc)
- information about the status of the server
- other custom information, managed by each organization by using the graphic interface of the application (technical coordinator, cluster, etc)

It is true that the automatic inventory tools (including OCS Inventory) are able to detect some technical information about the virtual servers, but such information is not as complete as those provided by each individual virtualization system.

The connector can synchronize in CMDBuild all the virtual machines connected to VMware VCenter, regardless of their status (on, off, etc).

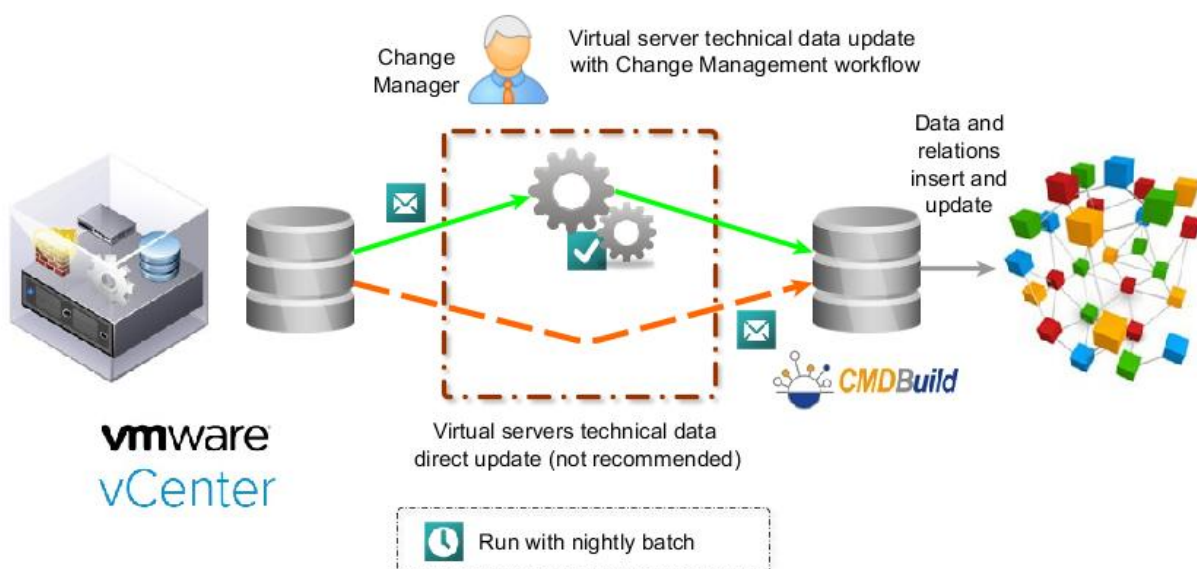
CMDBuild Connector

The aim of the connector is the synchronisation of information related to those servers supplying resources to the virtualisation environment, to virtual servers managed in the infrastructure and to those clusters containing the virtual servers.

At the end the connector can send a report with the list of the changes applied in the CMDB.

The connector included in CMDBuild `READY2USE` accesses the VMware VCenter through the APIs provided by the VMware Infrastructure (VI) SDK 2.5.0 (`com.vmware.vim25` e `com.vmware.vim25.mo`).

A connector scheme is shown below.



Available attributes and examples of "mapping"

Below you can find a sample of mapping from the VMware VCenter connector included in the application CMDBuild Ready2Use:

| VMWARE INFORMATION | CMDBUILD CLASS | CMDBUILD ATTRIBUTE | KEY |
|---|------------------|--------------------|-----|
| Cluster description | Cluster | Description | yes |
| Cluster name | Cluster | Name | |
| Virtual server description | VirtualServer | Description | yes |
| Virtual server Hostname | VirtualServer | Hostname | |
| Virtual server RAM | VirtualServer | RAM | |
| CPU number of the virtual server | VirtualServer | TotalCPUs | |
| Operating system of the virtual server | VirtualServer | OSVersion | |
| Notes about the virtual server | VirtualServer | Notes | |
| Status of the virtual server | VirtualServer | VSState | |
| Execution cluster of the virtual server | VirtualServer | ExecutionCluster | |
| Physical server description | PhysicalServer | Description | yes |
| Physical server hostname | PhysicalServer | Hostname | |
| Physical server brand | PhysicalServer | Brand | |
| Physical server model | PhysicalServer | Model | |
| Physical server OS | PhysicalServer | OS | |
| Physical server OS version | PhysicalServer | OSVersion | |
| Physical server RAM | PhysicalServer | RAM | |
| Physical server HD size | PhysicalServer | HDSize | |
| Physical server CPU | PhysicalServer | CPU | |
| Physical server served cluster | PhysicalServer | ServedCluster | |
| Network interface description | NetworkInterface | Description | yes |
| Network interface code | NetworkInterface | Code | |
| Network interface name | NetworkInterface | Name | |
| Network interface IPAddress | NetworkInterface | IPAddress | |
| Network interface hardware | NetworkInterface | Hardware | |

LLDP Devices

General Information

The most recent network devices support the discovery and communication LLDP and SNMP.

By using these protocols you can know:

device hostname

the list of the devices connected to each device port

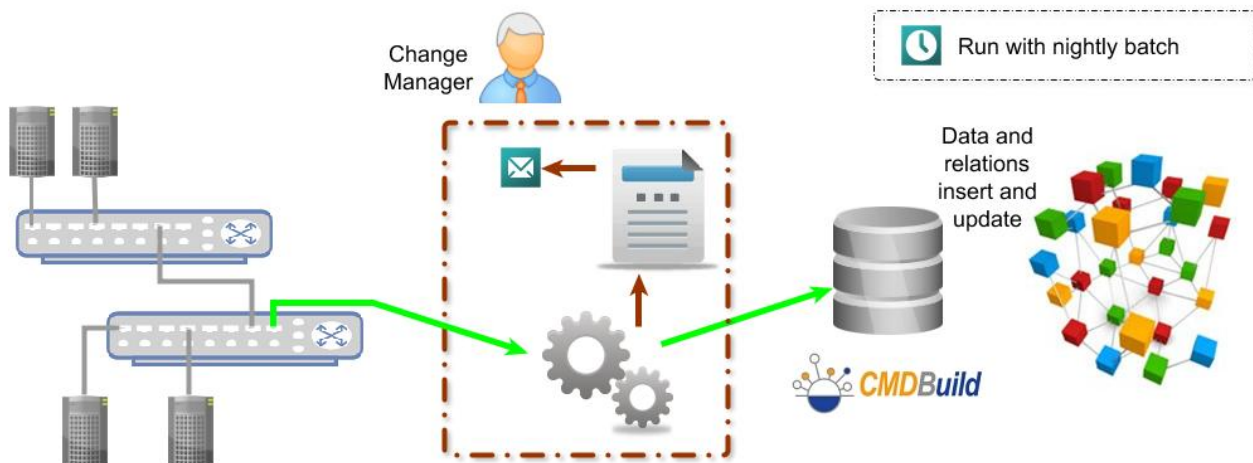
CMDBuild Connector

The aim of the connector is the synchronisation of information related to the network devices and to the connections with other network devices, servers, IP phones, etc.

The connector included in CMDBuild Ready2Use connects to a device port (switch or router) which is set as discovery origin, and which executes the analysis on all connected items by using a pre-connector which queries the devices and produces intermediate CSV files.

At the end the connector can send a report with the list of the changes applied in the CMDB.

A connector scheme is shown below.



Available attributes and examples of "mapping"

Below you can find a sample of mapping from the VMware VCenter connector included in the application CMDBuild Ready2Use:

| LLDP / SNMP (OID) INFORMATION | CMDBUILD CLASS | CMDBUILD ATTRIBUTE | KEY |
|-------------------------------|----------------|--------------------|-----|
| iso.0.8802.1.1.2.1.4.1.1.9 | Hardware | Hostname | yes |
| iso.0.8802.1.1.2.1.3.3.0 | NetworkDevice | Hostname | yes |
| iso.0.8802.1.1.2.1.3.4.0 | NetworkDevice | Description | |
| iso.0.8802.1.1.2.1.3.7.1.3 | NetworkDevice | PortNumber | |

| | | | |
|------------------------------------|------------------|------------|-----|
| iso.0.8802.1.1.2.1.4.1.1.8 | NetworkInterface | Name | yes |
| iso.0.8802.1.1.2.1.4.2.1.3 | NetworkInterface | IPAddress | |
| iso.0.8802.1.1.2.1.4.1.1.7 | NetworkInterface | MACAddress | |
| Reference to NetworkDevice card | NetworkInterface | Hardware | |
| Reference to NetworkInterface card | NetworkInterface | DstPort | |

Zabbix

General Information

Zabbix is an open source tool used to monitor networks and IT system and to notify any malfunction.

Such notifications are interesting for CMDBuild, which has to gather and manage any interruption in the services supplied to users.

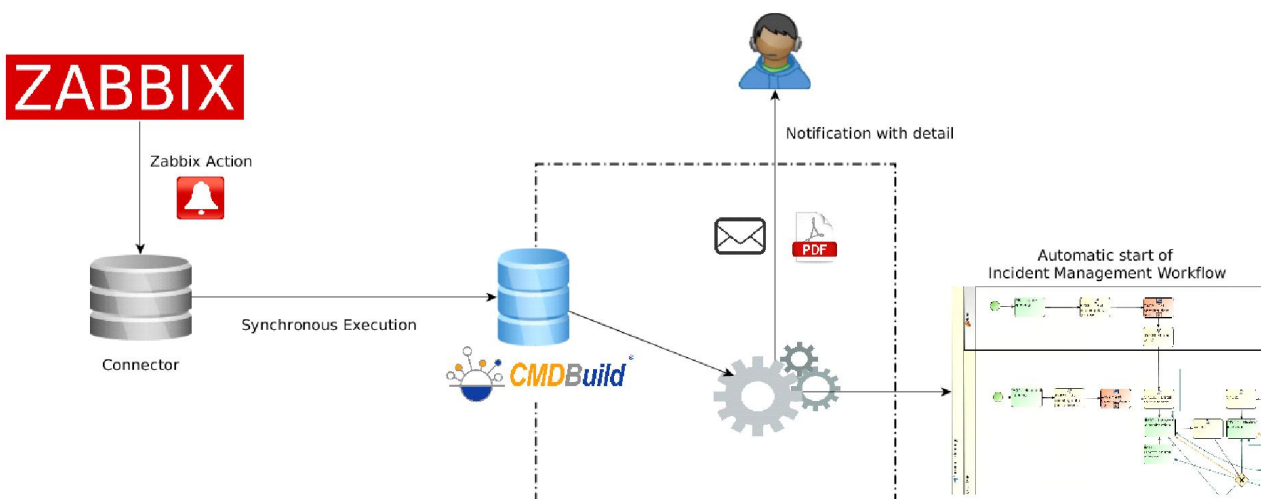
CMDBuild Connector

The aim of the connector included in CMDBuild Ready2Use is to:

- gather notifications through the API 3.0 of Zabbix
- file events in a proper new class of the CMDB
- send technical contact people a summarized report of the involved CIs
- start a new Incident Management process
- close automatically the Incident process if the breakdown is solved before the process is taken in charge

If certain hosts or services depend on others and you do not want to open as many Incidents as the problematic assets but only the "father" one, the dependency among Trigger has to be configured on the Zabbix side.

A connector scheme is shown below.



Nagios – GroundWork – NetEye

General Information

Among the management tools adopted for the control of information systems, a forefront role is played by the solutions for equipment monitoring and management of alarms (we would like to remember, among the open source applications, Nagios, NetEye, GroundWork, etc).

Often, however, the use of these tools does not include input integrations into the company CMDB and the output is limited to only sending notification emails.

As a hint for those who have to configure this kind of connectors, here is shown the description of two actual uses.

Optimize the activities of the Service Desk

The aim of this first solution is to assist the operators in assessing any malfunction and starting, at the same time, the Incident Management processes to solve it.

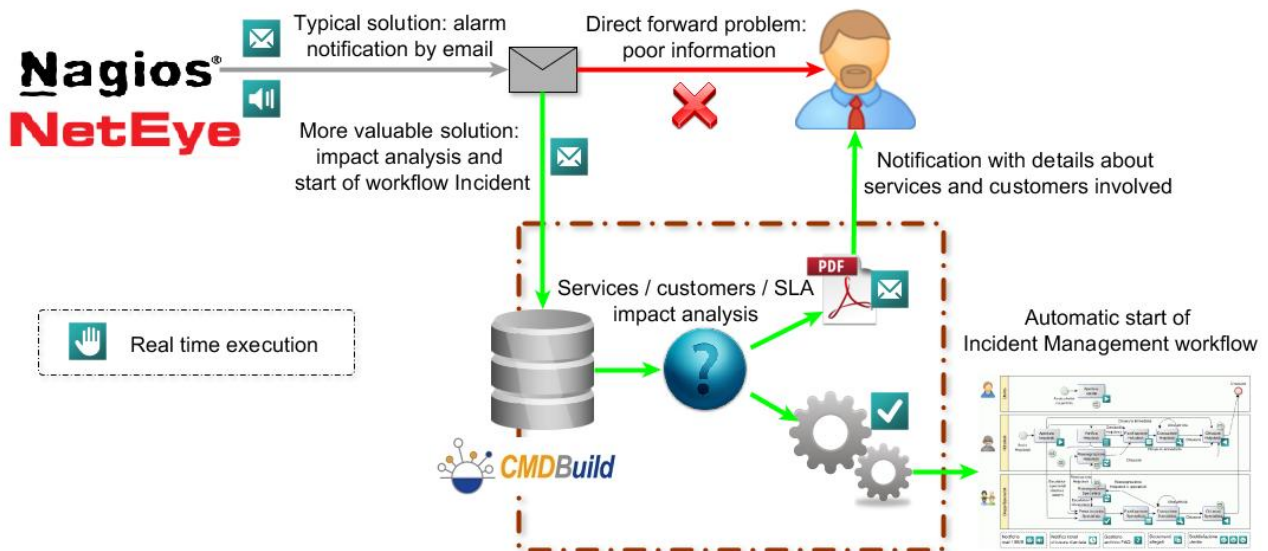
The most common solution provides that the monitoring system will send e-mails such as "Host xy down". In many cases, however, the technician who receives the notification is not able to instantly understand the seriousness of the problem and, for example, if he is outside the company headquarters on the days of availability, he does not have immediate access to the necessary additional information for a deeper and further evaluation.

The connector that is suggested to solve this kind of problems behaves as follows:

- it intercepts the e-mails sent by the monitoring system to the mailboxes (IMAP) of the technical support
- it analyzes the text of the e-mail, which, being produced by the monitoring tool, is normally based on a predefined and configurable pattern, and extracts the relevant information (for example, the seriousness of the problem and the hostname of the server)
- it uses the webservice of CMDBuild to access the CMDB and retrieve the information of interest, that, depending on the completeness of the configured data model, may include:
 - additional technical information on the server involved (memory, operating system, etc.)
 - list of the technical services activated on that server
 - list of the user services involved
 - list of the customers who have subscribed them
 - list of the SLA provided for each customer on a given service
 - critical details
- it produces a report containing the above-listed information and sends it to the technical operator, as attached file to the original e-mail reporting the problem
- it starts an Incident Management workflow in CMDBuild, precompiling the available information
- it records the event of down in a special CMDBuild entity (typically configured as "simple" class), connecting it to the server involved
- it reports a "down" in progress on the server tab involved

The configuration of a connector of this type, with an extensive application logic, requires the use of the Advanced Connector.

A schematic diagram of the solution is shown below:



For further information:

The implementation is described in a case history we have presented at the "CMDBuild Day 2012", held in Bologna, at the Emilia Romagna Regional Council last May 10th 2012.

Click on this link to watch video and slides (in Italian) of the intervention "Monitoraggio dei servizi e gestione degli allarmi con CMDBuild" (Monitoring of services and management of alarms with CMDBuild)

<http://www.cmdbuild.org/it/diffusione/cmdbuild-day/cmdbuild-day-2014/2012>

Optimize the configuration of the monitoring system

The setup activities of a monitoring system are far from being banal, since they have to produce the full list of hosts to be monitored, the services to be monitored on each of them and the dependencies in relation to other hosts on the network (to prevent a breakdown of a device producing hundreds of alarms sent by every other host that depends on it).

Very often such activities are manually carried out by duplicating information that should already be present in the CMDB, and forcing a double update activity in the case of changes to the network configuration (resulting in a high probability of error).

The solution, suggested by an external contributor (see references below), currently in beta version, is to enrich the data model of CMDBuild with additional elements that allow to obtain the automatic export of the configuration files of:

- Nagios / NetEye
- NagVis, plugin that allows to view the status of the checks within maps and diagrams
- Nagios Business Process, plugin that implements two main views:
 - Business View: it aggregates the result of the Nagios checks to view the status of high-level services

- Business Impact Analysis: it allows to perform impact analysis by simulating the down of one or more components and by displaying the effects on high level services

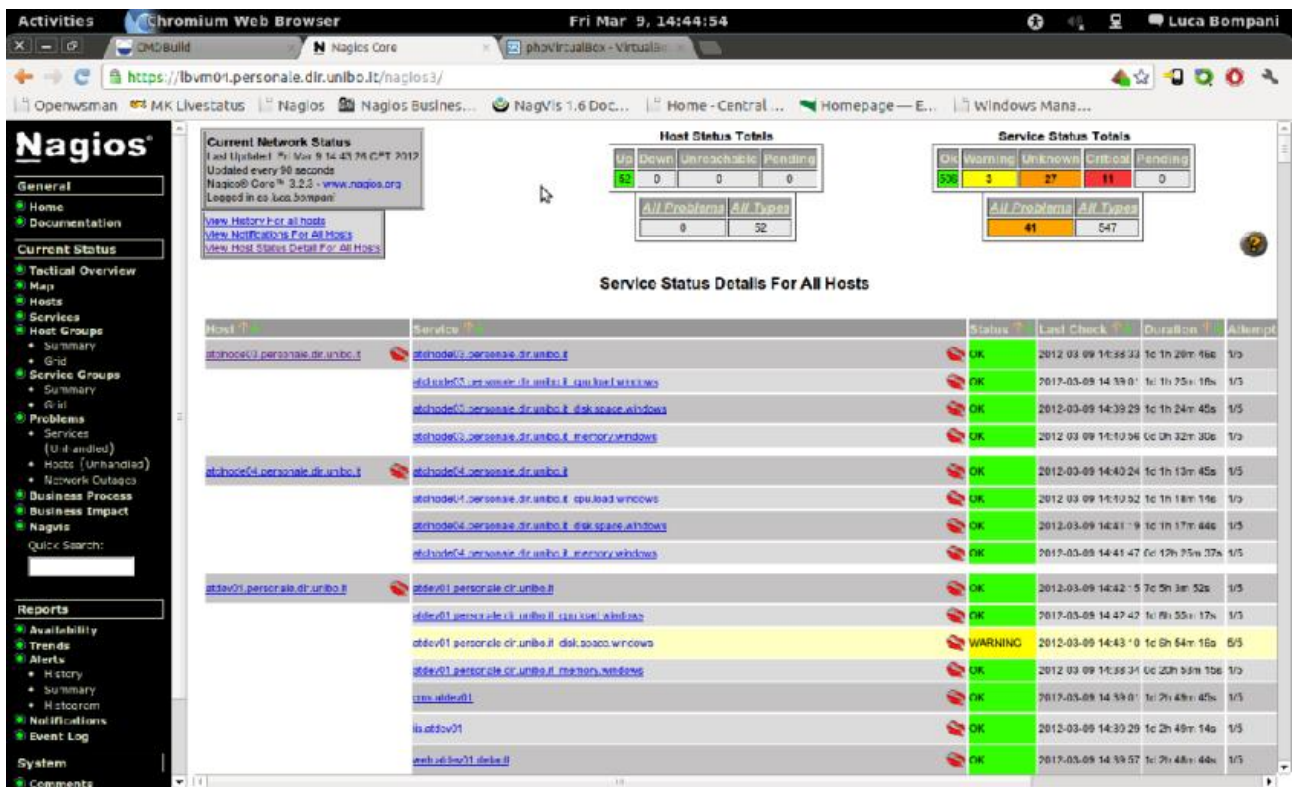
The connector consists of a Java application, different from the Basic Connector and from the Advanced Connector described above, which operates as follows:

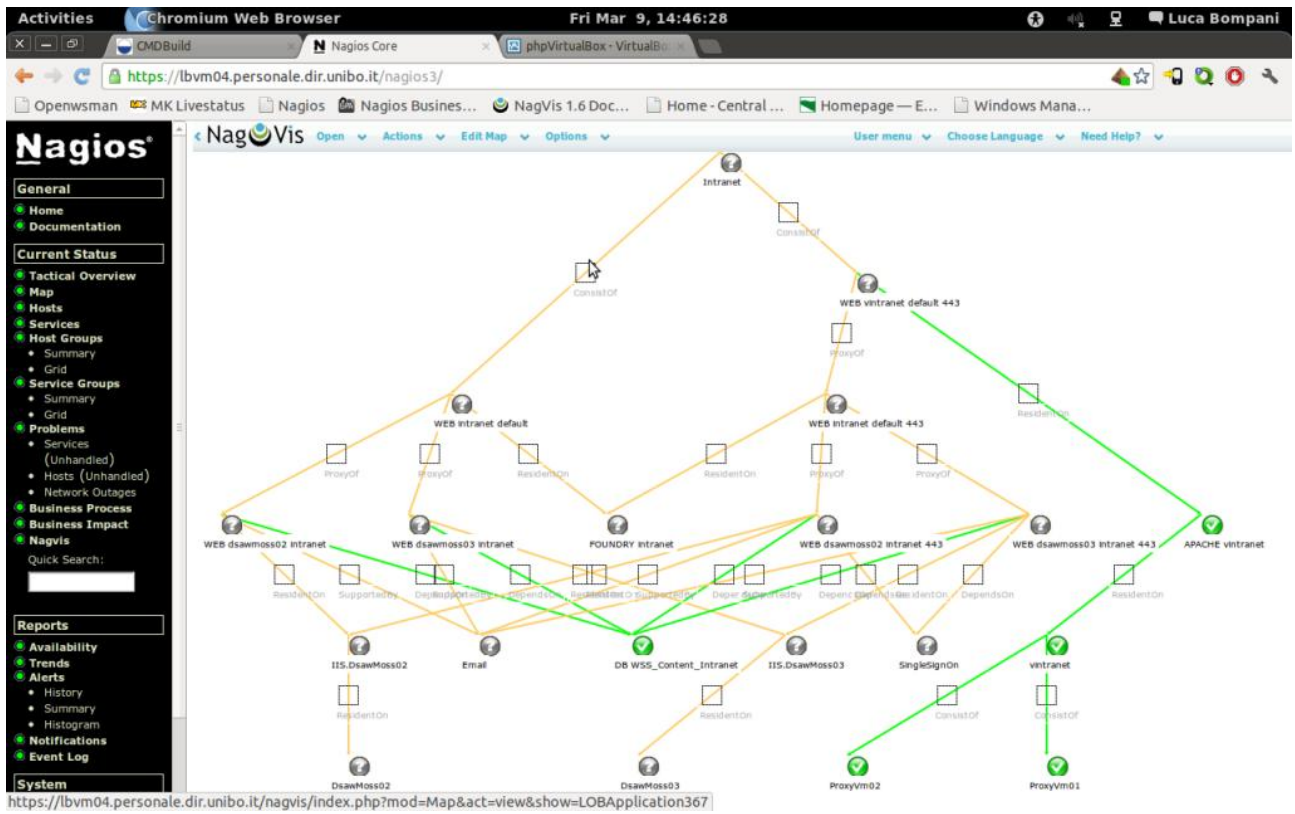
- it accesses the CMDB via a new webservice "CMDBf compliant" (<http://www.dmtf.org/standards/cmdbf>), which is implemented by the same contributor
- it interprets the assets and relations contained in CMDBuild dynamically, being guided by some specially crafted metadata in the form of additional attributes on the involved classes and domains (see below)
- it creates a representation of the in-memory data in the form of a graph, by using special opensource tools
- it uses standard algorithms for the sorting and the visit of the nodes of a graph and to generate the monitoring configuration files

The metadata included in the CMDB in the form of additional attributes are correlated:

- to the classes
 - "host = true" means that for that instance a Nagios host must be created
 - "serviceGroup = true" means for that instance a Nagios serviceGroup must be created
- to the domains
 - "checkAssociation = true" means that the report should be used to associate the Nagios services to the hosts
 - "checkDependency = true" means that the report should be used to generate the dependencies between the Nagios services
 - "checkPropagation = true" means that the report should be used to propagate the checks associated with an asset to another asset
 - "internalDependency = true" means that the report should be used in the generation of NagVis maps
 - "clusterAttribute = Cluster" means that that report should be used to group the relations in clusters: the check on a cluster fails only if all the cluster components fail

Three screenshots regarding, respectively, the integration with Nagios/NagVis NetEye, and NagiosBP are shown below.





Short Summary: All Business Processes

Priority 1
Alerting round the clock (24 x 7)

| Business Process | Status | Status Information |
|------------------|---------|--------------------|
| Intranet | UNKNOWN | |
| Portale Internet | UNKNOWN | |

Priority 2
Alerting Monday to Sunday 7:00 to 22:00

| Business Process | Status | Status Information |
|-------------------|---------|--------------------|
| Academic Projects | UNKNOWN | |
| Corsi di studio | UNKNOWN | |
| Dipartimenti | UNKNOWN | |
| Facolta | UNKNOWN | |
| SpaziVirtual | UNKNOWN | |
| UniboMagazine | UNKNOWN | |

Priority 3
Alerting Monday to Thursday 7:00 to 17:00, Friday 7:00 to 15:00

| Business Process | Status | Status Information |
|------------------|---------|--------------------|
| Atri siti web | UNKNOWN | |

[All Priorities] [Priority 1] [Priority 2] [Priority 3] [Show Traffic Lights]

For further information:

The implementation is described in a case history we have presented at the "CMDBuild Day 2012", held in Bologna, at the Emilia Romagna Regional Council last May 10th 2012.

Click on this link to watch video and slides (in Italian) of the intervention "Un prototipo di integrazione fra CMDBuild e NetEye / Nagios" (A prototype of the integration between CMDBuild and NetEye / Nagios):

<http://www.cmdbuild.org/it/diffusione/cmdbuild-day/cmdbuild-day-2014/2012>

Archi

General Information

TOGAF ¹ is an open framework to support the planning and management of enterprise information architecture, Archimate ² is a modelling language complementary to TOGAF for the description of IT architectures, Archi³ is the open source tool for the design of Archimate models.

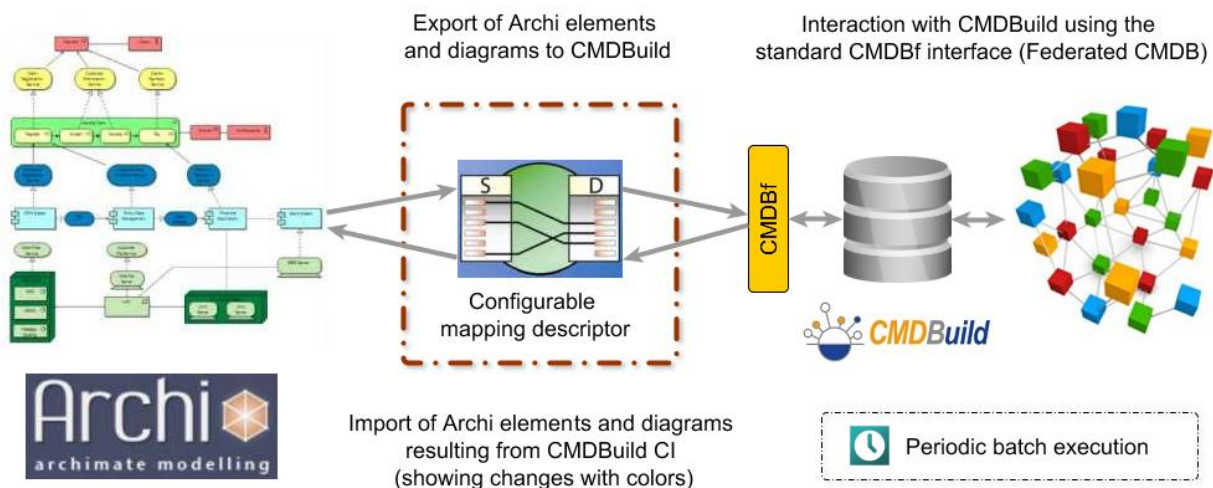
Archi allows to see a graphical overview of CI, customize colors and groupings and graphically navigate relationships, but does not have a structured database to store data.

On the other hand CMDBuild is a complete and structured repository, but does not provide a graphical view of all the infrastructure components through which analyze the impacts and graphically navigate relations.

⁴Starting from these needs two different organizations have decided to implement a plugin that synchronizes CMDBuild data into Archi, and vice versa. Each of the two organizations gained then knowledge of the work done by the other and this has led to join the efforts and release a unique product, created in partnership.

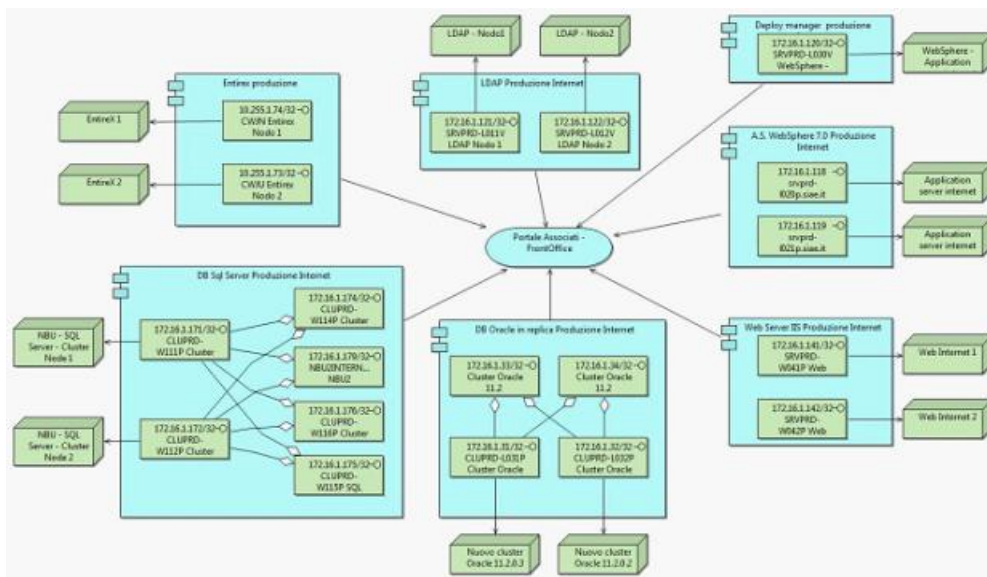
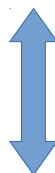
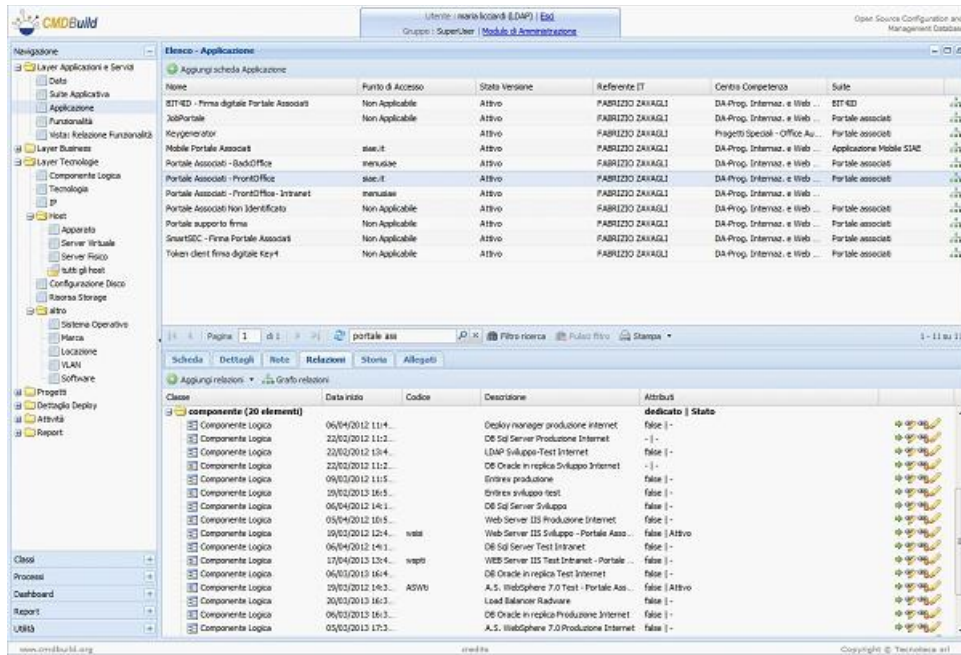
The plugin consists of a Java application directly managed by the two contributors, which can be freely downloaded at this address:

<https://bitbucket.org/bompani/cmdbuild-unibo/downloads>



Here are two screenshots, relevant to show how the same infrastructural information are presented and made available in the two CMDBuild and Archi interfaces.

- 1 <http://www.opengroup.org/togaf/>
- 2 <http://www.opengroup.org/subjectareas/enterprise/archimate>
- 3 <http://archi.cetis.ac.uk/>
- 4 Bologna University and SIAE Rome



For further information:

The implementation is described in a case history we presented at the "TOGAF, Archi and CMDBuild", held in Florence in the Tuscany Regional Council last May 28, 2013.

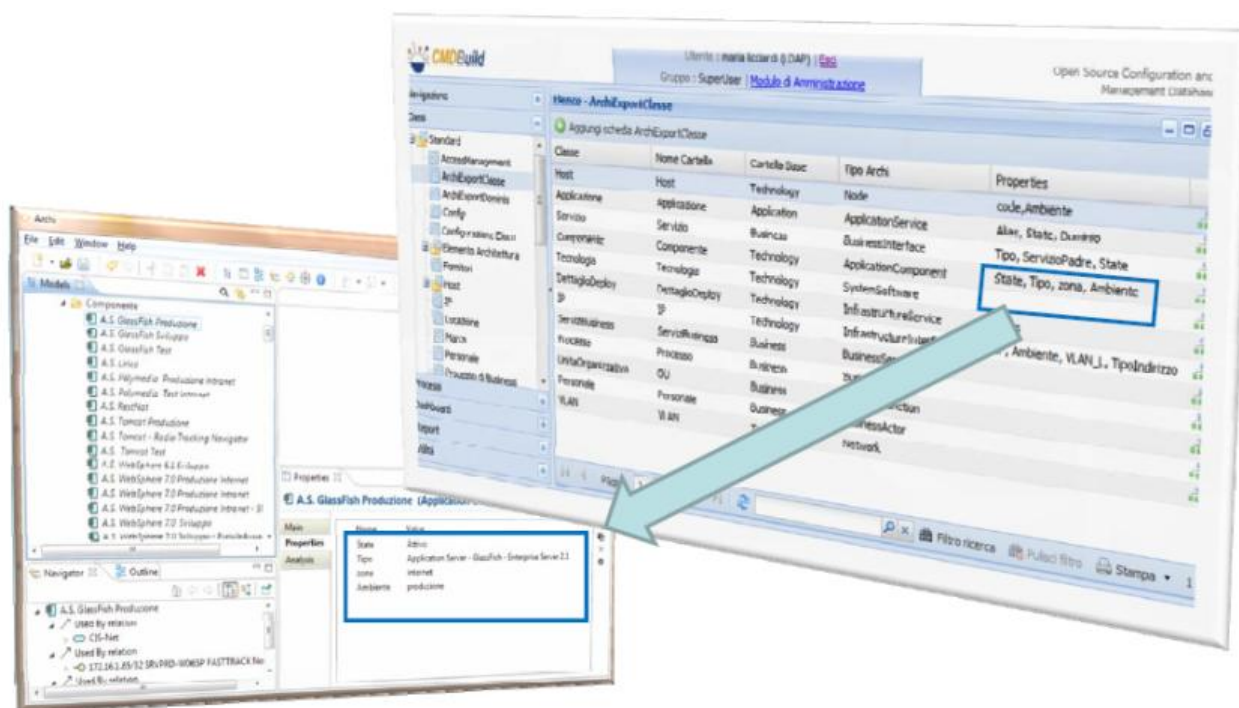
Click on this link to watch video and slides (in Italian) of the last intervention:

<http://www.cmdbuild.org/it/diffusione/convegni-e-workshop/togaf-archi-e-cmdbuild>

Operation modes

Also in this case the main functionality of the connector is to keep the information synchronized in two environments, defining the mapping rules between:

- Archi entities and CMDBuild classes
- Archi relations and CMDBuild domains



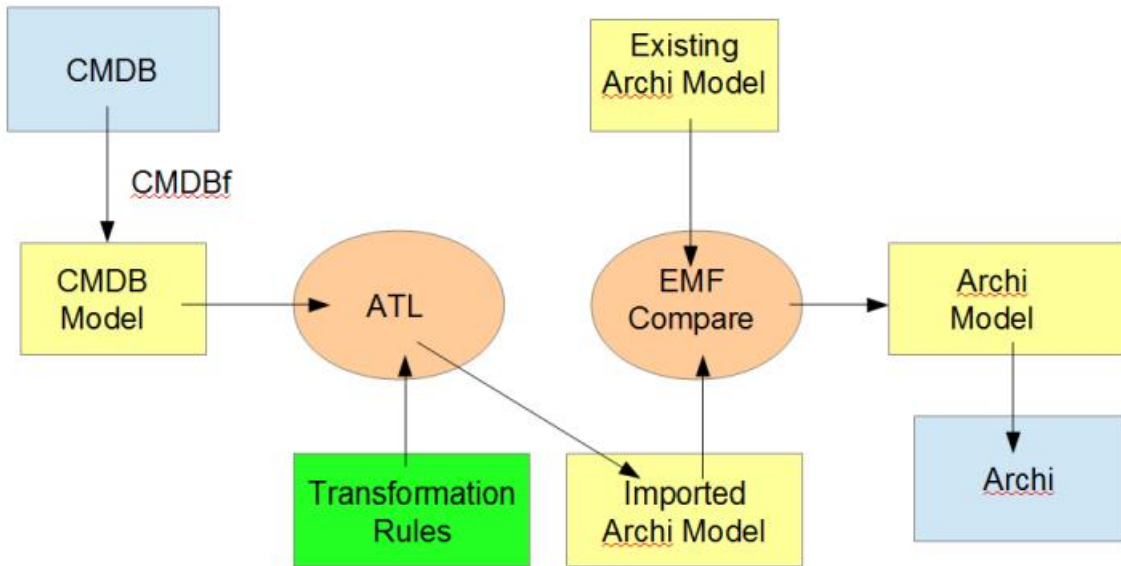
Main characteristics of the connector are:

- synchronizes data and diagrams (in particular imports in CMDBuild the Archi diagrams as attachments)
- accesses CMDBuild data using a webservice CMDBf-compliant (<http://www.dmtf.org/standards/cmdbf>), made by one of the two connector contributors
- uses the ATL standard language to define mapping between Archi <=> CMDBuild; ATL (Atlas Transformation Language-<http://www.eclipse.org/at/>) is based on EMF (Eclipse Modeling framework - <http://www.eclipse.org/modeling/emf/>)
- run the EMF models comparison using the Eclipse EMF Compare library
- handles bidirectional synchronization
- highlights assets that the connector would eliminate, finding them in the target system but

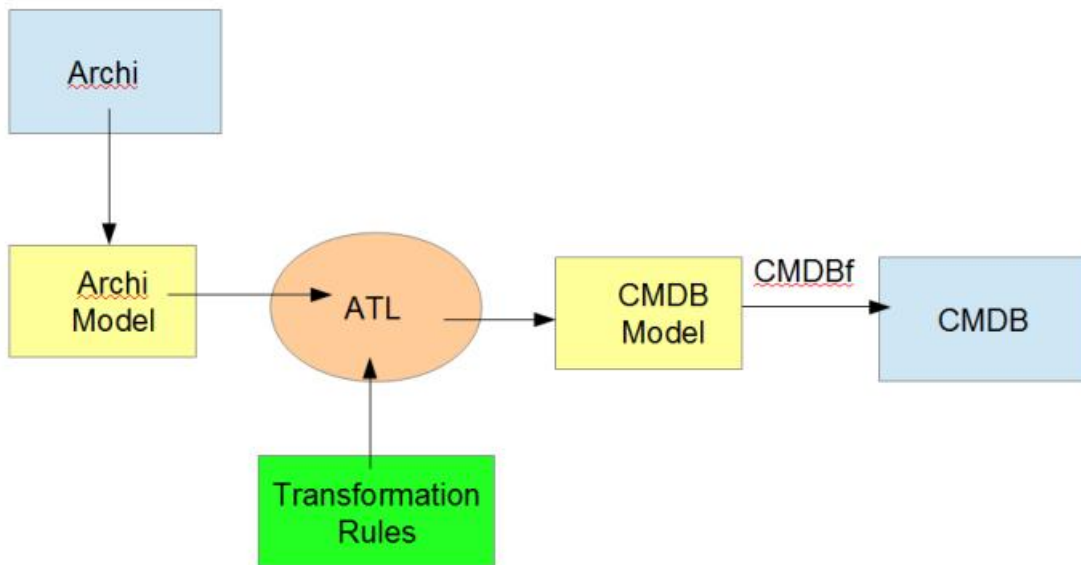
- not in the origin
- shows the differences between the two systems

Here are the operating patterns of the two types of synchronization.

Synchronizing CMDBuild => Archi



Synchronizing CMDBuild Archi => CMDBuild



Configuration

Following are some guidelines for the connector configuration.

For more information please refer to the instructions in the connector's download file.

Installing the plugin

The plugin installation requires the following:

- download the `archimate-cmdbuild-plugin.zip` file, located in the Bitbucket repository (<https://bitbucket.org/bompani/cmdbuild-unibo/downloads>)
- extract the ZIP archive
- extract the `plugins.zip` file inside the Archi plugins directory.
- copy the file `archimate-cmdb-plugin.jar` to the Archi plugins folder
- import `.xsd` files to create the classes and the domains in CMDBuild, files are available at <http://localhost/cmdbuild/services/cmdb-schema/>
- restart after upload

Plugin configuration files

Archi configuration directory `{HOME}/uk.ac.bolton/Archi` contains the configuration files of the plugin:

- `cmdbf.xml`
- `archi2cmdbuild.atl`
- `cmdbuild2archi.atl`

File `cmdbf.xml`

Here's an example of the configuration file `cmdbf.xml`:

```
<configuration>
  <cmdb id="http://www.cmdbuild.org">
    <queryWSDL>
      http://localhost:8080/cmdbuild/services/soap/CMDBfQuery?wsdl
    </queryWSDL>
    <registrationWSDL>
      http://localhost:8080/cmdbuild/services/soap/CMDBfRegistration?wsdl
    </registrationWSDL>
    <query>
      ... {CMDBf Query}
    </query>
    <archi2cmdb>
      file:/home/luca/uk.ac.bolton/Archi/archi2cmdbuild.atl
    </archi2cmdb>
    <cmdb2archi>
      file:/home/luca/uk.ac.bolton/Archi/cmdbuild2archi.atl
    </cmdb2archi>
  </cmdb>
</configuration>
```

Plugin configuration

To configure the plugin:

- install the ATL SDK plugin in Eclipse
- edit files `archi2cmdbuild.atl` and `cmdbuild2archi.atl` to define the transformations between the ArchiMate meta-models and CMDB meta-models

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.