



# **CMDBuild**

***Open Source Configuration and Management Database  
Technical Manual***

**Versione 0.5.0  
Settembre 2006**

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
Installazione del sistema.....	5
Requisiti software.....	5
Installazione del database.....	6
Installazione dell'applicazione.....	6
Configurazione dell'applicazione.....	6
Test di funzionamento dell'applicazione.....	7
Struttura dell'applicazione.....	9
Generalità.....	9
Diagrammi UML.....	9
Progettazione del database.....	11
Criteri di base.....	11
Descrittori integrativi.....	11
Utilizzo avanzato di PostgreSQL.....	11
Stored Procedures.....	11
Viste.....	11
Schema entità relazioni.....	11
Attuali limitazioni note del sistema.....	12
Export / Import.....	12
Campi tipo "reference".....	12

# 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.

## 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, nonché alla produzione di report e tabulati.

Il presente manuale è dedicato ai tecnici informatici cui sono demandate le attività di installazione delle componenti software e di amministrazione del database.

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

- Overview concettuale del sistema
- Administrator Manual
- User Manual

# Installazione del sistema

L'installazione di CMDBuild richiede l'utilizzo di uno o più server su cui suddividere le componenti logiche costitutive del sistema:

- server web
- componenti di elaborazione
- database

Vengono descritti di seguito i requisiti software richiesti da CMDBuild, le modalità di installazione e configurazione, i criteri di verifica e test.

Nella progettazione dell'infrastruttura sistemistica va considerato che l'attivazione di applicazioni web come quella in oggetto richiede la disponibilità di componenti hardware e di rete dotate di adeguati livelli di sicurezza, sia rispetto accessi esterni indesiderati (firewall, DMZ) che rispetto le esigenze di disponibilità continuativa on line del sistema (backup della linea internet, mirroring hardware) e di adeguate prestazioni di accesso.

## Requisiti software

L'installazione di CMDBuild richiede la presenza dei componenti software di seguito elencati.

### Sistema operativo

Qualunque sistema operativo supporti gli applicativi sotto elencati (incluso Linux).

### Database

PostgreSQL 8.0 o superiore, l'installazione base è sufficiente, accertarsi che sia attivato il supporto al linguaggio "plpgsql".

Sito di riferimento: <http://www.postgresql.org/>

### Web Server

CMDBuild richiede l'installazione di Apache 2.0 e di Jakarta Tomcat 4.3.31 o superiori.

Sito di riferimento: <http://www.apache.org/>

### Librerie Java

Le librerie Java sono necessarie per il funzionamento di Jakarta Tomcat.

CMDBuild richiede JDK 1.4 ma si consiglia l'utilizzo della versione 1.5.

Sito di riferimento: <http://www.sun.com/>

### Librerie già incluse

CMDBuild contiene una serie di librerie già all'interno del pacchetto di installazione, ed in particolare:

- postgresql-8.0-313.jdbc3.jar per il collegamento a PostgreSQL 8.0
- commons-dbcp-1.2.jar per la gestione del pool di connessione al database
- struts\* e commons\*, librerie utilizzate da Struts 1.2

Essendo CMDBuild basato su Struts all'interno del file WEB-INF sono presenti i relativi file di configurazione.

Ulteriori informazioni su Struts si trovano sul sito di riferimento: <http://struts.apache.org/>

Tutti i software richiesti sono Open Source (ad eccezione eventualmente del Sistema Operativo se si optasse per una soluzione diversa da Linux).

## Installazione del database

Per l'installazione del database vanno effettuate le seguenti operazioni:

- tramite uno strumento con interfaccia grafica (ad esempio pgAdmin3 di PostgreSQL) o da linea di comando creare il database utilizzando il nome "cmdb" o altro equivalente:

```
CREATE DATABASE cmdb
WITH OWNER = postgres
ENCODING = 'UNICODE'
TABLESPACE = pg_default;
```

- accertarsi che l'encoding sia UNICODE
- eseguire i comandi SQL contenuti nei file:
  - CMDB - creazione schema.txt  
se si è creato il database tramite interfaccia grafica (ad es. PGAdmin) accertarsi che non esista nessun language associato al database, in caso contrario commentare le righe:

```
-- CREATE TRUSTED PROCEDURAL LANGUAGE 'plpgsql'
-- HANDLER plpgsql_call_handler;
```
  - CMDB - inserimento dati.txt
  - CMDB - creazione triggers-funzioni trigger-aggiornamento sequence.txt

## Installazione dell'applicazione

Per l'installazione del database è sufficiente copiare e decomprimere all'interno della directory Webapps di Tomcat il file scaricato dal sito di CMDBuild.

## Configurazione dell'applicazione

Per configurare CMDBuild deve essere editato il file WEB-INF\struts-config.xml, in particolare relativamente ai data source configurati:

```
<!-- configuration for commons BasicDataSource -->
<data-sources>
<data-source key="cmdbSource" type="org.apache.commons.dbcp.BasicDataSource">
  <set-property
    property="driverClassName"
    value="org.postgresql.Driver" />
  <set-property
    property="url"
    value="jdbc:postgresql://localhost/cmdb" /> * CHANGE THIS *
```

```
<set-property
  property="username"
  value="postgresuser" /> * CHANGE THIS *
<set-property
  property="password"
  value="postgrepassword" /> * CHANGE THIS *
<set-property
  property="maxCount"
  value="50" />
</data-source>
</data-sources>
```

Devono essere modificati:

- l'indirizzo del server che ospita PostgreSQL
- il nome del database scelto in fase di installazione
- username e password necessari a CMDBuild per accedere al database.

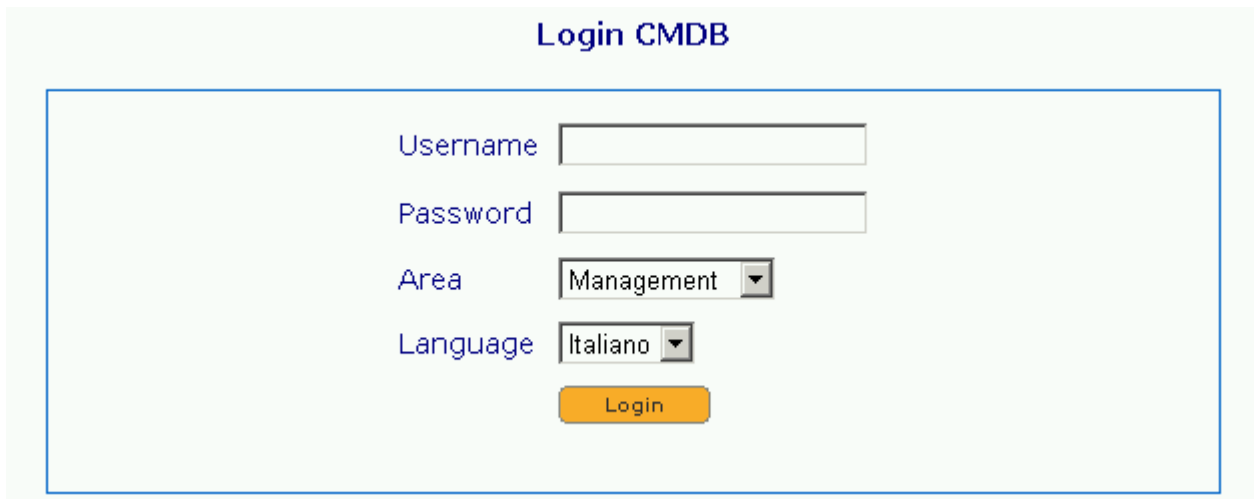
Una volta terminata la fase di configurazione deve essere riavviato Tomcat.

## Test di funzionamento dell'applicazione

Alla conclusione delle operazioni descritte, se tutto è stato fatto correttamente, si potrà aprire il browser e accedere all'indirizzo <http://localhost:8080/cmdb/>

Nel caso sia stata configurata durante l'installazione di Tomcat una porta diversa da quella standard, tale porta dovrà essere sostituita alla 8080 all'interno della URL sopra specificata.

Si presenterà la seguente schermata:



Autenticandosi con l'account:

Username: admin

Password: admin

si potrà a questo punto accedere all'applicazione CMDBuild.

I manuali disponibili sul sito (Overview Document, Administrator Manual, User Manual) forniranno il supporto necessario per un efficace utilizzo del sistema.

Nel caso si presentasse invece la seguente schermata:

## HTTP Status 404 - Servlet action is not available

**type** Status report

**message** Servlet action is not available

**description** The requested resource (Servlet action is not available) is not available.

Apache Tomcat/5.0.28

dovrebbe essere ipotizzato un errore di connessione con il database, CMDBuild non trova il database oppure sono sbagliati username e password per connettersi a PostgreSQL.

Si consiglia in tal caso di ritornare al paragrafo relativo alla configurazione di CMDBuild e di modificare il file struts-config.xml.

Nel log di Tomcat si possono trovare ulteriori informazioni sull'accaduto, quali ad esempio:

*GRAVE: Initializing application data source cmdbSource*

*org.apache.commons.dbcp.SQLNestedException: Cannot create PoolableConnectionFactory  
(Attivazione del backend fallita: FATAL: database "pippo" does not exist.)*



# Struttura dell'applicazione

## Generalità

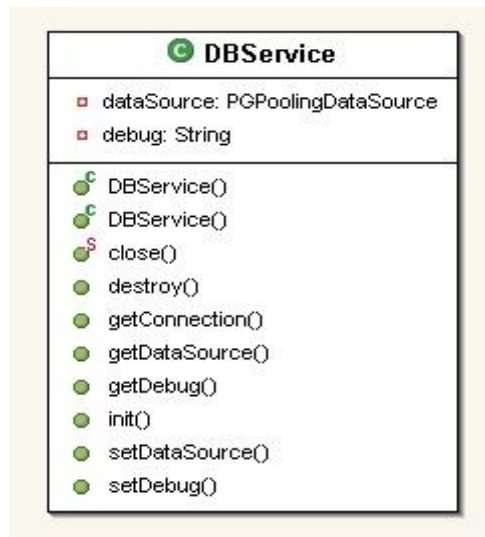
L'applicazione CMDBuild è stata realizzata nel linguaggio JAVA con utilizzo di pagine JSP per l'interfaccia web.

Gli strumenti individuati per la scrittura del software comprendono l'ambiente di sviluppo Eclipse e l'editor UML Omondo EclipseUML.

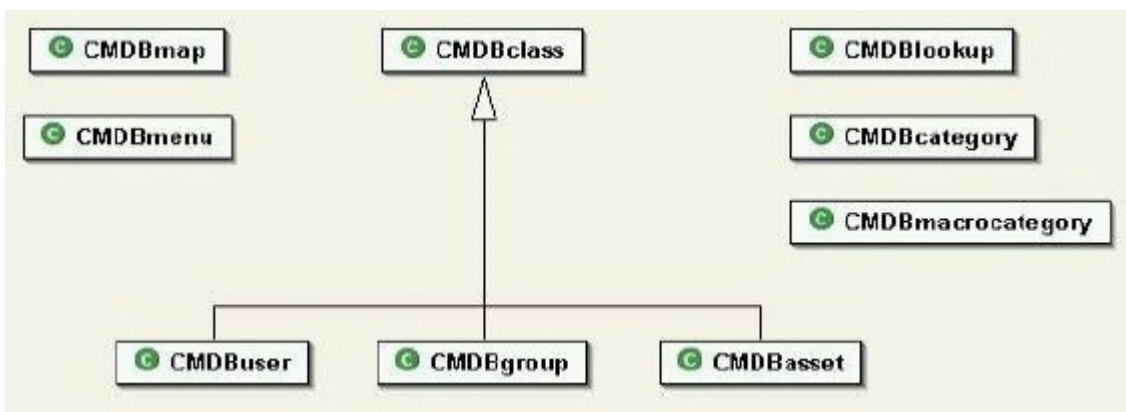
## Diagrammi UML

Si riportano di seguito alcuni diagrammi UML esemplificativi dell'architettura delle classi di CMDBuild.

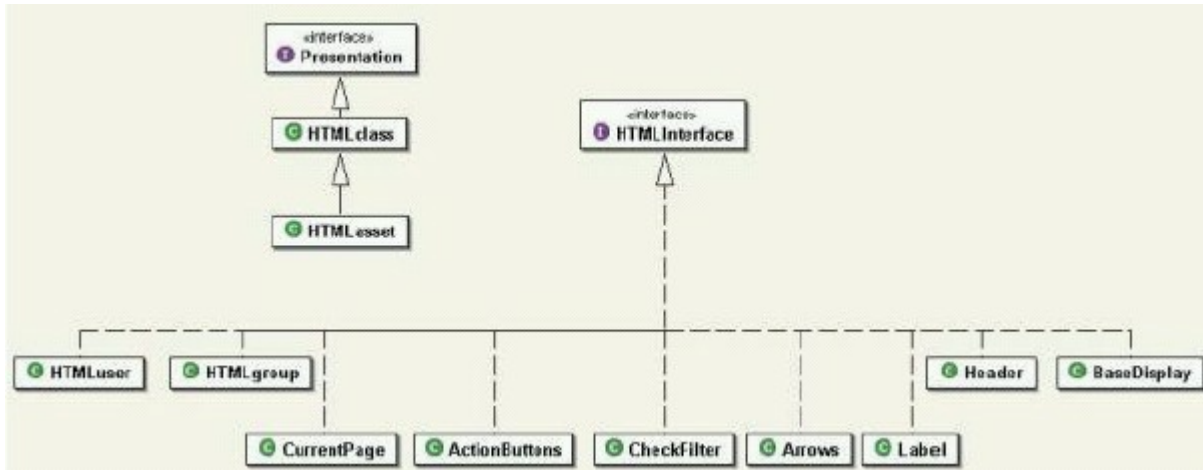
Il seguente diagramma descrive la classe delegata alla connessione al database:



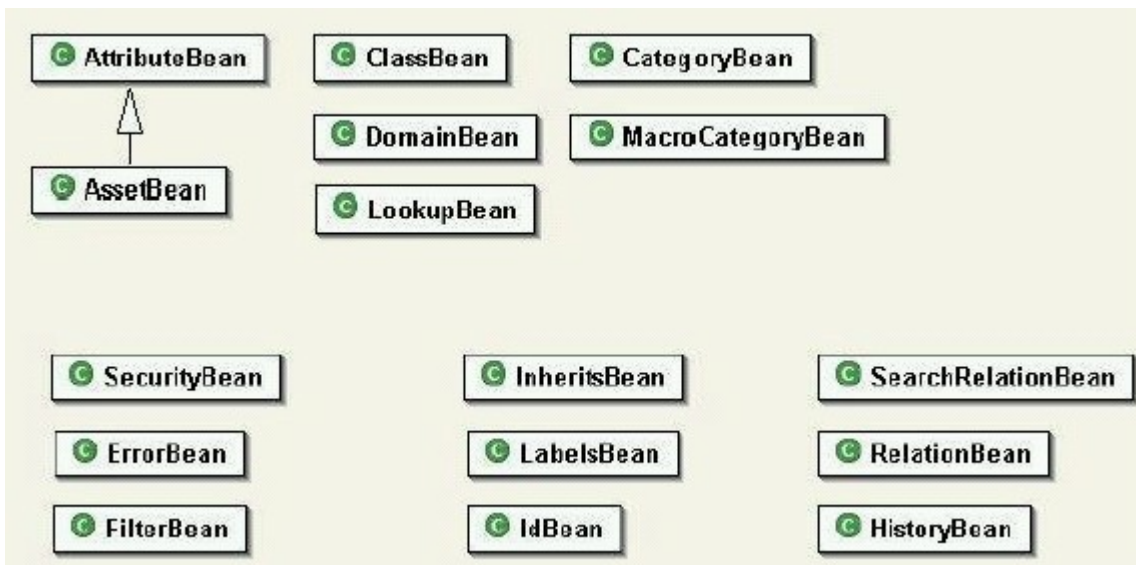
Segue il diagramma UML della classi e delle interfacce relativa alla gestione delle tabelle e dei domini nel database:



Il diagramma UML delle classi java delegate alla generazione delle schede di CMDBuild per l'editing dei dati è il seguente:



Gli oggetti sotto elencati sono invece utilizzati per il trasferimento delle informazioni all'interno di CMDBuild:



# Progettazione del database

## Criteri di base

...

*[da sviluppare]*

## Descrittori integrativi

...

*[da sviluppare]*

## Utilizzo avanzato di PostgreSQL

...

*[da sviluppare]*

## Stored Procedures

...

*[da sviluppare]*

## Viste

...

*[da sviluppare]*

## Schema entità relazioni

...

*[da sviluppare]*

# Attuali limitazioni note del sistema

## Export / Import

Si è riscontrato che in PostgreSQL il comando "pg\_dump", anche se lanciato con opzione "-o" salva (e consente a "pg\_restore" di ripristinare) gli "oid" delle righe delle tabelle, ma non salva (e quindi non consente il ripristino) gli "oid" delle classi all'interno del catalogo di sistema.

Essendo tali "oid" utilizzati da CMDBuild come propri riferimenti interni (classe "Class" e derivate, classe "Map" e derivate), a seguito del restore di un database risultano non corrette le relazioni stabilite fra gli oggetti di CMDBuild.

E' già stato realizzato e sarà reso disponibile nei prossimi giorni (e poi inserito in ogni rilascio) un comando di ripristino che va eseguito successivamente ad ogni "restore", che aggiorna i riferimenti in questione a partire da informazioni che sono comunque disponibili nel sistema (sottolineiamo a questo proposito che non esiste alcun rischio di perdita di dati).

## Campi tipo "reference"

La struttura di CMDBuild consente la definizione di campi di tipo "reference" all'interno di una superclasse ed il loro utilizzo nell'ambito delle classi derivate.

Non è al momento gestito il caso di campi "reference" ereditati lungo catene "superclassi – sottoclassi" di lunghezza superiore a due (non è quindi ad esempio possibile definire un campo "reference" all'interno della classe "asset" ed utilizzarlo nella sottoclasse "notebook" se esiste una ulteriore superclasse "computer" fra "asset" e "notebook").

Sempre relativamente ai campi di tipo "reference" non è al momento gestita l'opzione di cancellazione in modalità "cascade", cioè con eliminazione automatica di tutti gli elementi collegati anche su più livelli al record da cancellare (ad esempio non si può cancellare la scheda di un "fornitore" se esiste una scheda "contatti" con un campo di tipo "reference" sulla classe "fornitori" ed esistono dei contatti ancora attivi per quel fornitore).

E' quindi al momento necessario provvedere manualmente alla cancellazione degli elementi collegati prima di poter cancellare la scheda principale.

Tale limitazione è legata al fatto che le uniche cancellazioni ammesse in CMDBuild sono cancellazioni di tipo "logico", il che non consente di usufruire delle opzioni di cancellazioni previste in PostgreSQL per le "foreign key".