

# CMDB Import

---

## OVERVIEW

---

A CMDB (Configuration Management Database) is a repository of information which gathers and links components of an information system, thus enabling a better understanding of their organization.

Elements of a CMDB are named configuration items (CI). A configuration item is an instance of entity with modifiable attributes.

In MEGA 2009 SP5 R7, a tool allows you to import CMDB assets into a MEGA repository.

### Import format

Objects to import must be described in an XML file whose structure complies with the format defined by the XSD file supplied by MEGA.

The MEGA schema allows importing configuration items from different CMDB tools (BMC, HP, etc.) and thus avoiding discrepancies between versions.

### Imported objects

The objects you can import into MEGA are:

- CMDBs
- Configuration items
- Configuration item types

### CMDB

All elements imported are linked to a CMDB. A CMDB contains configuration items and their type.

Name of the CMDBs contained in the imported file appears in the import wizard. If no CMDB is specified in the imported file, the wizard automatically creates it.

### Configuration items

The different components imported into MEGA are created in the form of configuration items. Then you can map them with MEGA Architecture objects (Servers, workstations, artifacts, physical assets, etc.). See “Mapping CI with MEGA Architecture components”, page 8.

A configuration item can contain other configuration items. The link between an item and its sub-item is represented by a Configuration Item Composition.

#### [Configuration items properties](#)

The XML file must provide the name of the items to import as well as their properties.

If certain properties do not exist in MEGA, you can define them in the XML file so that MEGA can create them and link them to the appropriate items.

New properties appear in a dedicated tab of the item property window. See [example](#) page 12.

### Identifier

Configuration items imported into MEGA have a CMDB identifier. MEGA memorizes this identifier in order to recognize all configuration items. When importing items, the wizard checks if they already exist in the MEGA repository so that it can either create it or update it.

The CMDB identifier is displayed in the property window of the configuration item, under the **Characteristics** tab.

### Status

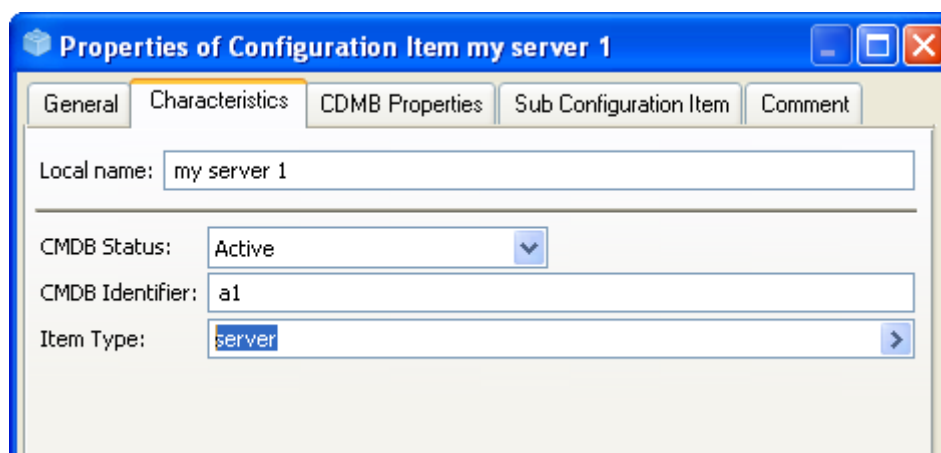
As an indication, you can give the **Status** of the configuration item:

- Active
- Deprecated
- Inactive

Contrary to the Delete option, the status definition has no impact on the item. See « Deleted objects», page 4.

## Configuration item type

A configuration item is associated to a type. In MEGA, the configuration item type is displayed in the item property window, under the **Characteristics** tab.



## Deleted objects

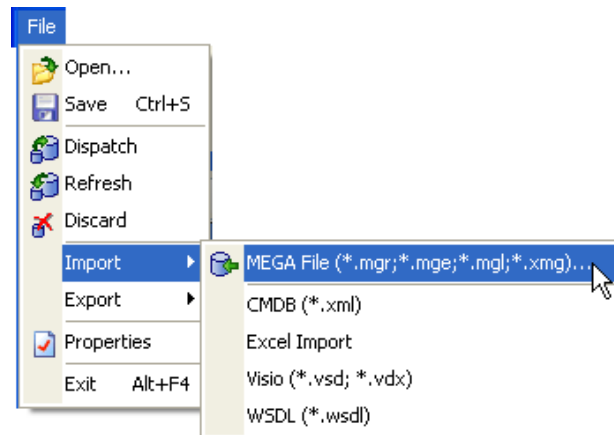
In the XML file you can specify that certain items already imported into MEGA must be deleted (because they have been removed from the CMDB source for example). To delete them in MEGA, you must check the **Delete in MEGA the objects marked for deletion in XML file** option in the import wizard.

## IMPORTING CMDB DATA INTO MEGA

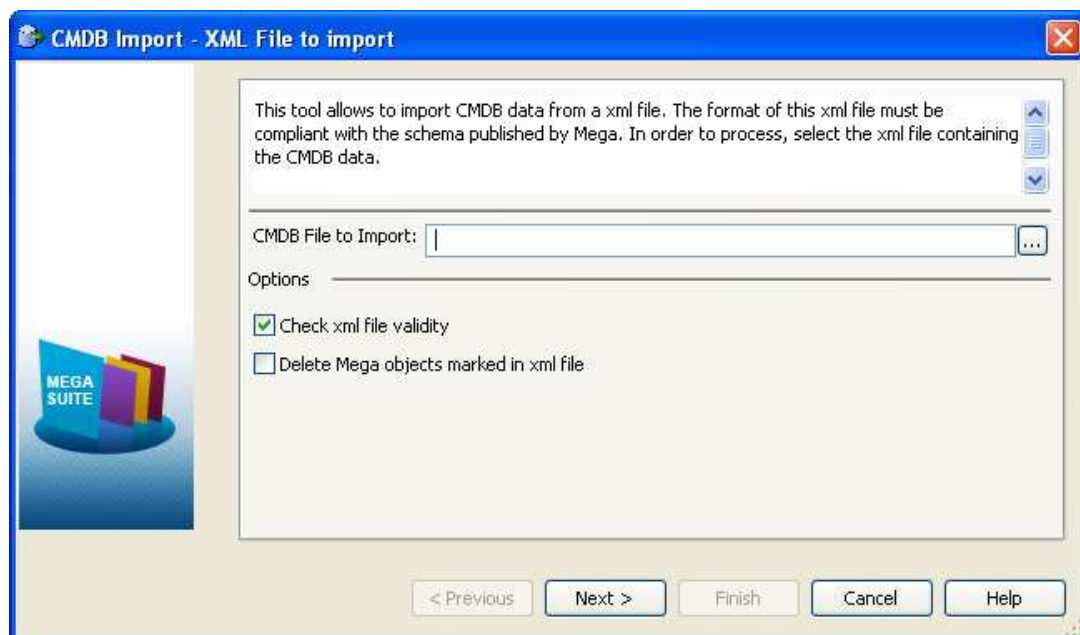
---

To import CMDB data into MEGA:

1. In the MEGA Desktop, click File > Import > Cmdb (\*.xml).



The CMDB Import wizard appears.



2. Click the **Browse** button  and select the file to be imported.

Two options are available:

- **Check XML file validity:** this option checks the validity of the file format which must follow the XML schema provided by MEGA. If the XML file is not correct, it cannot be imported. A message appears when the file is not valid. You can get more details by consulting the Error Log File. To access it, click the Help menu then **Technical Support > Error Log File > Edit**.
- **Delete in MEGA the objects marked for deletion in XML file:** if this option is checked, all CMDB items marked for deletion will be deleted in MEGA at the end of the import.

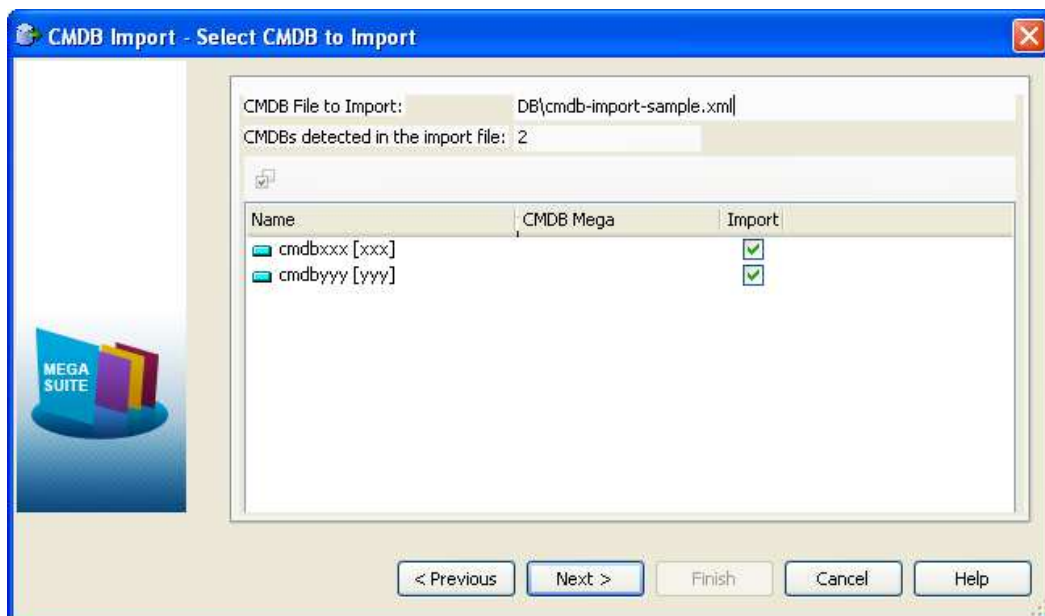
3. Click **Next**.

The Import wizard lists the different CMDBs identified in the file. Select those you want to import. If a CMDB has already been imported into MEGA, its name is indicated in the **CMDB MEGA** column.

---

A CMDB is a group of configuration items

---



4. Click **Next**.

A dialog box asks you to confirm the import process.

5. Click **Yes**.

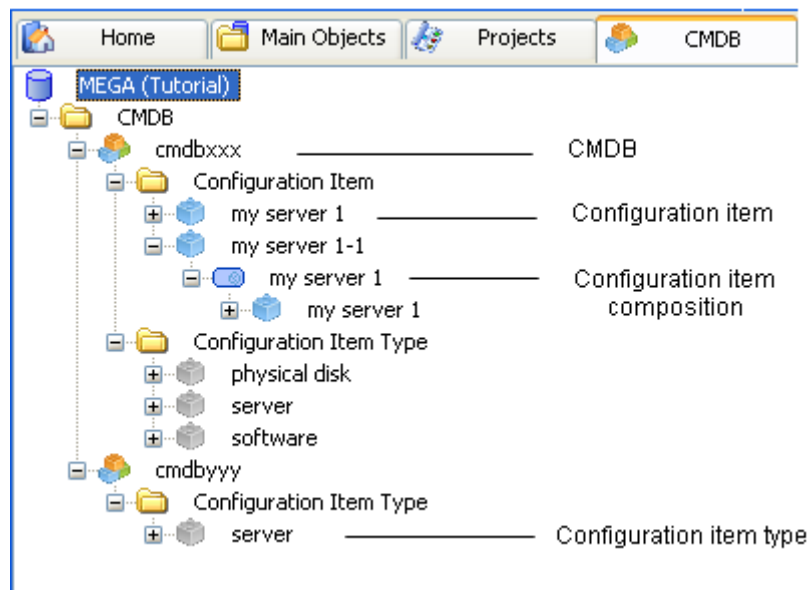
A report displays the result of the import. You can access the full report in the specified path.

Cmdb File To Import: 9 SP5 CP7\_R7\cmdbimportfilesample.xml  
cmdb's imported: 2/2  
Report Filepath: STRATION\DB\ADVENTURE\USER\USER\CmdbImport\_20120626\_18h43.txt  
Report Information:

6. Click **Finish**.

## Displaying CMDB configuration items

Configuration items imported into MEGA appear in a dedicated navigation tree.



## MAPPING CI WITH MEGA ARCHITECTURE COMPONENTS

---

Configuration items created in MEGA can be mapped with MEGA Architecture concepts. You can map them with elements of an application or elements of a resource architecture (Artifacts and Physical Assets).

Mapping between objects has to be done manually by the user in the mapping editor.

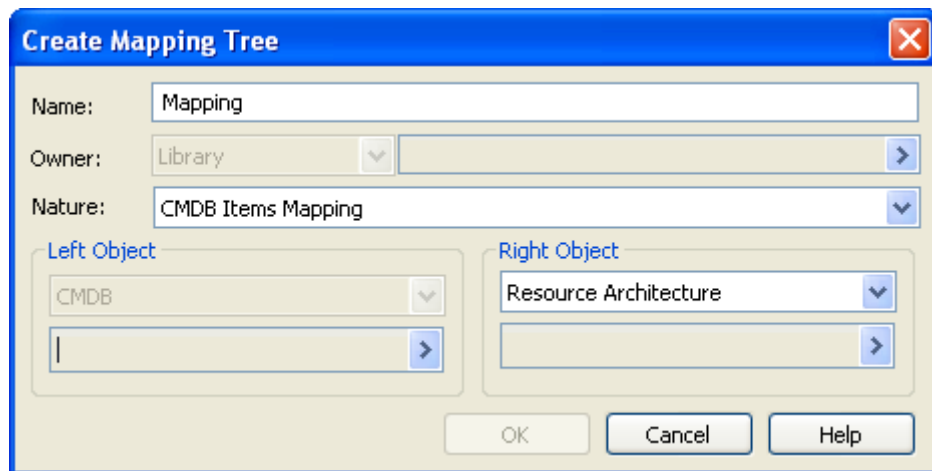
Property windows of MEGA Architecture concepts display a new page in which you can see the CMDB items with which they have been mapped.

### Mapping editor

To map a configuration item with a MEGA Architecture object:

1. In MEGA, click the menu **Tools > Mapping Editor**.
2. In the mapping editor, click **File > Create Mapping Tree**.

The **Create Mapping Tree** dialog box appears.



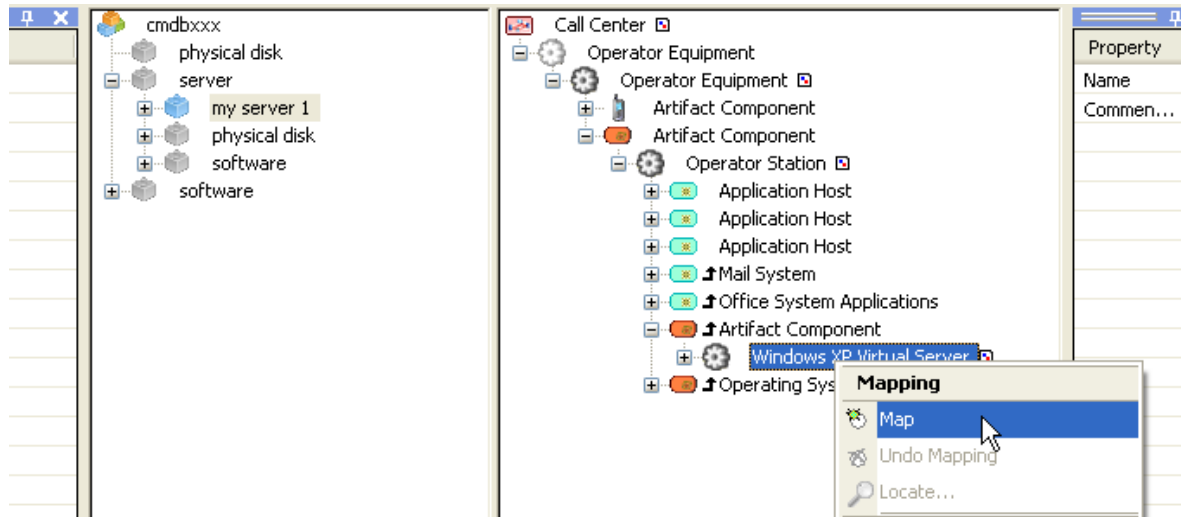
3. Select as **Nature** the "CMDB Elements Mapping".
4. In the **Left Object** frame, select the CMDB that contains the items to map.
5. In the **Right Object** frame, select the resource architecture that contains the architecture concepts.
6. Click **OK**.

The editor displays on the left the items of the CMDB, and on the right the components of the resource architecture.

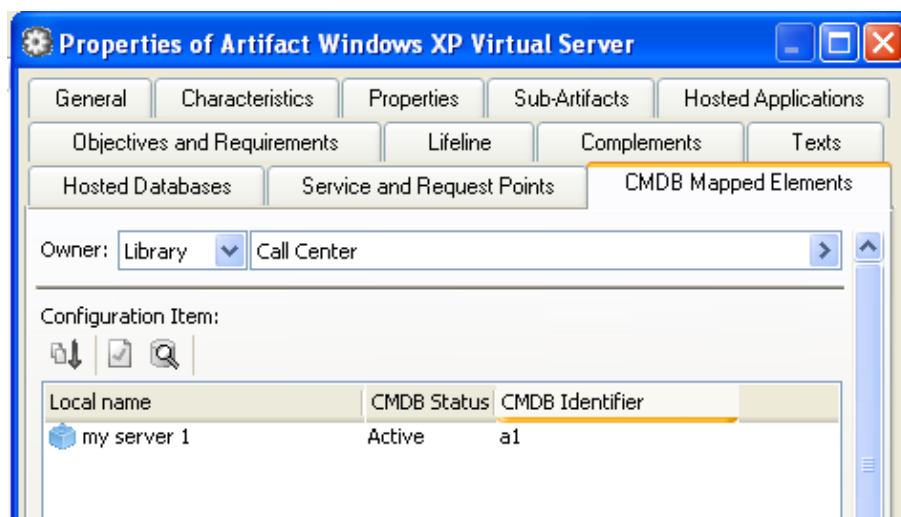


To create mapping between an item of the CMDB and a component of the resource architecture:

1. In the mapping editor, select an item on the left.
2. Select the equivalent object in the architecture on the right.
3. Open the pop-up menu of the second object and select **Map**.



The mapping is created from the last object selected. In the above example, “my server 1” is mapped with “Window XP Virtual Server”. The mapping appears in the properties of the “Window XP Virtual Server” artefact, under the **CMDB Mapped Elements** tab.




You can also display the mapped items in the resource architecture diagram.

You can display the mapped elements in the resource architecture diagram, by checking the **CMDB Elements** option in the diagram Views.

Views

Select All ☐

Communications	<input checked="" type="checkbox"/>
Interactions	<input checked="" type="checkbox"/>
Service and Request Points	<input checked="" type="checkbox"/>
Interactions and Communications	<input checked="" type="checkbox"/>
Architecture Use	<input checked="" type="checkbox"/>
Human Asset	<input checked="" type="checkbox"/>
Physical Asset	<input checked="" type="checkbox"/>
Resource Architectures	<input checked="" type="checkbox"/>
Sites	<input checked="" type="checkbox"/>
External References	<input checked="" type="checkbox"/>
Notes	<input checked="" type="checkbox"/>
..  CMDB Elements	<input checked="" type="checkbox"/>

## EXAMPLE OF XML FILE

---

The following extract is an example of an XML file containing CMDB elements to import.

First elements described in the file are CMDBs, which contain configuration item types and configuration items.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <cmdbroot xmlns="http://www.mega.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.mega.com cmdb.xsd">
- <cmdb id="xxx" cmdbid="xxx" name="cmdbxxx">
  <description />
- <types>
  - <ciType id="a123" cmdbid="azerty" name="server">
    <description>This is the comment of this configuration item</description>
    <properties>
      <property id="123_1" cmdbid="123_a" name="ip adress" type="string" length="15" />
      <property id="123_2" cmdbid="123_b" name="purchase date" type="date" format="yyyy/mm/dd" />
      - <property id="123_3" cmdbid="123_c" name="additional information" type="text">
        <description>This property is used to define additional information</description>
      </property>
    </properties>
  </ciType>
  - <ciType id="a456" superid="a123" cmdbid="qsdqgh" name="software" toto="45">
    <description>This is the comment of this configuration item</description>
  </ciType>
  <ciType id="a789" superid="a123" cmdbid="wxcvbn" name="physical disk" delete="no" />
</types>
- <data>
- <configurationItems>
  - <ci id="a" cmdbid="a1" name="my server 1" typeid="a123" status="active" delete="yes">
    <description>This server hosts IBM Rational Synergy</description>
    <propertyValue name="ip adress" propertyid="123_1">123.456.10.2</propertyValue>
    <propertyValue name="purchase date" propertyid="123_2">2009/04/15</propertyValue>
  </ci>
  - <ci id="b" cmdbid="b1" name="my server 1" typeid="a456" status="active" delete="no" tote="789">
    <propertyValue name="ip adress" propertyid="123_1">123.87.10.2</propertyValue>
    <propertyValue name="additional information" propertyid="123_3">bla bla bla</propertyValue>
  </ci>
</configurationItems>
- <compositions>
  <composition id="ab" cmdbid="ab" ownerid="a" usedid="b" />
  <composition id="ba" cmdbid="cd" ownerid="b" usedid="a" />
</compositions>
</data>
</cmdb>
- <cmdb id="yyy" parentid="xxx" cmdbid="yyy" name="cmdbyyy">
```

For each of these elements, you must define the name, potentially a description (comment), but most of all its CMDB identifier.

```
<ci id="a" cmdbid="a1" name="my server 1" typeid="a123" status="active" delete="yes">
```

## Configuration item type

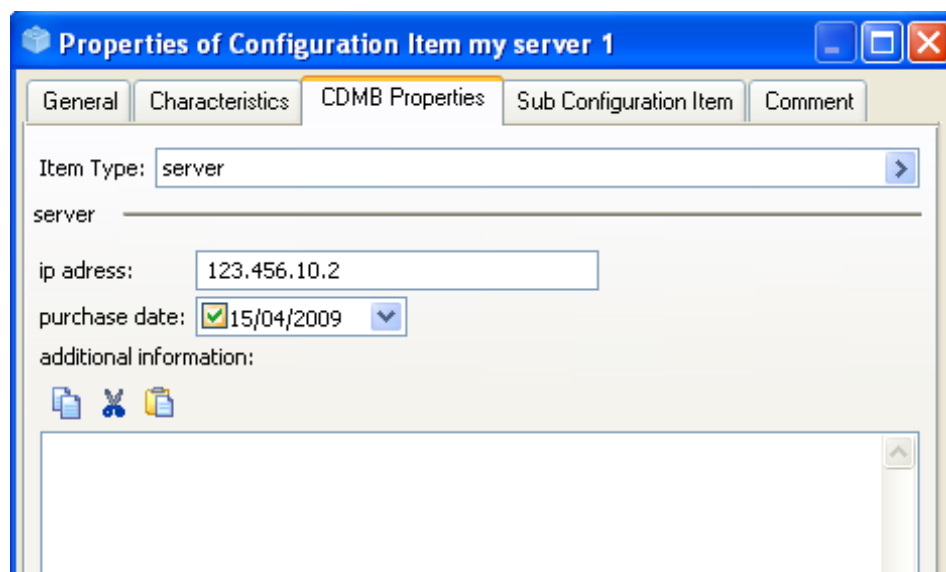
Configuration item types are associated to a CMDB.

If certain properties do not exist in MEGA, you can define them in the XML file so that MEGA can create them and link them to the appropriate items.

### Example of CMDB properties of a configuration item of "Server" type

```
- <types>
- <ciType id="a123" cmdbid="azerty" name="server">
  <description>This is the comment of this configuration item</description>
  - <properties>
    <property id="123_1" cmdbid="123_a" name="ip adress" type="string" length="15" />
    <property id="123_2" cmdbid="123_b" name="purchase date" type="date" format="yyyy/mm/dd" />
    - <property id="123_3" cmdbid="123_c" name="additional information" type="text">
      <description>This property is used to define additional information</description>
    </property>
  </properties>
</ciType>
</types>
```

Result in MEGA:



## Configuration item

List of configuration items contained in a CMDB is described in the <Data> tag.

Each configuration item refers to a type which has been defined previously in the same CMDB.

```
<ci id="a" cmdbid="a1" name="my server 1" typeid="a123" status="active" delete="yes">
```

On a configuration item, you can specify values of properties defined on the type it refers to.

## Configuration item composition

A configuration item can contain other configuration items. The link between an item and its sub-item is represented by a Configuration Item Composition.

```
</configurationItems>
```

```
<compositions>
```

```
  <composition id="ab" cmdbid="ab" ownerid="a" usedid="b" />
```

```
  <composition id="ba" cmdbid="cd" ownerid="b" usedid="a" />
```

```
</compositions>
```