



# **CMDBuild**

***Open Source Configuration and Management Database  
External Connectors***

**Versione 0.6.1  
Settembre 2007**

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

CMDBuild leverages many great technologies from the open source community:  
PostgreSQL, Apache, Tomcat, Eclipse, JasperReports, IReport  
We are thankful for the great contributions that led to the creation of that products

CMDBuild è un progetto realizzato e gestito da:

Comune di Udine – Servizio Sistemi Informativi e Telematici



Tecnoteca S.r.l. ([www.tecnoteca.com](http://www.tecnoteca.com))



Cogitek S.r.l. ([www.cogitek.it](http://www.cogitek.it))



CMDBuild è rilasciato con licenza GPL ([www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html))  
Copyright ©2006 Tecnoteca srl

Il sito ufficiale di CMDBuild è <http://www.cmdbuild.org>

## Sommario

Introduzione.....	4
I moduli di CMDBuild.....	4
Interoperabilità.....	5
Scopo del manuale.....	5
OCS Inventory.....	6
Generalità.....	6
Criteri di implementazione.....	6
Installazione interfaccia.....	10
Definizione visuale “mapping” attributi.....	12
Interoperabilità con altri sistemi di inventory.....	16
DCL – Double Choco Latte.....	18
Generalità.....	18
Personalizzazioni disponibili.....	18
Alfresco.....	20
Generalità.....	20
Appendice.....	21
Attributi per definizione “mapping” OCS Inventory.....	21
Viste per database OCS Inventory.....	24
Esempio file XSLT di “mapping” per OCS Inventory.....	26

# Introduzione

CMDBuild è una applicazione Open Source per la configurazione e gestione del database della configurazione (CMDB) degli oggetti in uso presso il Dipartimento IT di una organizzazione.

Gestire un Database della Configurazione significa mantenere aggiornata e disponibile per gli altri processi la base dati relativa agli elementi informatici utilizzati, alle loro relazioni ed alle loro modifiche nel tempo.

CMDBuild si ispira alle "best practice" ITIL (Information Technology Infrastructure Library), ormai affermatesi come "standard de facto", non proprietario, per la gestione dei servizi informatici secondo criteri orientati ai processi.

Con CMDBuild l'amministratore del sistema può costruire autonomamente il proprio CMDB (da cui il nome del progetto), grazie ad un apposito programma di configurazione che consente di aggiungere progressivamente nel sistema nuove classi di oggetti, nuovi attributi e nuove tipologie di relazioni.

Tramite il sistema implementato per la gestione del workflow è poi possibile definire in modo visuale con un editor esterno nuovi processi operanti sulle classi trattate nel sistema, importarli in CMDBuild ed eseguirli secondo i criteri indicati.

Sono infine disponibili sistemi di interfaccia con applicazioni open source esterne specializzate in attività connesse a CMDBuild (Automatic Inventory, Help Desk e prossimamente Document Management).

## I moduli di CMDBuild

Il sistema CMDBuild comprende due moduli principali:

- il Modulo Schema, dedicato alla definizione iniziale ed alle successive modifiche della struttura dati (classi e sottoclassi, attributi delle classi, tipologie di relazioni fra classi)ImportAsset

## Interoperabilità

I dati archiviati nel CMDB devono essere accessibili, compatibilmente con le politiche di sicurezza definite, da tutte le applicazioni interessate alla loro gestione, nell'ambito dei processi ITIL o dedicate ad esigenze amministrative.

Per garantirne la consultazione e modifica da applicazioni esterne è previsto lo sviluppo di un webservice SOAP nel quale saranno disponibili in modo controllato metodi di accesso esterno ai dati di CMDBuild.

Nell'ottica dell'interoperabilità con altri sistemi sono state anche implementate interfacce con applicazioni open source esterne preesistenti specializzate in attività connesse a CMDBuild:

- Automatic Inventory (OCS Inventory)
- Help Desk (DCL - Double Choco Latte)
- Document Management (Alfresco, prevista per la release 0.7 di CMDBuild)

Questo nell'ambito di una filosofia complessiva volta a:

- estendere le funzionalità di CMDBuild riutilizzando in modo “embedded” soluzioni valide già presenti sul mercato, con ovvi benefici di costi e di tempistiche
- suggerire agli utenti di CMDBuild l'utilizzo esteso delle applicazioni individuate, al fine di risolvere eventuali proprie esigenze applicative e ritrovando l'integrazione con CMDBuild come valore aggiunto
- fornire un servizio complessivo di consulenza ai propri Clienti
- valorizzare soluzioni open source di qualità attinenti alle problematiche affrontate da CMDBuild

## Scopo del manuale

Il presente manuale comprende sezioni differenziate per ognuno dei sistemi di interfaccia resi disponibili in CMDBuild, e al momento in particolare per:

- OCS Inventory (<http://ocsinventory.sourceforge.net/>)
- DCL - Double Choco Latte (<http://dcl.sourceforge.net/>)
- Alfresco (<http://www.alfresco.com/>), la cui sezione sarà sviluppata nelle release successive

Ogni sezione comprende sia una descrizione concettuale dell'interfaccia, sia eventuali indicazioni tecniche e sistemistiche di installazione e configurazione.

# OCS Inventory

## Generalità

La raccolta manuale delle variazioni effettuate al numero ed alle caratteristiche dei beni hardware e software gestiti può introdurre problemi di ritardo o imprecisione nell'aggiornamento dei dati.

Per una gestione semiautomatica di tali aggiornamenti è stato individuato il prodotto Open Source OCS Inventory, in grado di:

- rendere disponibili programmi “agenti” da installare sui computer da controllare (con sistema operativo Windows, Linux o Mac OS)
- raccogliere le informazioni provenienti dagli agenti (relativamente a BIOS, processore, RAM, device di input, controller, periferiche, schede aggiuntive, impostazioni rete, sistema operativo, applicazioni software, ecc)
- archivarle in un proprio database

Il sistema di interfaccia fra CMDBuild e OCS Inventory è basata sui seguenti criteri:

- confronto periodico fra i dati archiviati in CMDBuild ed i dati rilevati da OCS Inventory, reso possibile dalla disponibilità di:
  - una descrizione formale personalizzata del “mapping” fra i due insiemi di attributi (basata su file XML e XSLT)
  - API specifiche per l'interrogazione del database di CMDBuild
- alimentazione del sistema di workflow di CMDBuild con richieste di modifica automatiche (variante semplificata di un processo di Change Management) generate sulla base delle variazioni di configurazione rilevate

Il secondo criterio è ovviamente dipendente dai requisiti di controllo esplicito e di gestione dei ruoli e delle responsabilità richiesti da ITIL.

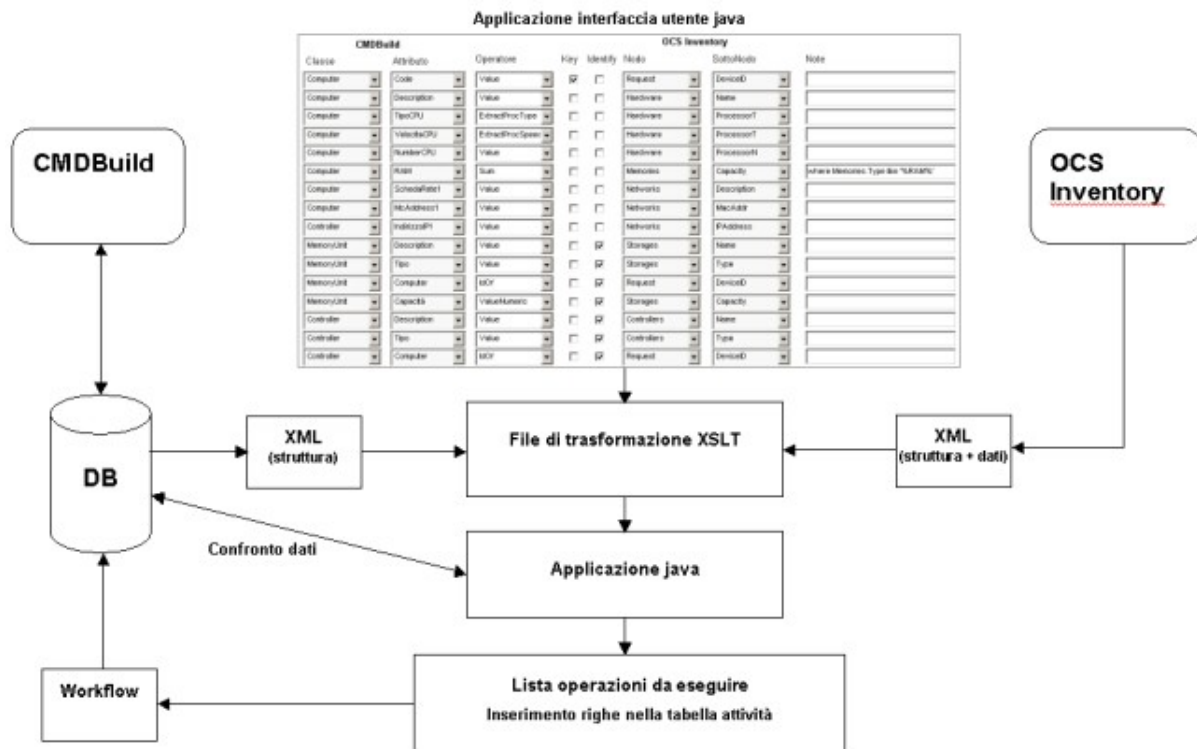
I criteri di “mapping” fra classi e relativi attributi presenti nei due sistemi sono personalizzabili (tramite l'applicazione visuale descritta nel seguito) per poter corrispondere alla struttura delle classi definita da ciascun ente in CMDBuild in funzione delle proprie necessità e della propria organizzazione interna.

## Criteri di implementazione

L'applicazione di interfaccia è basata in particolare sui seguenti elementi:

- file XML con struttura dati estratta da CMDBuild
- file XML con dati e struttura, estratto da OCS Inventory
- file XSLT di “mapping” fra i due file XML, con modalità specifiche per ogni cliente, derivato dalla compilazione di una form guidata
- insieme di viste che devono essere create sul database di OCS Inventory (la definizione completa è riportata in appendice)
- applicazione java di rilevamento delle modifiche rilevate ai dati della configurazione e produzione dei comandi di richiesta modifiche da eseguire tramite specifico processo di workflow

Segue lo schema grafico dei flussi dati:



Il modulo di definizione visuale delle modalità di “mapping” viene descritto al paragrafo successivo.

Sulla base della modalità di controllo implementate l’applicazione java:

- interroga il database PostgreSQL tramite alcune chiamate API rese disponibili da CMDBuild
- analizza le variazioni da gestire nell’ambito dei dati complessivi rilevati da OCS Inventory ed acceduti attraverso le viste definite nel database (i relativi script di creazione sono riportati in appendice)
- per ogni modifica riscontrata e tramite una opportuna API, attiva un’istanza di un processo semplificato di “Change Management” (denominato “ImportAsset”, richiede la presa visione ed approvazione da parte del responsabile della configurazione):
  - creando una attività iniziale di richiesta di approvazione della modifica
  - valorizzando il campo descrizione con una descrizione riepilogativa della richiesta
  - valorizzando un campo “di servizio” con la lista dei comandi API (e relativi parametri) che il motore di workflow dovrà poi eseguire in caso di approvazione

E’ stata predisposta ed è fornita con il rilascio una variante del processo “ImportAsset”, denominato “ImportAssetMail”, basato sull’inoltro di una mail di notifica e sul contestuale aggiornamento del database di CMDBuild (in questo caso si tratta quindi di una approvazione implicita anziché esplicita).

Il sistema estremamente generico sopra descritto (basato su schemi XML, file XSLT generato in modo visuale, wizard di configurazione, viste di accesso al database) si presta per poter essere utilizzato in altri ambiti ove siano richieste funzionalità di aggiornamento periodico di CMDBuild da database esterni.

Tali ambiti possono comprendere sistemi di inventory diversi da CMDBuild, ma anche altre tipologie di basi dati, quali ad esempio archivi di dipendenti o per la gestione amministrativa dei cespiti.

## Prerequisiti CMDBuild

### Aggiornamento con approvazione esplicita

Per attivare l'applicazione di interfaccia è necessario, come descritto più sopra, creare il processo "ImportAsset" che sarà alimentato con le variazioni di configurazioni riscontrate (una istanza di processo per ogni variazione di configurazione).

Per creare il processo all'interno di CMDBuild è necessario procedere come segue:

- creare una nuova classe derivata da Activity, specifica per la tipologia del processo in questione, così come visibile nella figura sottostante:



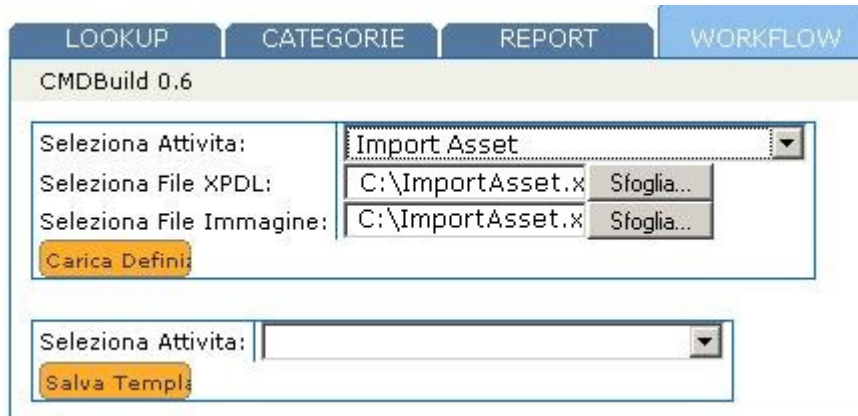
www.cmdbuild.org      CMDBuild è un progetto di: Tecnoteca srl, Comune di Udine, Cogitek srl      Copyright © Tecnoteca srl

- selezionare la classe appena creata, successivamente entrare nel TAB "Attributi" e aggiungere un attributo denominato "accept", di tipo booleano, con descrizione come riportato nell'immagine sottostante.





- dal TAB “Impostazioni” entrare nel TAB di secondo livello “Workflow” e importare il file ImportAsset.xpdl contenente la definizione del processo per l'attività ImportAsset appena creata, cliccando sul pulsante “Carica definizione” per terminare l'import (vedi screenshot seguente)



### Aggiornamento con notifica via mail

Nel caso si preferisca ricevere notifica dell'aggiornamento senza approvazione esplicita può essere utilizzato il processo “ImportAssetMail”, presente nel rilascio (o disegnarne uno simile).

Anche in questo caso il processo sarà alimentato con le variazioni di configurazioni riscontrate (una istanza di processo per ogni variazione di configurazione).

Per creare il processo all'interno di CMDBuild è necessario procedere come segue:

- creare una nuova classe derivata da Activity, specifica per la tipologia del processo in questione, così come è stato fatto per il processo precedente, modificandone opportunamente il nome.
- dal TAB “Impostazioni” entrare nel TAB di secondo livello “Workflow” e importare il file ImportAssetMail.xpdl contenente la definizione del processo per l'attività ImportAssetMail appena creata, tramite il pulsante “Carica definizione” (come fatto precedentemente)

Nel seguito del presente manuale sono descritte le modalità di installazione di OCS Inventory, i parametri richiesti per la configurazione dell'applicazione di interfaccia (wizard mapping attributi) ed i criteri da seguire per la definizione delle viste nel database di origine dei dati di interesse.

## Installazione OCS Inventory

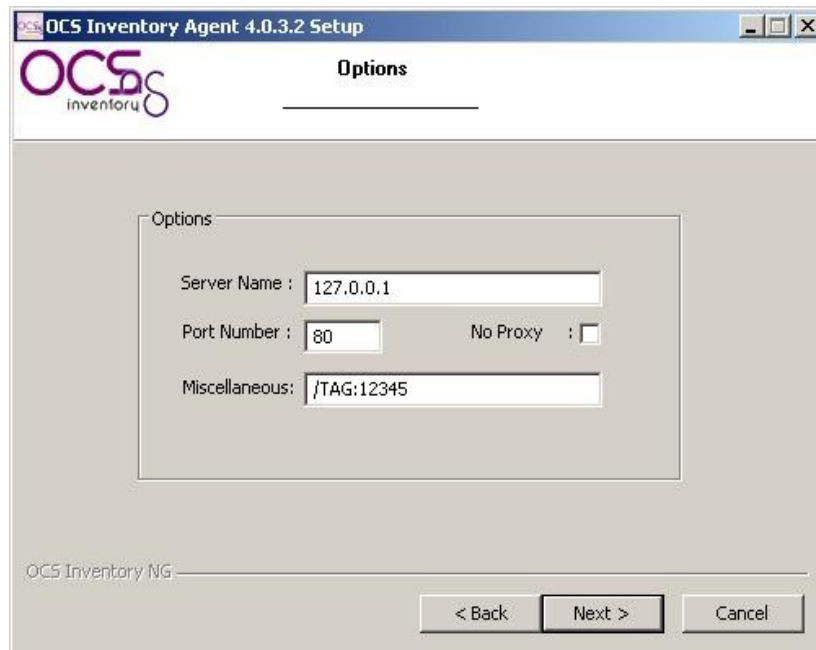
Prima di procedere all'installazione dell'interfaccia è necessario configurare correttamente il server di OCS Inventory ed il relativo agente sulle macchine da inventariare.

Per la configurazione del server si rimanda alla documentazione di OCS Inventory disponibile sul sito dell'applicazione, mentre per la configurazione dell'agente si consiglia di procedere come segue:

- scaricare i file necessari per procedere all'installazione relativi alla piattaforma desiderata;
- installare la versione che permette di installare l'agente come servizio di Windows (nella versione 1.01 questo file si chiama OcsAgentSetup.exe)
- nel corso dell'installazione, nella schermata per la scelta delle opzioni, indicare l'indirizzo

del server (Server Name), il numero della porta del server (Port Number) e l'identificativo univoco che si vuole eventualmente utilizzare per l'identificazione dell'asset, ad esempio il numero di inventario aziendale.

La sintassi da utilizzare per la specifica dell'identificativo univoco è visibile nella figura sottostante:



Per modificare eventualmente i parametri specificati in fase di setup spostarsi nella cartella del client su cui è installato l'agente e digitare il seguente comando:

```
OCSInventory /server:[indirizzo IP] /pnum:80 /tag: [codice univoco PC]
```

La versione di OCS Inventory correntemente supportata dall'interfaccia sviluppata è la versione 1.01.

## Installazione interfaccia

L'installazione dell'applicazione di interfaccia richiede i seguenti passaggi:

- posizionamento dei file di programma
- posizionamento dello schema XML delle classi di CMDBuild
- creazione script per l'esecuzione periodica

Per il corretto funzionamento del sistema è richiesto Java 1.5

### Posizionamento file di programma


Il file "ConnectorInventory", di tipo "jar", va posizionato in una sottocartella del sistema su cui si opera, che si suggerisce sia situata nell'ambito della installazione di CMDBuild.

Nella cartella scelta va creata una ulteriore sottocartella chiamata obbligatoriamente "xml", nella quale vanno posizionati:

- lo schema delle classi di CMDBuild “cmdbuild-schema.xml”, ottenuto come sotto specificato
- i file di trasformazione “transform.xml” e “configuration.xml” creati tramite l'interfaccia visuale descritta più avanti nel presente capitolo

### Estrazione schema classi CMDBuild

Per l'estrazione in formato XML dello schema delle classi di CMDBuild è necessario utilizzare il Modulo Schema, spostarsi nel TAB “Impostazioni” e poi nella voce “Export”.



The screenshot shows the CMDBuild web interface. At the top, there's a navigation bar with tabs: SCHEMA, IMPOSTAZIONI (selected), and UTENTI. Below this, there's a sub-navigation bar with tabs: LOOKUP, CATEGORIE, REPORT, WORKFLOW, EXPORT (selected), and SETUP. The main content area shows 'CMDBuild 0.55' and a form titled 'Esporta Schema Classi'. The form has a dropdown menu 'Seleziona Tipo:' with 'XML' selected and a button 'Esporta Sche' highlighted. On the left, there's a sidebar with a tree view of categories like Account, Attività, Asset, etc. The footer contains the website URL, project information, and copyright notice.

A questo punto è sufficiente cliccare su “EsportaSchema” e selezionare la cartella di destinazione.

Non deve essere modificato il nome del file affinché questo possa essere riconosciuto dal programma di interfaccia.

### Script esecuzione periodica

Per eseguire manualmente il file “jar” è sufficiente posizionarsi nella directory in cui si trova il file e scrivere il comando:

```
java -jar ConnectorInventory.jar percorsoXML/ nomeProcesso
```

dove:

- percorsoXML è il percorso dove trovare i file xml (esempio ./xml/)
- nomeProcesso è il nome dell'attività che si vuole utilizzare (ad esempio “ImportAsset” oppure “ImportAssetMail”)

Un esempio di comando quindi potrebbe essere:

```
java -jar ConnectorInventory.jar ./xml/ ImportAsset
```

Per l'utilizzo a regime con il sistema CMDBuild si raccomanda di creare uno script e richiamarne l'esecuzione periodica all'interno del “cron” di sistema.

Si riporta di seguito un esempio dello script da creare:

```
#!/bin/bash
# -----
#
# esecuzione jar
java -jar /usr/local/ConnectorInventory/ConnectorInventory.jar 2>> error.log
```

## Definizione visuale “mapping” attributi

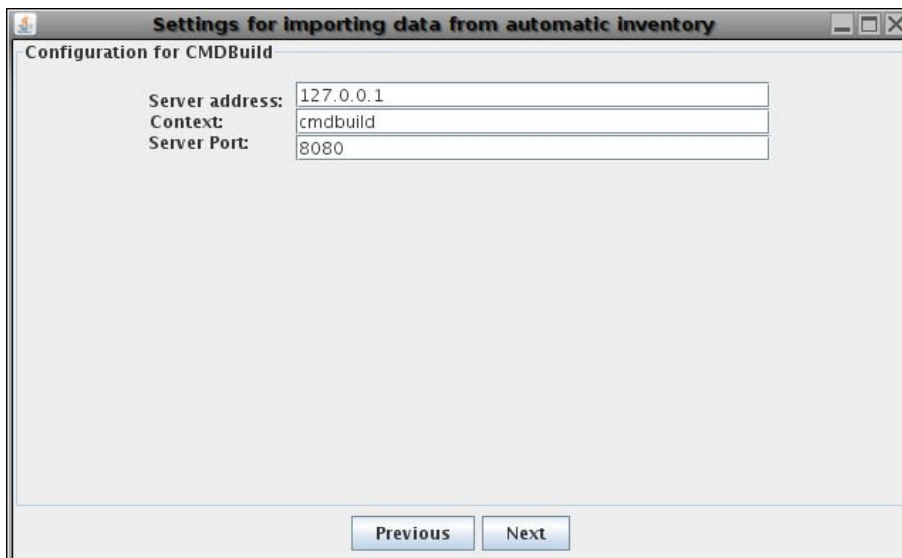
L'operazione di definizione visuale del “mapping” fra gli attributi gestiti in CMDBuild e quelli rilevati in OCS Inventory viene effettuata tramite una sequenza di passaggi guidati a e produce il file XSLT interpretato dall'applicazione di interfaccia.

Il file XSLT può essere poi editato manualmente per eventuali integrazioni di ulteriori criteri non gestiti dal wizard implementato (filtri, vincoli, ecc).

### Passaggio 1

Il primo passaggio proposto nel wizard richiede di indicare i parametri di configurazione di CMDBuild, necessari all'applicazione di interfaccia per sapere dove si trova il server e come chiamare le API remote:

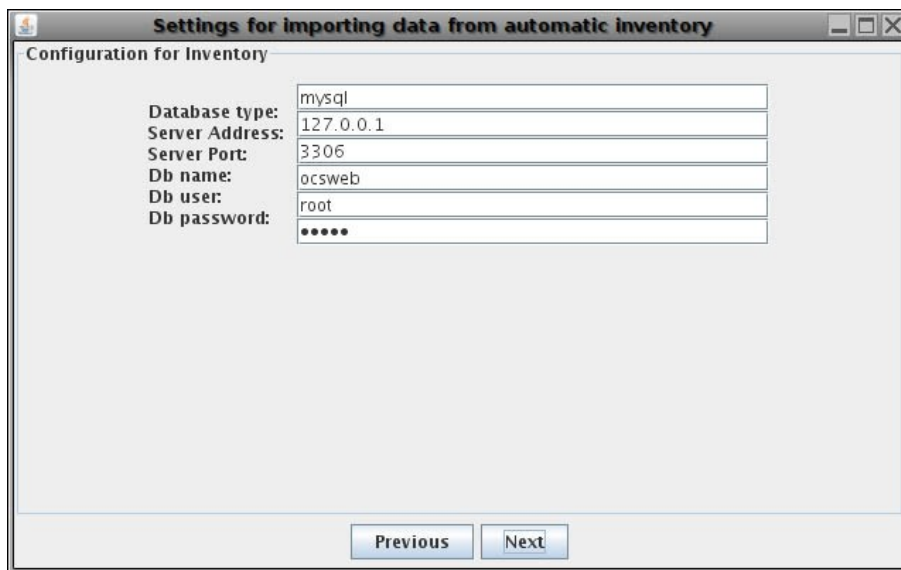
- indirizzo del server su cui è installato CMDBuild
- nome del file di contesto di CMDBuild
- porta del server / Tomcat



### Passaggio 2

Il secondo passaggio del wizard richiede di indicare i parametri di accesso al database di OCS Inventory, dove poter leggere le informazioni rilevate sui computer gestiti:

- tipo di database (attualmente supportato solo MySQL)
- indirizzo del server su cui si trova il database di OCS Inventory
- porta del database del server su cui si trova OCS Inventory
- nome del database di OCS Inventory
- nome utente
- nome password



Settings for importing data from automatic inventory

Configuration for Inventory

Database type:

Server Address:

Server Port:

Db name:

Db user:

Db password:

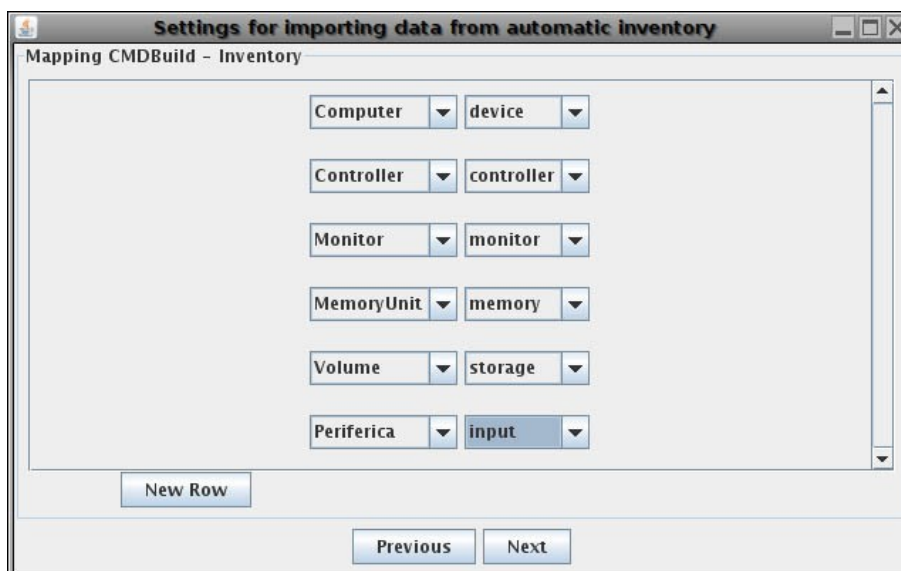
Previous Next

### Passaggio 3

Il terzo passaggio del wizard consente di correlare le categorie di informazioni rilevate sul campo ed archiviate in OCS Inventory (corrispondenti a tabelle del database) con le classi definite in CMDBuild.

Tale correlazione è poi tradotta nel file XSLT (vedi appendice) con una sezione del tipo:

```
<xsl:template match="/inventory/device">
  <Computer key="Code" identify="Code">
    ...
  </Computer>
</xsl:template>
```



Settings for importing data from automatic inventory

Mapping CMDBuild - Inventory

Computer	device
Controller	controller
Monitor	monitor
MemoryUnit	memory
Volume	storage
Periferica	input

New Row

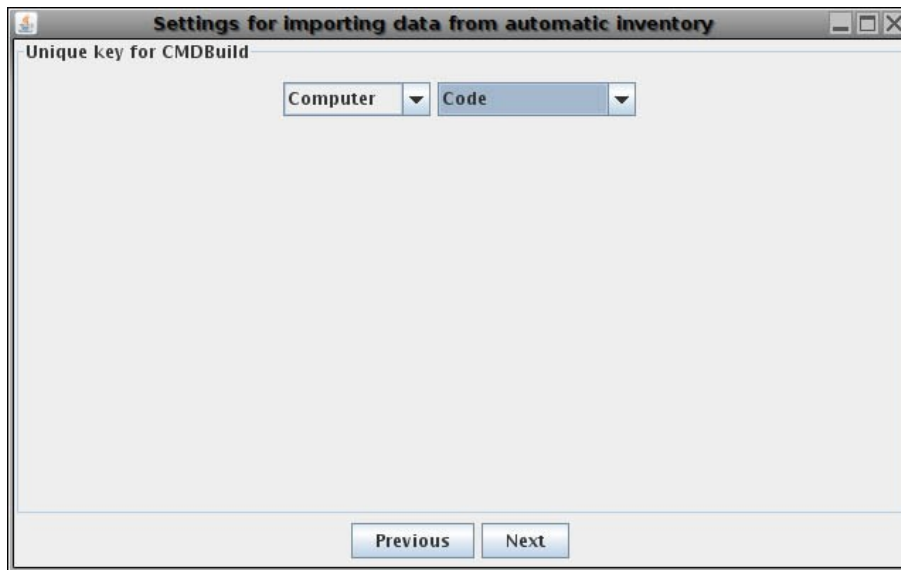
Previous Next

### Passaggio 4

Il quarto passaggio richiede di specificare l'informazione da considerare come identificativo univoco del computer in CMDBuild.

Tale informazione è richiesta solamente per la classe principale ('master') del sistema di interfaccia

(le classi contenenti le informazioni correlate sono invece indicate come 'slave').

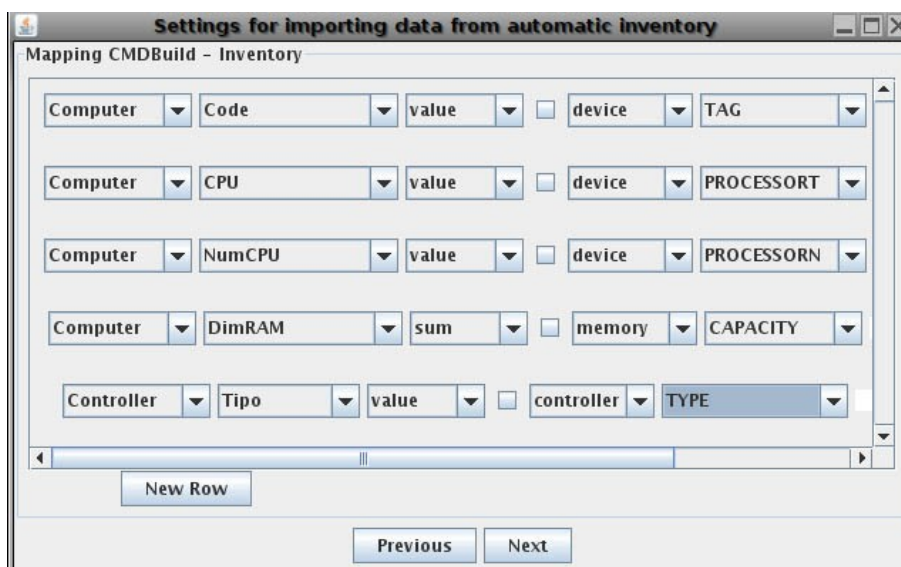


Nell'ambito del file XSLT la chiave univoca corrisponde a definizioni del tipo key = "Code"

Si suggerisce di valutare con attenzione quale attributo utilizzare come campo univoco del computer: un numero di cespite interno (da specificare "una tantum" all'installazione dell'Agente OCS tramite l'apposito flag "tag"), il MAC Address della scheda di rete (non necessariamente invariante in caso di una sua sostituzione), ecc.

### Passaggio 5

Il quinto passaggio del wizard comprende l'effettiva definizione dei criteri di "mapping" fra i due sistemi da interfacciare.



La parte a destra della form è riservata a CMDBuild, la parte a sinistra ad OCS Inventory.

I campi prevedono controlli strutturali in funzione dei due modelli dati riconosciuti dal sistema, ad

esempio una volta scelto il nodo “Computer” come classe di CMDBuild sarà poi possibile selezionare solamente un attributo fra quelli definiti per quella classe.

Il sistema di “mapping” prevede non solamente l'associazione fra due attributi, ma anche fra un attributo ed una “funzione” definita sul secondo.

Al momento sono disponibili le seguenti funzioni:

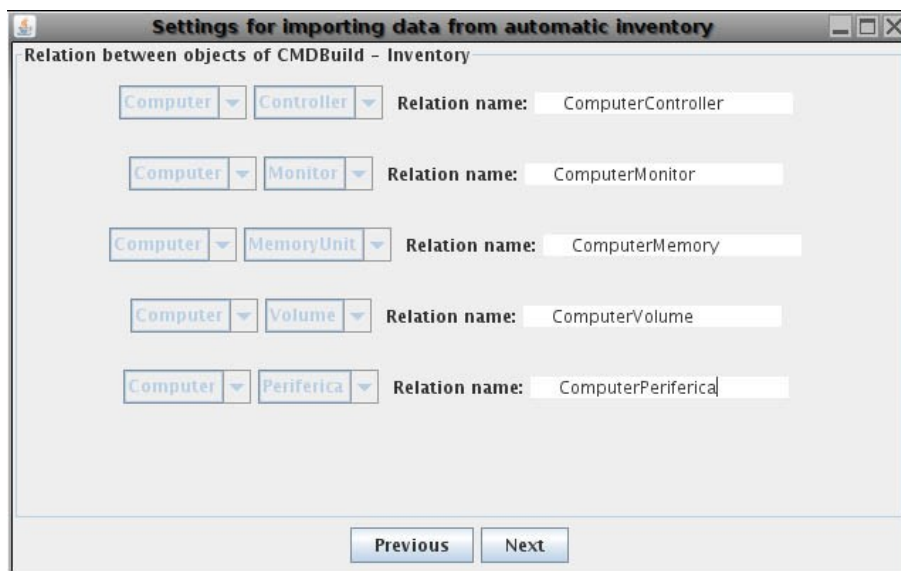
- value: riporta semplicemente il contenuto del nodo che si sceglie
- sum: effettua la somma dei valori contenuti nei nodi di OCS Inventory che vengono selezionati
- count: conta il numero di nodi che vengono selezionati
- free\_expr: permette di scrivere una propria espressione “Xpath” libera

L'insieme degli attributi segnalati come “identificativi” tramite l'apposito check consentiranno all'applicazione di interfaccia di riconoscere l'oggetto in questione e discriminare quindi fra l'operazione di modifica di un suo attributo e quella di cancellazione / reinserimento dell'intero oggetto (corrisponde nel file XSLT al tag “identify”).

### Passaggio 6

Il sesto passaggio prevede la identificazione dei domini esistenti fra le classi oggetto di aggiornamento periodico automatico, necessari all'applicazione di interfaccia per creare le relazioni fra gli oggetti rilevati.

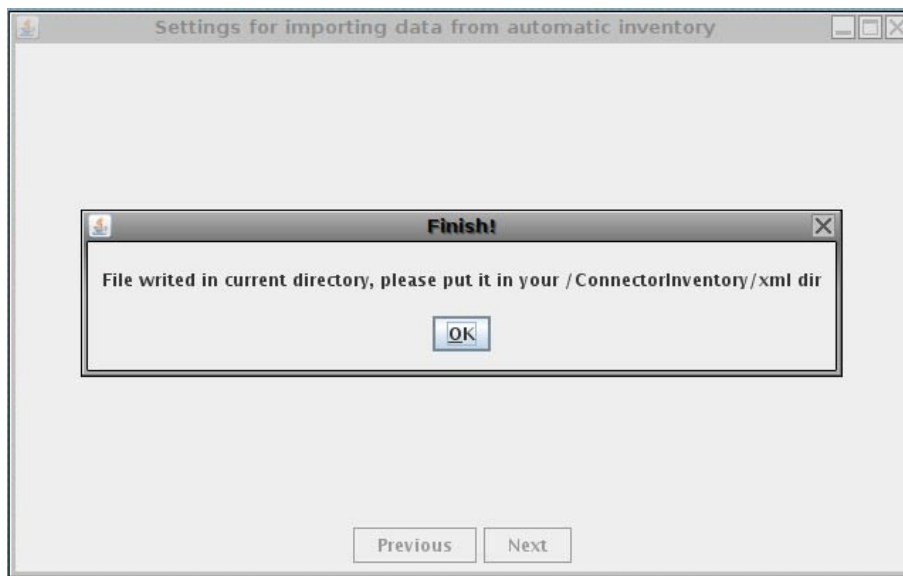
I nomi dei domini vanno digitati facendo attenzione alla esatta coincidenza con i nomi definiti in CMDBuild.



### Passaggio 7

Il settimo ed ultimo passaggio prevede solamente la conferma finale dell'operazione e l'indicazione della cartella in cui il file XSLT viene salvato.

Ricordiamo che il file va poi copiato, senza rinominarlo, nella sottocartella “xml” della cartella “ConnectorInventory”.



**Attenzione:** il wizard sopra descritto è rilasciato in forma provvisoria in attesa di un suo rifacimento nell'ambito del Modulo Schema.

Fra i bug della versione attuale segnaliamo la mancata gestione del salvataggio dei dati inseriti nel caso ci si sposti in una pagina precedente.

## Interoperabilità con altri sistemi di inventory

La genericità del sistema sopra descritto rende possibile in linea di massima utilizzare gli stessi meccanismi per interagire con applicazioni di inventariazione automatica diverse da OCS Inventory.

Sarà in tal caso necessario effettuare alcune verifiche tecniche relativamente alla struttura del database da interfacciare ed alla modalità di correlazione fra i due modelli di dati.

La configurazione dell'interfaccia con altri sistemi di inventory richiede:

- la definizione di “viste” nel database nativo del sistema utilizzato, al fine di “mascherare” differenze di struttura rispetto OCS Inventory
- la disponibilità di un driver JDBC per il database in questione
- la definizione dei parametri di connessione

### Definizione “Viste”

La struttura dati resa visibile all'esterno del database nativo del programma di Inventory scelto deve comprendere:

- una “vista” corrispondente alla tabella “master”, in cui saranno elencati i PC rilevati limitatamente alle proprie caratteristiche generali (in particolare conterrà il codice di identificazione univoca del cespite: numero di cespite, MAC Address scheda rete, serialnumber PC, ecc)
- più “viste” corrispondenti alle tabelle “slave”, in cui saranno elencate le periferiche rilevate per ogni PC, associate tramite riferimenti nel database



Nelle viste corrispondenti alle tabelle “slave” è indispensabile che il primo campo contenga il collegamento (effettuato tramite “id” numerico) con la vista corrispondente alla tabella “master”.

Nella vista “master” è invece previsto che il primo campo contenga l'ID numerico utilizzato per il collegamento con le schede “slave” e il secondo campo contenga il codice univoco del computer necessario per il suo riconoscimento sicuro nel database di CMDBuild.

I nomi delle viste definite devono contenere il prefisso “CMDBuild\_” per essere riconosciute dal programma di interfaccia.

### **Esempio 1**

Nella seguente vista “slave” relativa alle memorie di massa, il campo “Hardware\_ID” contiene l'ID di riferimento alla scheda “master” del Computer:

```
CREATE OR REPLACE VIEW CMDBuild_storage
as select HARDWARE_ID, ID, MANUFACTURER, MODEL, DESCRIPTION, TYPE, DISKSIZE
from storages;
```

### **Esempio 2**

Nella seguente vista “master” il campo “DEVICE” contiene l'ID con cui il computer viene riferito nelle schede “slave”, il campo “TAG” contiene il codice univoco con cui identificare il computer in CMDBuild (specificato manualmente “una tantum” oppure rilevato in automatico, con i problemi del caso, utilizzando il MAC Address o il serialnumber):

```
CREATE OR REPLACE VIEW CMDBuild_device
(DEVICE, TAG, DEVICEID, NAME, WORKGROUP, USERDOMAIN, OSNAME, OSVERSION,
OSCOMMENTS, PROCESSORT, PROCESSORS, PROCESSORN, MEMORY, SWAP)
as select ID, TAG, DEVICEID, NAME, WORKGROUP, USERDOMAIN, OSNAME, OSVERSION,
OSCOMMENTS, PROCESSORT, PROCESSORS, PROCESSORN, MEMORY, SWAP
from hardware, accountinfo WHERE ID=HARDWARE_ID
```

I criteri descritti per l'interfacciamento di un'altra applicazione di inventory possono essere utilizzati in senso esteso per implementare sistemi di aggiornamento da altre tipologie di basi dati (archivi di dipendenti, archivi cespiti in applicazioni ERP, ecc).

# DCL – Double Choco Latte

## Generalità

L'applicazione CMDBuild si occupa della gestione dell'inventario degli asset, su cui offre la possibilità di definire processi di gestione tramite un meccanismo "general purpose" basato sul linguaggio XPDL.

Il sistema consente la definizione di processi generici utili per le normali attività di gestione, ma non comprendenti le funzionalità avanzate disponibili in un sistema di Help Desk sofisticato.

Qualora le funzionalità base ottenibili fossero ritenute sufficienti si potrà decidere di gestire il processo di Help Desk internamente al sistema CMDBuild (vedi esempio descritto nel manuale "Workflow Tutorial"), definendo nel suo modello dati le ulteriori classi necessarie (ticket, ordini di lavoro, ecc).

Tale soluzione gestisce classi e processi in modo integrato ed omogeneo, consentendo l'archiviazione e consultazione delle correlazioni fra il processo stesso e le corrispondenti entità trattate nel sistema CMDBuild.

Qualora le funzionalità e l'interfaccia utente disponibili non fossero invece ritenute sufficienti è stata prevista l'integrazione con il sistema open source di larga diffusione DCL (Double Choco Latte), che comprende funzioni per:

- apertura nuovo ticket con classificazione, descrizione, caricamento file allegati (vedi screenshot successivo)
- creazione / assegnazione / modifica / annullamento / aggiornamento degli ordini di lavoro relativi ai ticket non direttamente risolvibili, con notifica via mail alle persone interessate
- registrazione in un apposito archivio di "timecard" (foglio di lavoro) degli orari e delle operazioni svolte
- utilizzo tabelle account, personale tecnico, dipartimenti personale tecnico, canale di ricezione / severità / priorità / stato della segnalazione, prodotti / moduli oggetto di segnalazioni

## Personalizzazioni disponibili

Le personalizzazioni effettuate da Tecnoteca sui sorgenti di DCL riguardano:

- associazione di un ticket ad un asset ricercabile nella corrispondente tabella del database di CMDBuild
- associazione di un ticket ad un account da una lista popolata accedendo alla corrispondente tabella del database di CMDBuild

Le integrazioni effettuate consentono quindi di operare da DCL sugli asset e sui relativi assegnatari gestiti in CMDBuild, non sono al momento gestite ulteriori interazioni o integrazioni fra i due prodotti.

Segue lo screenshot della form di apertura ticket personalizzata per la selezione dell'asset e dell'account dal database di CMDBuild:

Work orders ▾ Progetti ▾ Tickets ▾ Report ▾ Amministrazione ▾ Aiuto ▾

Vai al ticket

**Tickets**

- Miei ticket
- Modifica immagine**
- Chiudi immagine
- Statistiche
- Cerca
- Sfoggia

**Opzioni**

- Nuovo Work Order
- Nuovo progetto
- Nuovo Ticket
- Versione stampa

### Aggiungi ticket

**Responsabile:**

**Contatto:**  **Telefono Contatto:**

**E-Mail Contatto:**

---

**Cerca un bene per (parte di) codice:**   -

**Beni trovati:**

**Cliente:**

Nota: la selezione per cliente si basa sul risultato della ricerca effettuata. Per visualizzare tutti i beni di un cliente, prima premere *visualizza tutti*, quindi selezionare un cliente.

**Prodotto:**  **Versione:**

**Module\*:**

**Priorità?:**  **Tipologia:**

**Canale:**

**Sommario:**

**Oggetto:**

**Allega File:**

**Soluzione:**

**\*\* I campi evidenziati sono Obbligatori!**

La versione modificata di DCL è resa disponibile con il codice sorgente a chi ne faccia richiesta tramite la pagina dei contatti presente sul sito del progetto CMDBuild.

# Alfresco

## Generalità

La presente sezione sarà sviluppata nelle prossime release di CMDBuild, essendone prevista l'integrazione a partire dalla versione 0.7 del prodotto.

# Appendice

## Attributi per definizione “mapping” OCS Inventory

Si riporta di seguito la lista delle entità ed informazioni rilevati da OCS Inventory e di cui è richiesto il mapping nelle corrispondenti classi ed attributi definiti in CMDBuild (ultima colonna).

OCS Inventory	Vista I/F CMDBuild	Descrizione	Esempio	CMDBuild	
Hardware.Id AccountInfo.Tag -- Hardware.Name Hardware.Workgroup Hardware.UserDomain -- -- Hardware.OSName Hardware.OSVersion Hardware.ProcessorT Hardware.ProcessorS Hardware.ProcessorN Hardware.Memory Hardware.WinProdKey	CMDBuild_device	Id Tag Serial Name Workgroup UserDomain Manufacturer ProductName OSName OSVersion ProcessorT ProcessorS ProcessorN Memory WinProdKey	Id Chiave Numero di serie Nome Gruppo lavoro Dominio Produttore Nome prodotto Nome sistema operativo Versione sistema operativo Tipo processore Velocità processore Numero processori Memoria RAM Chiave prodotto	2 lisa HUB5390NQK Kelly Tecnoteca Kelly Hewlett-Packard HP Compaq nc6120 (PN936AV) Microsoft Windows XP Professional 5.1.2600 AMD Sempron(tm) 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	Riferimento macchina Produttore MotherBoard Modello MotherBoard Numero di serie M.B. Tipo chassis Produttore Versione Data	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	Riferimento macchina Produttore Nome Titolo Descrizione Versione Tipo	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	Riferimento macchina Lettera Tipo File System Spazio totale Spazio libero Numero files Nome volume	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	Riferimento macchina Tipo Produttore Titolo Descrizione Interfaccia Tipo puntatore	2 Pointing (Periferiche standard di sistema Periferica USB Human Interface Periferica USB Human Interface 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	Riferimento macchina Titolo Descrizione Dimensione Scopo Tipo Velocità Numero slots	2 Memoria fisica 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	Riferimento macchina Nome Modello Descrizione Tipo		... ...
Monitors.Hardware_Id Monitors.Manufacturer Monitors.Caption Monitors.Description Monitors.Type Monitors.Serial	CMDBuild_monitor	Device_Id Manufacturer Caption Description Type Serial	Riferimento macchina Produttore Titolo Descrizione Tipo Numero di serie	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	Riferimento macchina Descrizione Tipo TipoMIB Velocità Macaddress Stato Indirizzo IP Maschera IP IP Gateway IP DNS IP sottorete IP DHCP server	2 Scheda Fast Ethernet VIA compat. 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	Riferimento macchina Tipo Nome Titolo Descrizione	2 Serial Porta di comunicazione (COM1) Porta di comunicazione (COM1) Porta di comunicazione	... ...
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	Riferimento macchina Nome Descrizione Codice Utilizzo Stato Condivisione	2 Slot di sistema Slot di sistema PC11 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	Riferimento macchina Produttore Nome Versione Cartella Commento Nome file Dimensione file Sorgente	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	Riferimento macchina Produttore Nome Descrizione	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	Riferimento macchina Produttore Nome Modello Descrizione Tipo Dimensione disco	2 (unità disco standard) Generic Flash Disk USB Device //./PHYSICALDRIVE1 Unità disco 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	Riferimento macchina Nome Chipset Memoria Risoluzione Frequenza refresh	2 RADEON 9200 PRO Family (Microsoft C.) RADEON 9200 PRO AGP (0x5960) 128 1280 x 1024 64	... ...

## Viste per database OCS Inventory

Si riportano di seguito le definizioni delle viste che dovranno essere definite nel database di OCS Inventory per poter attivare l'interfaccia con CMDBuild.

### Catalogo viste

```
CREATE OR REPLACE VIEW CMDBuild_catalog
AS select TABLE_NAME from INFORMATION_SCHEMA.VIEWS
where table_name like '%CMDBuild_%' and table_name <> 'CMDBuild_catalog';
```

### Vista 'master'

```
CREATE OR REPLACE VIEW CMDBuild_device
(DEVICE, TAG, DEVICEID, NAME, WORKGROUP, USERDOMAIN, OSNAME, OSVERSION,
OSCOMMENTS, PROCESSORT, PROCESSORS, PROCESSORN, MEMORY, SWAP, IPADDR,
WINCOMPANY, WINOWNER, WINPRODID, WINPRODKEY)
as select ID, TAG, DEVICEID, NAME, WORKGROUP, USERDOMAIN, OSNAME, OSVERSION,
OSCOMMENTS, PROCESSORT, PROCESSORS, PROCESSORN, MEMORY, SWAP, IPADDR,
WINCOMPANY, WINOWNER, WINPRODID, WINPRODKEY
from hardware, accountinfo
WHERE ID=HARDWARE_ID;
```

### Viste 'slave'

```
CREATE OR REPLACE VIEW CMDBuild_controller
as select HARDWARE_ID, MANUFACTURER, NAME, CAPTION, DESCRIPTION, VERSION, TYPE
from controllers;
```

```
CREATE OR REPLACE VIEW CMDBuild_drive as
select HARDWARE_ID, LETTER, TYPE, FILESYSTEM, TOTAL, FREE, NUMFILES, VOLUMN
from drives;
```

```
CREATE OR REPLACE VIEW CMDBuild_input
as select HARDWARE_ID, TYPE, MANUFACTURER, CAPTION, DESCRIPTION,
INTERFACE, POINTTYPE
from inputs;
```

```
CREATE OR REPLACE VIEW CMDBuild_memory
as select HARDWARE_ID, CAPTION, DESCRIPTION, CAPACITY, PURPOSE,
TYPE, SPEED, NUMSLOTS
from memories;
```

```
CREATE OR REPLACE VIEW CMDBuild_bios
as select HARDWARE_ID, SMANUFACTURER, SMODEL, SSN, TYPE,
BMANUFACTURER, BVERSION, BDATE
from bios;
```

```
CREATE OR REPLACE VIEW CMDBuild_monitor
as select HARDWARE_ID, ID, MANUFACTURER, CAPTION, DESCRIPTION, TYPE, SERIAL
from monitors;
```

```
CREATE OR REPLACE VIEW CMDBuild_modem
as select HARDWARE_ID, ID, NAME, MODEL, DESCRIPTION, TYPE
from modems;
```

```
CREATE OR REPLACE VIEW CMDBuild_port
```



```
as select HARDWARE_ID, ID, TYPE, NAME, CAPTION DESCRIPTION
from ports;
CREATE OR REPLACE VIEW CMDBuild_slot
as select HARDWARE_ID, ID, NAME, DESCRIPTION, DESIGNATION,
        PURPOSE, STATUS, PSHARE
from slots;

CREATE OR REPLACE VIEW CMDBuild_software
as select HARDWARE_ID, ID, PUBLISHER, NAME, VERSION, FOLDER, COMMENTS,
        FILENAME, FILESIZE, SOURCE
from softwares;

CREATE OR REPLACE VIEW CMDBuild_sound
as select HARDWARE_ID, ID, MANUFACTURER, NAME, DESCRIPTION
from sounds;

CREATE OR REPLACE VIEW CMDBuild_storage
as select HARDWARE_ID, ID, MANUFACTURER, MODEL, DESCRIPTION, TYPE, DISKSIZE
from storages;

CREATE OR REPLACE VIEW CMDBuild_video
as select HARDWARE_ID, ID, NAME, CHIPSET, MEMORY, RESOLUTION
from videos;
```

## Esempio file XSLT di “mapping” per OCS Inventory

Segue un esempio del file XSLT di “mapping” fra la struttura dati di CMDBuild e di OCS Inventory, creato tramite l'interfaccia visuale descritta nel presente manuale.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" version="1.0" encoding="iso-8859-1" indent="yes"/>

<xsl:template match="/">
  <CMDBUILD>
    <xsl:apply-templates/>
  </CMDBUILD>
</xsl:template>

<xsl:template match="/inventory/device">
  <Computer key="Code" identify="Code">
    <Code><xsl:value-of select="/inventory/device/DEVICECODE"/></Code>
    <Marca></Marca>
    <Modello></Modello>
    <Description><xsl:value-of select="NAME"/></Description><!-- //TODO -->
    <CPU><xsl:value-of select="PROCESSORT"/></CPU>
    <NumCPU><xsl:value-of select="PROCESSORN"/></NumCPU>
    <DimRAM><xsl:value-of select="sum(/inventory/memory/CAPACITY)"/></DimRAM>
    <NumHD><xsl:value-of select="count(/inventory/storage[TYPE='Fixedxhard disk media'])"/></NumHD>
    <DimTotHD><xsl:value-of select="sum(/inventory/storage[TYPE='Fixedxhard disk media']/DISKSIZE)"/></DimTotHD>
    <SchedaRetel1><xsl:value-of select="/inventory/network[position()=1]/TYPE"/></SchedaRetel1>
    <SchedaRete2><xsl:value-of select="/inventory/network[position()=2]/TYPE"/></SchedaRete2>
    <MacAddress1> <xsl:value-of select="/inventory/network[position()=1]/MACADDR"/></MacAddress1>
    <MacAddress2> <xsl:value-of select="/inventory/network[position()=2]/MACADDR"/></MacAddress2>
    <IndirizzoIP1><xsl:value-of select="/inventory/network[position()=1]/IPADDRESS"/></IndirizzoIP1>
    <IndirizzoIP2><xsl:value-of select="/inventory/network[position()=2]/IPADDRESS"/></IndirizzoIP2>
    <NumSeriali><xsl:value-of select="count(/inventory/port[TYPE='Serial'])"/></NumSeriali>
    <NumParallele><xsl:value-of select="count(/inventory/port[TYPE='Parallel'])"/></NumParallele>
    <NumUSB><xsl:value-of select="count(/inventory/controller[TYPE='USB Controller'])"/></NumUSB>
    <SchedaVideo><xsl:value-of select="/inventory/video/NAME"/></SchedaVideo>
    <RAMSchedaVideo><xsl:value-of select="/inventory/video/MEMORY"/></RAMSchedaVideo>
    <RisolSchedaVideo><xsl:value-of select="/inventory/video/RESOLUTION"/></RisolSchedaVideo>
  </Computer>
</xsl:template>

<xsl:template match="/inventory/software">
  <!--
  <Software identify="TipoSoftware" relation="">
    <Code></Code>
    <Description>Description</Description>
    <TipoSoftware><xsl:value-of select="NAME"/></TipoSoftware>
    <Categoria>LookUp</Categoria>
  </Software>
  -->
</xsl:template>

<xsl:template match="/inventory/storage">
  <MemoryUnit identify="Code,Description,Produttore,Tipo,Dimensione,Computer"
  relation="MemoryUnitComputer">
    <Code><xsl:value-of select="MODEL"/></Code>
    <Description><xsl:value-of select="DESCRIPTION"/></Description>
    <Produttore><xsl:value-of select="MANUFACTURER"/></Produttore>
    <Tipo><xsl:value-of select="TYPE"/></Tipo>
    <Dimensione><xsl:value-of select="DISKSIZE"/></Dimensione>
  </MemoryUnit>
</xsl:template>

```

```

    </MemoryUnit>
</xsl:template>

<xsl:template match="/inventory/controller" >
    <Controller identify="Code,Description,Tipo" relation="ControllerComputer">
        <Code><xsl:value-of select="NAME"/></Code>
        <!--<Produttore><xsl:value-of select="MANUFACTURER"/></Produttore -->
        <Description><xsl:value-of select="NAME"/></Description>
        <Tipo><xsl:value-of select="TYPE"/></Tipo>
    </Controller>
</xsl:template>

<xsl:template match="/inventory/drive">
    <Volume identify="Description,Code,Dimensione" relation="VolumeComputer">
        <Description><xsl:value-of select="TYPE"/></Description>
        <Code><xsl:value-of select="LETTER"/></Code>
        <FileSystem><xsl:value-of select="FILESYSTEM"/></FileSystem>
        <Dimensione><xsl:value-of select="TOTAL"/></Dimensione>
    </Volume>

</xsl:template>
<xsl:template match="/inventory/monitor">
    <Monitor identify="Description,Tipo" relation="Collegamento">
        <Code><xsl:value-of select="MANUFACTURER"/></Code>
        <Description><xsl:value-of select="CAPTION"/></Description>
        <Tipo><xsl:value-of select="CAPTION"/></Tipo>
        <Dimensione></Dimensione>
    </Monitor>
</xsl:template>

<xsl:template match="/inventory/bios">
</xsl:template>
<xsl:template match="/inventory/input">
</xsl:template>
<xsl:template match="/inventory/network">
</xsl:template>
<xsl:template match="/inventory/memory">
</xsl:template>
<xsl:template match="/inventory/slot">
</xsl:template>
<xsl:template match="/inventory/video">
</xsl:template>
<xsl:template match="/inventory/port">
</xsl:template>
<xsl:template match="/inventory/printer">
</xsl:template>
<xsl:template match="/inventory/sound">
</xsl:template>
    <xsl:template match="/inventory/modem">
</xsl:template>

<!-- For CMDBUILD -->
<xsl:template match="/CMDBUILD/Computer">
    <Computer key="Code" identify="Code">
        <Code><xsl:value-of select="./Code"/></Code>
        <Marca><xsl:value-of select="./Marca"/></Marca>
        <Modello><xsl:value-of select="./Modello"/></Modello>
        <Description><xsl:value-of select="./Description"/></Description>
        <CPU><xsl:value-of select="./CPU"/></CPU>
        <NumCPU><xsl:value-of select="./NumCPU"/></NumCPU>
        <DimRAM><xsl:value-of select="./DimRAM"/></DimRAM>
        <NumHD><xsl:value-of select="./NumHD"/></NumHD>
        <DimTotHD><xsl:value-of select="./DimTotHD"/></DimTotHD>
        <SchedaRetel><xsl:value-of select="./SchedaRetel"/></SchedaRetel>
    </Computer>
</xsl:template>

```

```

        <SchedaRete2><xsl:value-of select="./SchedaRete2"/></SchedaRete2>
        <MacAddress1> <xsl:value-of select="./MacAddress1"/></MacAddress1>
        <MacAddress2> <xsl:value-of select="./MacAddress2"/></MacAddress2>
        <IndirizzoIP1><xsl:value-of select="./IndirizzoIP1"/></IndirizzoIP1>
        <IndirizzoIP2><xsl:value-of select="./IndirizzoIP2"/></IndirizzoIP2>
        <NumSeriali><xsl:value-of select="./NumSeriali"/></NumSeriali>
        <NumParallele><xsl:value-of select="./NumParallele"/></NumParallele>
        <NumUSB><xsl:value-of select="./NumUSB"/></NumUSB>
        <SchedaVideo><xsl:value-of select="./SchedaVideo"/></SchedaVideo>
        <RAMSchedaVideo><xsl:value-of select="./RAMSchedaVideo"/></RAMSchedaVideo>
        <RisolSchedaVideo><xsl:value-of select="./RisolSchedaVideo"/></RisolSchedaVideo>
    </Computer>
</xsl:template>

<xsl:template match="/CMDBUILD/Software">
<!--
    <Software identify="TipoSoftware">
        <Code><xsl:value-of select="./Code"/></Code>
        <Description><xsl:value-of select="./Description"/></Description>
        <TipoSoftware><xsl:value-of select="./TipoSoftware"/></TipoSoftware>
        <Categoria><xsl:value-of select="./Categoria"/></Categoria>
    </Software>
-->
</xsl:template>

<xsl:template match="/CMDBUILD/MemoryUnit">
        <MemoryUnit identify="Code,Description,Produttore,Tipo,Dimensione"
relation="MemoryUnitComputer">
        <Code><xsl:value-of select="./Code"/></Code>
        <Description><xsl:value-of select="./Description"/></Description>
        <Produttore><xsl:value-of select="./Produttore"/></Produttore>
        <Tipo><xsl:value-of select="./Tipo"/></Tipo>
        <Dimensione><xsl:value-of select="./Dimensione"/></Dimensione>
    </MemoryUnit>
</xsl:template>

<xsl:template match="/CMDBUILD/Controller" >
    <Controller identify="Code,Description,Tipo" relation="ControllerComputer">
        <Code><xsl:value-of select="./Code"/></Code>
        <Description><xsl:value-of select="./Description"/></Description>
        <Tipo><xsl:value-of select="./Tipo"/></Tipo>
    </Controller>
</xsl:template>

<xsl:template match="/CMDBUILD/Volume" >
    <Volume identify="Description,Code,Dimensione" relation="VolumeComputer">
        <Description><xsl:value-of select="."/></Description>
        <Code><xsl:value-of select="."/></Code>
        <FileSystem><xsl:value-of select="."/></FileSystem>
        <Dimensione><xsl:value-of select="."/></Dimensione>
    </Volume>

</xsl:template>
<xsl:template match="/CMDBUILD/Monitor" >
    <Monitor identify="Description,Tipo" relation="Collegamento">
        <Code><xsl:value-of select="./Code"/></Code>
        <Description><xsl:value-of select="./Description"/></Description>
        <Tipo><xsl:value-of select="./Tipo"/></Tipo>
        <Dimensione><xsl:value-of select="./Dimensione"/></Dimensione>
    </Monitor>
</xsl:template>

<xsl:template match="/CMDBUILD/bios">
</xsl:template>
    
```

```
<xsl:template match="/CMDBUILD/input">
</xsl:template>
<xsl:template match="/CMDBUILD/network">
</xsl:template>
<xsl:template match="/CMDBUILD/memory">
</xsl:template>
<xsl:template match="/CMDBUILD/slot">
</xsl:template>
<xsl:template match="/CMDBUILD/video">
</xsl:template>
<xsl:template match="/CMDBUILD/port">
</xsl:template>
<xsl:template match="/CMDBUILD/printer">
</xsl:template>
<xsl:template match="/CMDBUILD/sound">
</xsl:template>
<xsl:template match="/CMDBUILD/modem">
</xsl:template>
</xsl:stylesheet>
```