



CMDBuild

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

**Versione 0.8.0
Maggio 2008**

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)



Cogitek S.r.l. (www.cogitek.it)



CMDBuild è rilasciato con licenza GPL (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.....	6
Scopo del manuale.....	6
Aggiornamento schede da fonti dati esterne.....	7
Generalità.....	7
Criteri di implementazione.....	8
Wizard di configurazione dell'interfaccia.....	9
Script esecuzione periodica daemon di aggiornamento.....	16
OCS Inventory.....	17
Generalità.....	17
Definizione viste.....	20
DCL – Double Choco Latte.....	22
Generalità.....	22
Personalizzazioni disponibili.....	22
Alfresco.....	24
Utilizzo applicativo.....	24
Modalità di installazione dell'interfaccia.....	24
OpenOffice e MystiqueXML.....	25
Utilizzo applicativo.....	25
Modalità di installazione dell'interfaccia.....	25
Appendice.....	26
Attributi per definizione “mapping” OCS Inventory.....	26
Viste per database OCS Inventory.....	29
Esempio file XSLT di “mapping” per OCS Inventory.....	31

Introduzione

CMDBuild è una applicazione Open Source per la configurazione e gestione del database della configurazione (CMDB) degli oggetti e servizi informatici 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.

Il sistema consente la stampa di report tramite il motore open source JasperReports, sia report tabulari prodotti tramite un wizard interno, che report di elevata complessità ottenibili importando template disegnati tramite un editor visuale esterno. Il sistema di reportistica comprende anche la possibilità di ottenere stampe generate in Open Office tramite un apposito meccanismo di "stampa unione" con CMDBuild.

E' inoltre possibile allegare documenti, immagini ed altri file alle schede archiviate in CMDBuild ed operare ricerche full text sui documenti di testo (tramite integrazione con il diffuso sistema open source di Gestione Documenti Alfresco).

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.

E' infine disponibile un sistema di interfaccia con fonti dati esterne (database e server mail) per l'aggiornamento dati da altri sistemi, utilizzato in particolare per aggiornare l'inventario automatico dell'hardware rilevabile in rete (tramite integrazione con il sistema open source OCS Inventory).

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)
- il Modulo Gestione Dati, dedicato all'inserimento ed aggiornamento nel sistema dei dati descrittivi e delle relazioni funzionali fra le diverse entità, alla gestione di documenti allegati, alla produzione di report e tabulati, nonché alla definizione e controllo dei processi per la gestione dei servizi informatici

Ovviamente l'interfaccia per apportare modifiche strutturali al sistema sarà accessibile solamente agli utenti abilitati al ruolo di amministratore.

Il presente manuale è dedicato alla configurazione di connettori con sistemi esterni, sia relativamente alla attivazione del sistema di inventariazione automatica OCS Inventory, sia relativamente alla sincronizzazione di basi dati esterne, sia relativamente alla presa in carico di email finalizzate all'avvio di nuovi workflow.

Sono disponibili sul sito di CMDBuild (<http://www.cmdbuild.org>) manuali specifici dedicati a:

- Overview concettuale del sistema
- Administrator Manual
- User Manual
- Technical Manual
- Workflow Tutorial

Interoperabilità

I dati archiviati nel CMDB devono essere alimentabili, 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.

Nell'ottica dell'interoperabilità con altri sistemi sono stati implementati meccanismi di interfaccia con applicazioni e database esterni, utilizzate in particolare (ma non solo) per integrare funzionalità aggiuntive tramite applicazioni open source specializzate in attività connesse a CMDBuild:

- Automatic Inventory (OCS Inventory)
- Help Desk (DCL - Double Choco Latte)
- Document Management (Alfresco)
- Stampa unione (Open Office e MystiqueXML)

Questo nell'ambito di una filosofia complessiva volta a:

- estendere le funzionalità di CMDBuild riutilizzando 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/>)
- OpenOffice e MystiqueXML (<http://www.openoffice.org/> e <http://mystiquexml.yacme.com/>)

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

Aggiornamento schede da fonti dati esterne

Generalità

Presso gli enti che già lo utilizzano sono state individuate necessità diverse di aggiornamento periodico (ad esempio notturno) del database di CMDBuild a partire da fonti dati esterne.

Tali necessità riguardano in prima battuta la gestione dell'inventario automatico dei cespiti (vedi anche capitolo successivo), ma possono interessare ad esempio il database del personale (alimentato dal sistema gestionale dell'ente) oppure l'archivio dei cespiti (con aggiornamento di alcune informazioni a partire dal database contabile dei cespiti gestito dal personale amministrativo).

Si è quindi scelto di implementare un meccanismo generico, utilizzabile con alcuni accorgimenti in tutti i casi sopra descritti o ad essi simili.

Tale meccanismo è basato sui seguenti criteri e requisiti:

- la fonte dati esterna è accessibile dal server di CMDBuild e basata su uno dei database supportati (al momento MySQL, PostgreSQL, Oracle, SQLServer)
- l'aggiornamento viene effettuato su base periodica tramite un daemon generico fornito con CMDBuild
- il comportamento del daemon è guidato da un file di trasformazione XML (XSLT) opportunamente configurato e supportato da API specifiche sviluppate internamente a CMDBuild
- il daemon deve trovare nel database esterno "viste" opportune che utilizzerà come "catalogo" degli oggetti da interrogare
- più daemon possono essere schedulati contemporaneamente tramite utilizzo di file XSLT diversi
- la configurazione del file XSLT è effettuabile in modo guidato tramite un wizard visuale disponibile nel Modulo Schema
- le attività di aggiornamento sono basate su un processo di gestione (in ottica ITIL un workflow di Change Management) che ne garantisce l'espletamento in modo controllato (tramite meccanismi di approvazione esplicita da parte di un utente responsabile, tramite notifiche via mail, ecc)

Si segnala infine che lo stesso wizard implementato per la configurazione del file XSLT consente anche la configurazione di meccanismi di avvio automatico di nuove istanze di processi tramite mail. Non si tratta di attività di sincronizzazione dati, ma trattandosi comunque di un supporto alla interoperabilità viene trattato nell'ambito di questo stesso manuale.

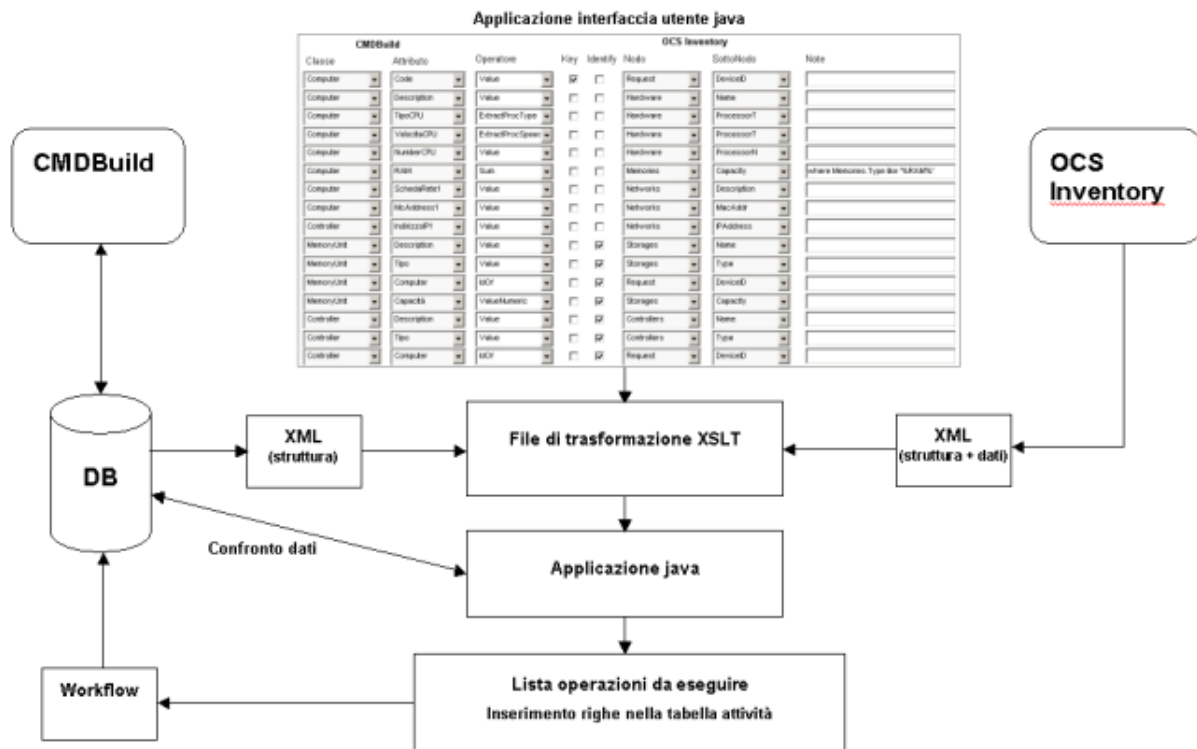
I criteri di "mapping" fra classi e relativi attributi presenti nei due sistemi sono personalizzabili (tramite il wizard sopra indicato) per consentire la definizione di criteri personalizzati in funzione delle necessità e della organizzazione interna di ogni ente.

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 dal database esterno (ad esempio quello di OCS Inventory) tramite viste opportune
- 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 esterno (la definizione completa per il caso di OCS Inventory 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:



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 nel database esterno ed acceduti attraverso le viste definite
- per ogni modifica riscontrata e tramite una opportuna API, attiva un'istanza di un processo semplificato di “Change Management” (nel rilascio di CMDBuild è compresa la definizione di un processo di esempio utilizzabile con OCS Inventory e denominato “ImportAsset”), che 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 modifica

che si intende applicare

- 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

Sempre a titolo esemplificativo è stata predisposta ed è fornita in ogni 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, a tale proposito si veda il capitolo successivo).

Wizard di configurazione dell'interfaccia

La configurazione dell'interfaccia con database esterni viene effettuata da una apposita funzione visuale del Modulo Schema.

La stessa funzione consente anche di configurare l'avvio di istanze di processi tramite ricezione di mail, con testo libero o con testo strutturato in sottocampi.

Nel caso si configuri l'interfaccia con una fonte dati esterna è richiesto che il database con cui collegarsi:

- risulti attivo ed accessibile sia durante le fasi di configurazione in corso di descrizione, che ovviamente durante le fasi di successiva operatività
- contenga specifiche viste prodotte secondo gli esempi riportati in appendice, fornendo a CMDBuild un “catalogo” delle tabelle e dei campi disponibili per l'interfacciamento

Il file di configurazione XSLT prodotto in entrambi i casi dal sistema al termine della presente procedura viene archiviato nella cartella nella sottocartella “xml” della cartella “ConnectorInventory”, dove sarà recuperato dall'apposito daemon che eseguirà le attività periodiche di aggiornamento.

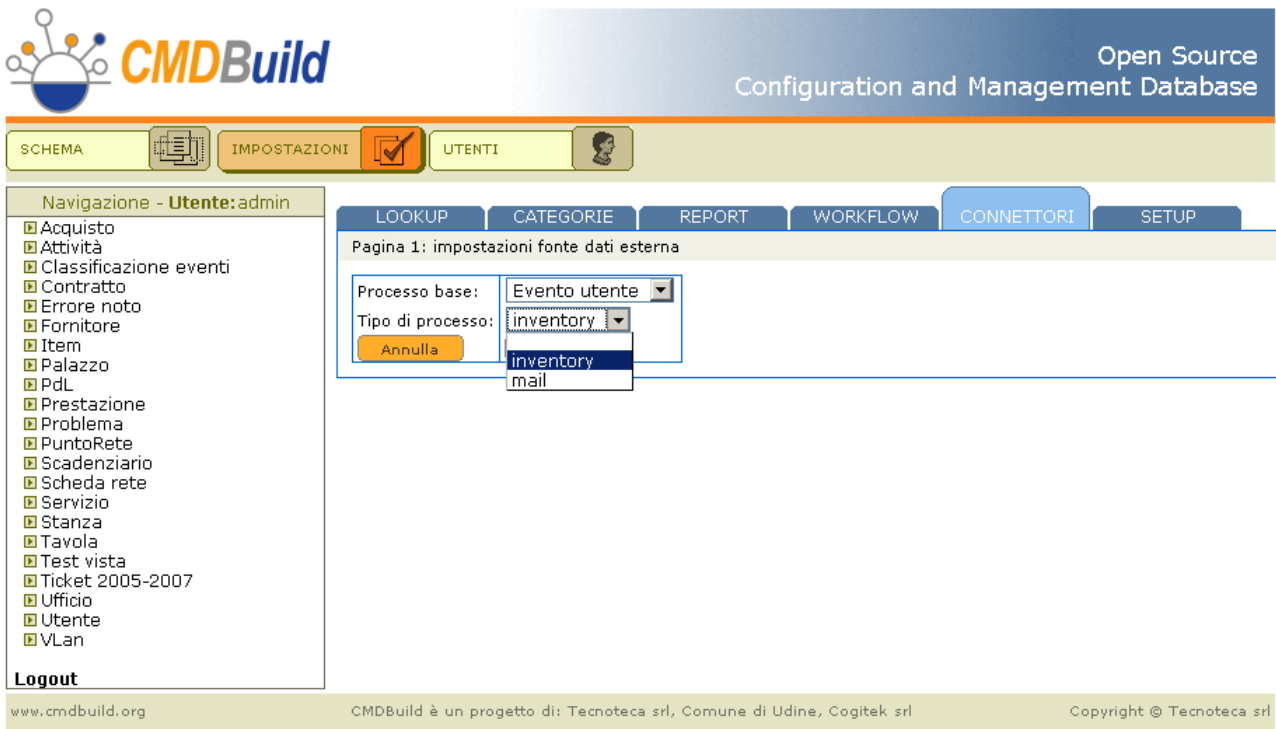
Per esigenze particolari e solamente da personale esperto il file XSLT può essere poi editato manualmente per eventuali integrazioni di ulteriori criteri non gestiti dal wizard implementato (filtri, vincoli, ecc).

Passaggio 1

In base al tipo di connettore che si vuole configurare, il primo passaggio proposto nel wizard richiede di associare un processo definito in CMDBuild con una fonte dati che:

- utilizzi quel processo per aggiornare i dati di CMDBuild in modo controllato (il processo richiederà cioè una autorizzazione all'aggiornamento o per lo meno una mail di notifica)
- avvii una nuova istanza del processo

Il primo utilizzo risolverà ad esempio la necessità di aggiornare in modo controllato i dati tecnici di un computer con il sistema di inventariazione automatica OCS Inventory, oppure di segnalare il personale cessato per disabilitarne gli account sui vari sistemi.



Il secondo utilizzo consentirà ad esempio di accettare un nuovo ticket di helpdesk ricevuto tramite mail prodotta con una form web preesistente, recuperando data, utente e titolo e decodificando poi gli eventuali attributi strutturati.

Passaggio 2a

Il secondo passaggio del wizard richiede di indicare i parametri di accesso al database esterno, ad esempio il database ove OCS Inventory raccoglie le informazioni rilevate sui computer gestiti:

- tipo di database (sono al momento supportati MySQL, PostgreSQL, Oracle, SQLServer)
- indirizzo del server su cui si trova il database esterno
- porta di funzionamento del database esterno
- nome del database interessato
- nome utente e password



SCHEMA IMPOSTAZIONI **UTENTI**

LOOKUP CATEGORIE REPORT WORKFLOW CONNETTORI SETUP

Pagina 2: impostazioni fonte dati esterna

Processo base: EventoUtente
 Tipo di database: MySQL
 Indirizzo del database: localhost
 Porta del database: 3306
 Nome del database: ocsinventory
 Utente del database: root
 Password del database: ****

Indietro Annulla Avanti

Logout

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

Passaggio 3a

Il terzo passaggio del wizard consente di correlare le categorie di informazioni presenti sul database esterno (ad esempio le tabelle del database di OCS Inventory) con le classi definite in CMDBuild.



SCHEMA IMPOSTAZIONI **UTENTI**

LOOKUP CATEGORIE REPORT WORKFLOW CONNETTORI SETUP

Pagina 3: mapping classi

Classe CMDBuild	Classe Master	Tabella esterna
Computer	<input checked="" type="radio"/>	device
Scheda rete	<input type="radio"/>	port
	<input type="radio"/>	
	<input type="radio"/>	
	<input type="radio"/>	

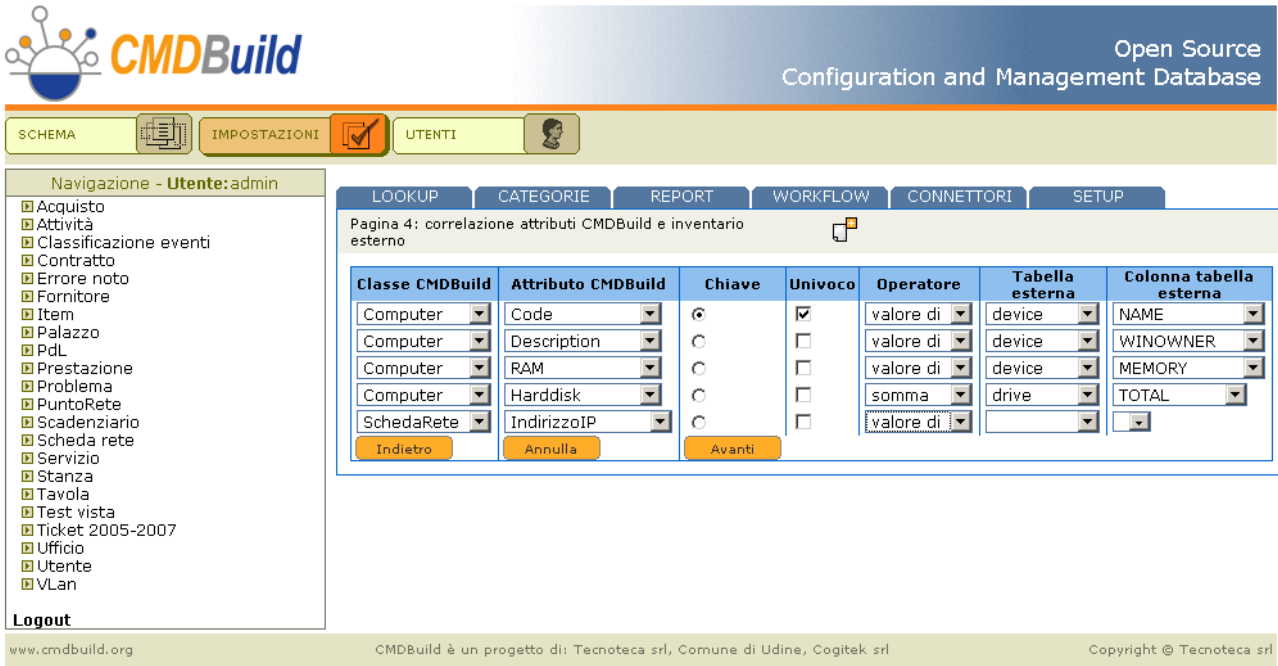
Indietro Annulla Avanti

Logout

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

Passaggio 4a

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



The screenshot shows the CMDBuild configuration interface. At the top, there's a navigation bar with 'SCHEMA', 'IMPOSTAZIONI', and 'UTENTI'. Below it, a sidebar lists various categories like 'Acquisto', 'Attività', etc. The main area is titled 'Pagina 4: correlazione attributi CMDBuild e inventario esterno'. It contains a table for defining mappings between CMDBuild attributes and external database attributes.

Classe CMDBuild	Attributo CMDBuild	Chiave	Univoco	Operatore	Tabella esterna	Colonna tabella esterna
Computer	Code	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>	valore di	device	NAME
Computer	Description	<input type="radio"/>	<input type="checkbox"/>	valore di	device	WINOWNER
Computer	RAM	<input type="radio"/>	<input type="checkbox"/>	valore di	device	MEMORY
Computer	Harddisk	<input type="radio"/>	<input type="checkbox"/>	somma	drive	TOTAL
SchedaRete	IndirizzoIP	<input type="radio"/>	<input type="checkbox"/>	valore di		

Buttons: Indietro, Annulla, Avanti.

La parte a destra della form è riservata a CMDBuild, la parte a sinistra al database esterno (ad esempio 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").

Lo stesso passaggio del wizard richiede anche di specificare quale attributo vada considerato come identificativo univoco del computer in CMDBuild.

Tale informazione è richiesta solamente per la classe principale (quella definita come "Master" al passaggio precedente) 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 5a

Il quinto 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.



The screenshot shows the CMDBuild web interface. At the top right, it says "Open Source Configuration and Management Database". The navigation bar includes "SCHEMA", "IMPOSTAZIONI", "UTENTI", and a user profile icon. A left sidebar lists various categories like "Acquisto", "Attività", "Contratto", etc. The main content area is titled "Pagina 5: relazioni le classi di CMDBuild" and contains a table with three columns: "Classe1", "Classe2", and "Dominio".

Classe1	Classe2	Dominio
Computer	SchedaRete	[Dropdown]
Indietro	Annulla	Avanti

At the bottom of the page, there is a footer with the website URL "www.cmdbuild.org", a copyright notice "Copyright © Tecnoteca srl", and a statement "CMDBuild è un progetto di: Tecnoteca srl, Comune di Udine, Cogitek srl".

Passaggio 6a

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



SCHEMA IMPOSTAZIONI **UTENTI**

LOOKUP CATEGORIE REPORT WORKFLOW CONNETTORI SETUP

Pagina 6: salvataggio file

Il file di mapping fra CMDBuild e la fonte dati selezionata verrà memorizzato nella directory remota ./connectorInventory/xmlEventoUtente per poter essere utilizzato dal programma di aggiornamento.

Indietro Annulla Avanti

Logout

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

Passaggio 2b

Nel caso si stia configurando l'avvio automatico di un processo da ricezione di messaggio di mail, vengono richiesti i parametri di accesso al server di posta:

- protocollo (IMAP o POP3)
- indirizzo del server di posta
- porta di funzionamento del servizio di posta in ingresso
- account di posta da cui scaricare le mail e relativa password

SCHEMA IMPOSTAZIONI **UTENTI**

LOOKUP CATEGORIE REPORT WORKFLOW CONNETTORI SETUP

Pagina 2: impostazioni fonte dati esterna

Processo base:

Protocollo:

Indirizzo mailserver:

Porta POP/IMAP:

Indirizzo di posta :

Password :

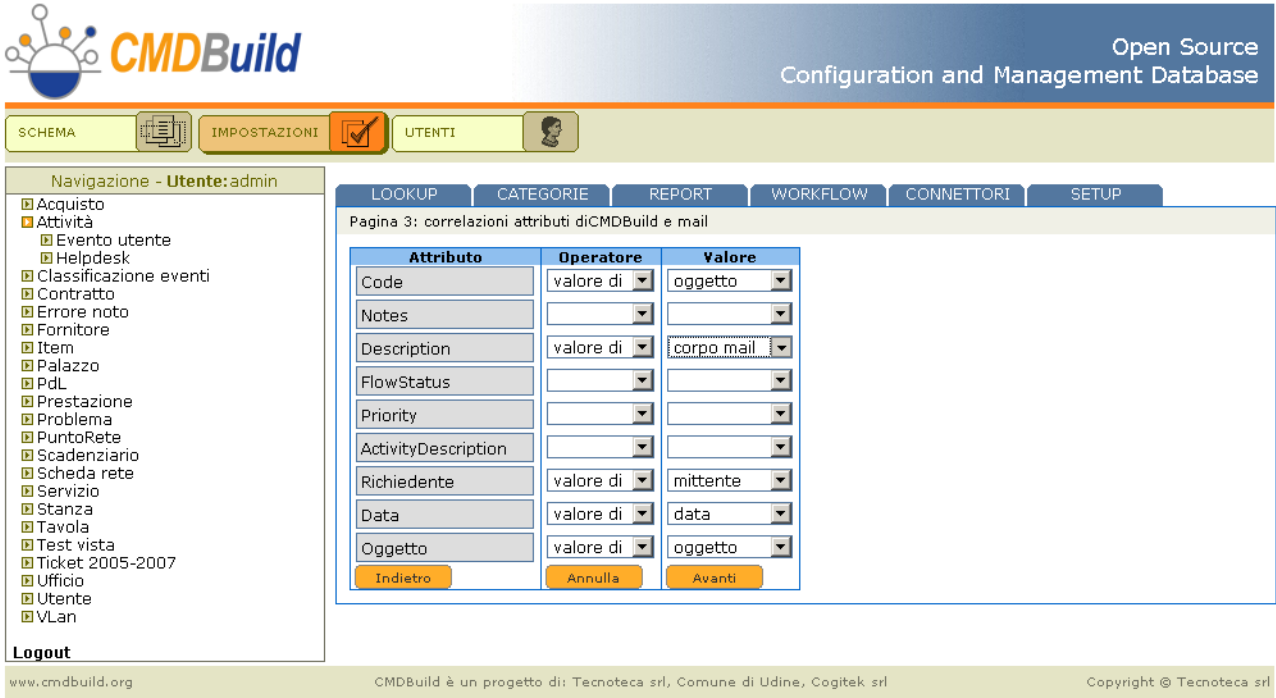
Indietro Annulla Avanti

Logout

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

Passaggio 3b

Il terzo passaggio comprende l'effettiva definizione dei criteri di "mapping" fra i due insiemi di informazioni da trattare.



E' possibile utilizzare l'operatore "custom" se si vuole associare a un attributo di CMDBuild un sottocampo codificato nel corpo della mail (purchè questa utilizzi il formato previsto da CMDBuild).

A titolo di esempio, se il corpo della mail è strutturato nella forma:

<cmdbuild:descrizioneproblema>bloccante</cmdbuild:descrizioneproblema>

<cmdbuild:tipoproblema>bloccante</cmdbuild:tipoproblema>

<cmdbuild:categoriaproblema>software gestionale</cmdbuild:categoriaproblema>

possiamo configurare tre ulteriori righe associando:

Campo CMDBuild	Operatore	Campo mail
"Descrizione"	"Custom"	"descrizioneproblema"
"Effetto"	"Custom"	"tipoproblema"
"Categoria"	"Custom"	"categoriaproblema"

Si segnala che:

- il processo di gestione mail elimina tutti i tag html, quindi eventuali testi formattati perderanno la formattazione ed i colori
- il processo elimina le mail dalla casella di posta da cui legge, non essendo tutti i protocolli in grado di marcarle come lette

Script esecuzione periodica daemon di aggiornamento

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
```


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 utilizza gli stessi meccanismi descritti al capitolo precedente per la gestione di aggiornamenti da fonti dati esterne.

Viene quindi effettuato:

- il 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 XLST)
 - API specifiche per l'interrogazione del database di CMDBuild
- l'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.

La lista completa delle informazioni rese disponibili da OCS Inventory e quindi mappabili sulle classi ed attributi di CMDBuild è riportata in appendice.

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:



SCHEMA IMPOSTAZIONI UTENTI

Navigazione - Utente: admin

- Attività
 - ImportAsset
 - Asset
 - Centro di costo
 - Contratto di manutenzione
 - Controller
 - Fattura
 - Fornitore
 - Licenza
 - Memory unit
 - Personale
 - Scadenziario
 - Struttura
 - Ubicazione
 - Volume

Logout

CLASSI ATTRIBUTI DOMINI

Ricerca scegli...

Nome ImportAsset

Descrizione Importazione dati da

Superclasse

Eredita da Attività

Gestore activity

Stato Attivo

conferma annulla

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.

Inserisci un nuovo attributo in ImportAsset

Nome accept

Descrizione Accetta cambiamenti

Tipo Boolean

Obbligatorio No

Default

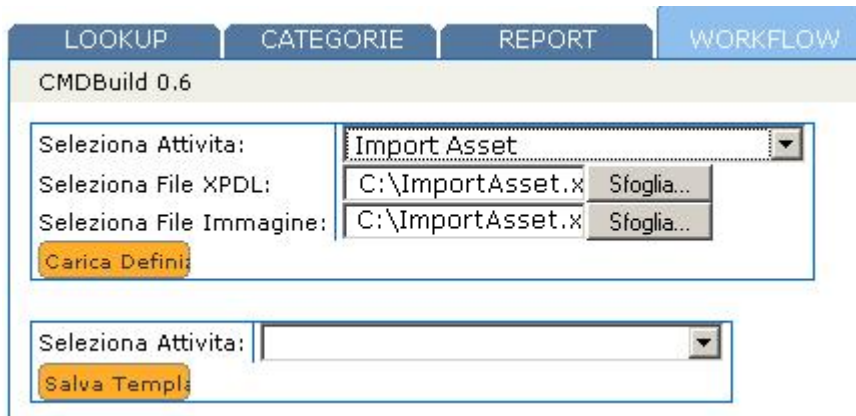
Unico

BaseDSP

Stato Attivo

conferma annulla

- 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.

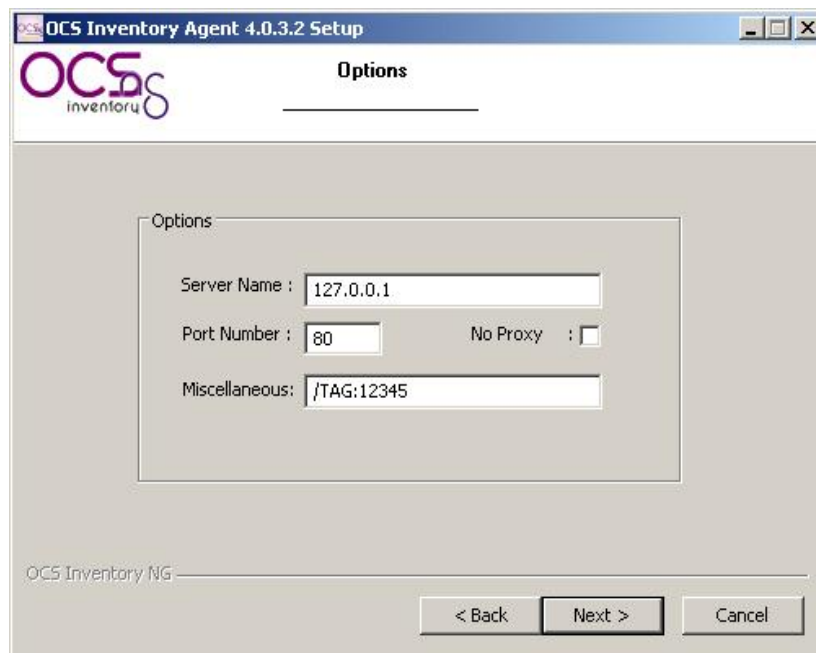
Configurazione OCS Inventory

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.

Definizione viste

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

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
```

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.

Pur disponendo CMDBuild già al suo interno di un sistema di workflow in grado di descrivere processi di gestione tramite un meccanismo "general purpose" basato sul linguaggio XPDL, possono essere talvolta ricercate soluzioni di "trouble ticketing" specifiche.

Il vantaggio della soluzione "interna" è quello di avere una integrazione completa fra le diverse informazioni trattate nel sistema, utili anche ai fini delle attività di ricerca e reporting (la scheda di un ticket può visualizzare ad esempio le relazioni con l'utente che l'ha aperto, l'utente che lo ha risolto, l'asset interessato, ecc, come viceversa le schede di un utente o di un asset possono visualizzare le relazioni con i relativi ticket aperti).

Con la soluzione "interna" sono stati implementati presso utilizzatori del sistema, in ottica strettamente ITIL3, i principali processi di Service Desk: gestione incidenti, richiesta servizi, richiesta informazioni, richieste di modifica standard e non standard, ordini di lavoro, solleciti).

Qualora si ritenesse preferibile trattare l'Help Desk nell'ambito di applicazioni esterne a CMDBuild, viene resa disponibile l'interfaccia con il sistema open source DCL, 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?: **Tipo:**

Canale:

Sommario:

Oggetto:

Allega File:

Soluzione:

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

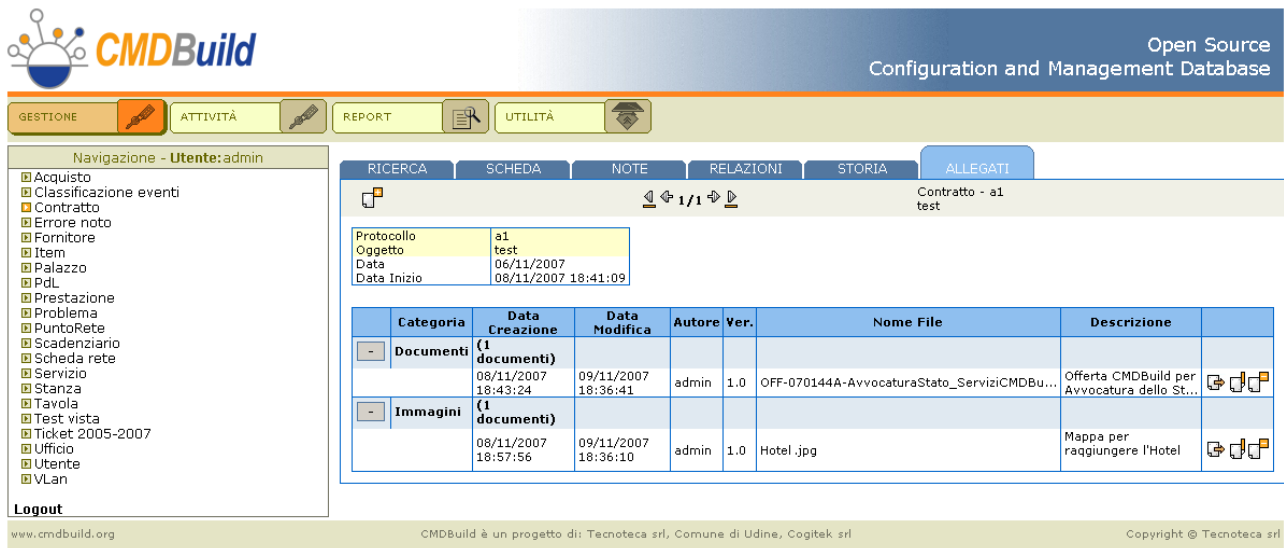
La versione modificata di DCL è resa disponibile con il codice sorgente a chi ne faccia richiesta tramite la pagina dei contatti presente sul sito del progetto CMDBuild.

Alfresco

Utilizzo applicativo

L'utilizzo applicativo delle funzionalità di gestione di documenti allegati alle schede di CMDBuild è descritto in un apposito paragrafo dello "User Manual".

Si riporta di seguito lo screenshot esemplificativo:



The screenshot displays the CMDBuild web interface. At the top, there is a navigation bar with the logo and the text "Open Source Configuration and Management Database". Below this is a menu with options: GESTIONE, ATTIVITÀ, REPORT, and UTILITÀ. The main content area is divided into a left sidebar for navigation (listing various document types like Acquisto, Contratto, etc.) and a central panel. The central panel has tabs for RICERCA, SCHEDA, NOTE, RELAZIONI, STORIA, and ALLEGATI. The ALLEGATI tab is active, showing a document titled "Contratto - a1 test". Below the document details, there is a table listing attached files:

	Categoria	Data Creazione	Data Modifica	Autore	Ver.	Nome File	Descrizione
-	Documenti (1 documenti)	08/11/2007 18:43:24	09/11/2007 18:36:41	admin	1.0	OFF-070144A-AvvocaturaStato_ServiziCMDBu...	Offerta CMDBuild per Avvocatura dello St...
-	Immagini (1 documenti)	08/11/2007 18:57:56	09/11/2007 18:36:10	admin	1.0	Hotel .jpg	Mappa per raggiungere l'Hotel

At the bottom of the interface, there is a footer with the website URL "www.cmdbuild.org", the project information "CMDBuild è un progetto di: Tecnoteca srl, Comune di Udine, Cogitek srl", and the copyright notice "Copyright © Tecnoteca srl".

Modalità di installazione dell'interfaccia

Le modalità di installazione e configurazione dell'interfaccia sono riportate in appositi paragrafi del "Technical Manual".

OpenOffice e MystiqueXML

Utilizzo applicativo

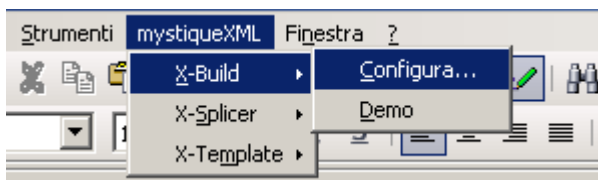
L'utilizzo applicativo delle funzionalità di produzione di stampe unione fra modelli di OpenOffice e dati estratti da CMDBuild è descritto in appositi paragrafi dell' "Administrator Manual" (importazione modello compilato) e dello "User Manual" (definizione set dati da utilizzare e descrittore da esportare).

Modalità di installazione e configurazione dell'interfaccia

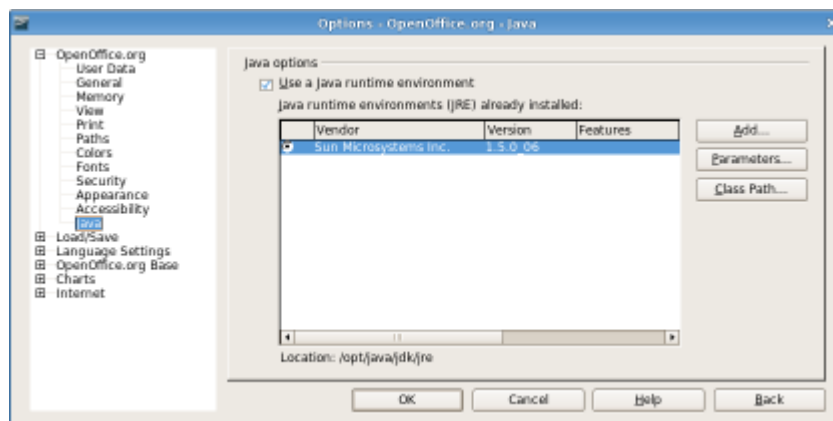
Le modalità di installazione lato server sono riportate in un apposito paragrafo del "Technical Manual", ricordiamo che è richiesta l'installazione sullo stesso server di CMDBuild della suite Open Office versione 2.3 o superiori e che l'istanza di Open Office deve essere raggiungibile sulla porta standard 8100.

Relativamente alla installazione lato client è richiesto di:

- installare la suite Open Office versione 2.3 o superiori
- installare il plugin MystiqueXML, presente nel file di download di CMDBuild o recuperabile dal sito <http://mystiquexml.yacme.com/> (su windows doppio click sull'eseguibile)
- configurare il plugin tramite il menu mystiqueXML => X-Build => Configura (confermare tutti i passaggi del wizard con il pulsante "Avanti" e premere "OK" alla fine)



Nel caso si riscontrino errori nel funzionamento del plugin può essere utile in alcuni casi intervenire sulle impostazioni di Open Office modificando i riferimenti alla JRE da utilizzare:



Per altre necessità e per disporre di informazioni più dettagliate sul funzionamento del plugin MystiqueXML si consiglia di consultare la documentazione presente sul sito del progetto (<http://mystiquexml.yacme.com/>).

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>
    <SchedaRete1><xsl:value-of select="/inventory/network[position()=1]/TYPE"/></SchedaRete1>
    <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>
```