

# CUDB LDAP Schema Management Graphical User Interface

---

## USER GUIDE

**Copyright**

© Ericsson AB, 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Document Purpose and Scope	1
1.2	Revision Information	1
1.3	Typographic Conventions	1
<b>2</b>	<b>Running the Application</b>	<b>3</b>
<b>3</b>	<b>Window Management</b>	<b>5</b>
3.1	Main Window	5
3.2	Project View	6
3.3	Editor View	6
3.3.1	Schema Editor	6
3.3.1.1	Overview Tab	7
3.3.1.2	Source Code Tab	7
3.3.2	Attribute Editor	8
3.3.2.1	Overview Tab	8
3.3.2.2	Source Code Tab	9
3.3.3	Object Class Editor	10
3.3.3.1	Overview Tab	10
3.3.3.2	Source Code Tab	11
3.4	Log View	12
<b>4</b>	<b>Managing Projects</b>	<b>15</b>
4.1	Creating a Project	15
4.2	Deleting a Project	16
<b>5</b>	<b>Managing Schemas</b>	<b>17</b>
5.1	Creating an Empty Schema	17
5.2	Importing an Existing Schema	18
5.3	Commenting Schemas	18
5.4	Renaming a Schema	20
5.5	Deleting a Schema	20
<b>6</b>	<b>Managing Attributes</b>	<b>23</b>
6.1	Creating an Attribute	23
6.2	Modifying an Attribute	25
6.3	Moving an Attribute	26
6.4	Deleting an Attribute	27



<b>7</b>	<b>Managing Object Classes</b>	<b>29</b>
7.1	Creating an Object Class	29
7.2	Modifying an Object Class	31
7.3	Moving an Object Class	32
7.4	Deleting an Object Class	33
	<b>Glossary</b>	<b>35</b>
	<b>Reference List</b>	<b>37</b>



# 1 Introduction

This document describes the Graphical User Interface (GUI) of the CUDB LDAP Schema Management tool.

## 1.1 Document Purpose and Scope

The CUDB LDAP Schema Management GUI is a graphical application that allows editing the Lightweight Directory Access Protocol (LDAP) schema files. The document describes how to work with projects and schemas, and how to edit schemas within a project.

The screenshots contained in this user guide may show different font sizes or font types, depending on the graphical libraries used to run the tool.

## 1.2 Revision Information

**Rev. A** This document is based on 1/1553-CNH 160 6161/9 and contains editorial changes only.

## 1.3 Typographic Conventions

Typographic conventions can be found in the following document:

- *Typographic Conventions*





## 2 Running the Application

The Schema Management GUI must be installed to an external device running Linux, not on the System Controllers (SC) of the CUDB nodes. Run the application with the following command:

```
<DIR_TOOLS>/schemaMgmt/schemaMgmt
```

In the above command, `<DIR_TOOLS>` is the Schema Management Tool installation directory

**Note:** The execution directory of the above command is also the workspace directory of the Schema Management GUI. To ensure workspace persistency, execute the command from a non-temporary directory.

The `schemaMgmt` command supports the following command options:

- `--loglevel`: Use the `-l` | `--loglevel` flag to change the log level of the application as shown below:

```
<DIR_TOOLS>/schemaMgmt/schemaMgmt -logLevel [ ALL |
DEBUG | INFO | WARN | ERROR ]
```

The default log level is `INFO`.

- `--help`: For more information on how to use the application command, execute it with the `-h` | `--help` flag as shown below:

```
<DIR_TOOLS>/schemaMgmt/schemaMgmt -h
```

The expected output must be similar to the below example:

```
USAGE
schemaMgmt : starts the application with log level INFO
schemaMgmt -l|--logLevel [ALL, DEBUG, INFO, WARN, ERROR] : starts the application with the log level specified
schemaMgmt -h|--help : shows this help
```





## 3 Window Management

This section describes the main components of the Schema Management GUI.

### 3.1 Main Window

The Schema Management interface consists of three main components as shown below in Figure 1. Click on a colored area for a description of the respective view.

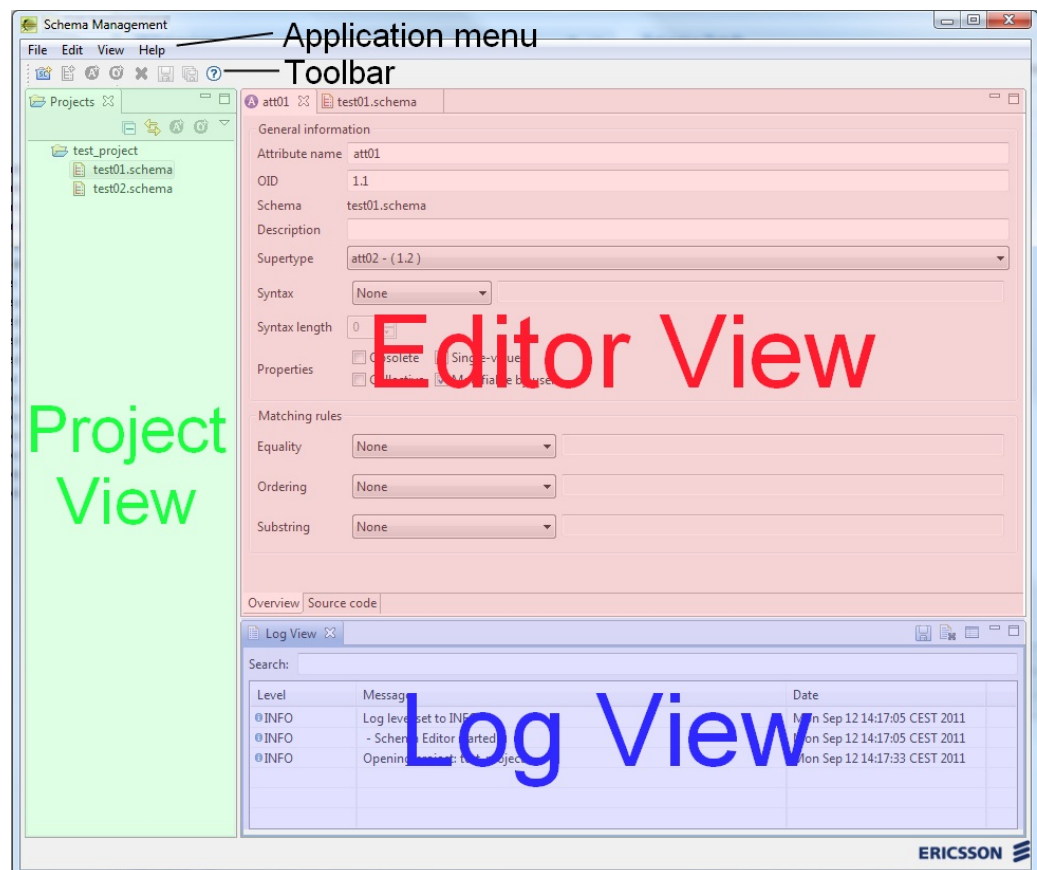


Figure 1 Schema Management GUI views.

The three major window areas are as follows:

- **Project View:** Shows the different projects and schemas on a tree view.
- **Editor View:** Shows the schema, attribute and object class editors, organized in tabs.
- **Log View:** Shows the application log.



These views can be minimized or maximized with the two icons in the upper right corner of each view.

## 3.2 Project View

The Project View is the main entry point of the graphical interface. It provides contextual menus for creating and deleting schemas within a project, and creating and deleting object classes and attributes within a schema.

The Project View displays the information about the projects and schemas in an hierarchical way. Projects are the main containers for schemas. Therefore, schemas can only exist in the context of a project. All data within a schema (that is, objects and attributes) are visible to all schemas in the project. This means that attributes defined in one schema can be used in other schemas as well.

**Note:** Although the Schema Management GUI allows the existence of cross-references between the schemas, these are not allowed in OpenLDAP, where the schema load order defines the allowed dependencies.

For more information about the projects, see Section 4 on page 15.

## 3.3 Editor View

Editors provide the actual graphical interface for modifying specific data. As a general rule, data is not actually modified until saved.

The view contains three different editors:

- The **Schema Editor** (see Section 3.3.1 on page 6).
- The **Attribute Editor** (see Section 3.3.2 on page 8)
- The **Object Class Editor** (see Section 3.3.3 on page 10).

All editors are shown in a new tab in the Editor View when they are opened.





### 3.3.1 Schema Editor

The Schema Editor provides information about the object classes and attributes in a schema, and also shows the comments contained in the schema file. These comments have certain limitations: see Section 5.3 on page 18 for more information.

The Schema Editor consists of the Overview Tab (see Section 3.3.1.1 on page 6) and the Source Code Tab (see Section 3.3.1.2 on page 7).

### 3.3.1.1 Overview Tab

The Overview Tab is shown in Figure 2. This tab provides three panes that are used to show the schema comments, and the names and OIDs of the object classes and attributes in the schema. The object classes and attributes listed in the tab can be edited by double-clicking on them, which opens the corresponding editor.

New object classes and attributes can be added by clicking on the  and  buttons respectively, while existing object classes and attributes can be deleted by clicking on the  or  buttons, respectively. Attribute and object class lists can be sorted by name or OID by clicking on the column title of each table.

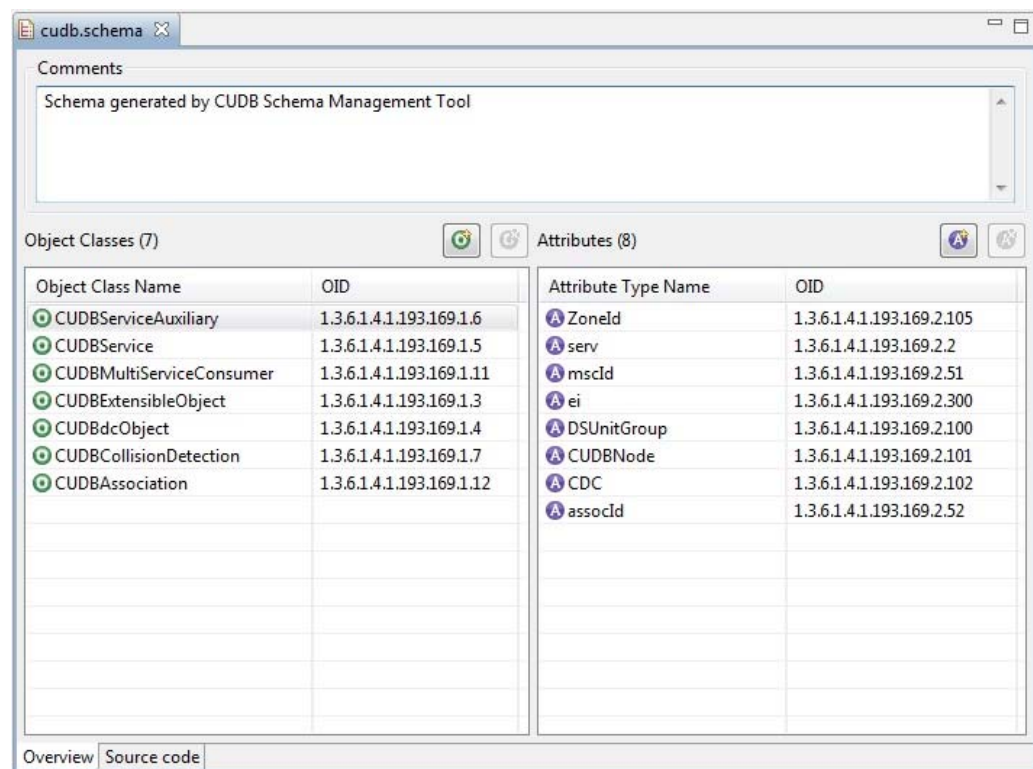
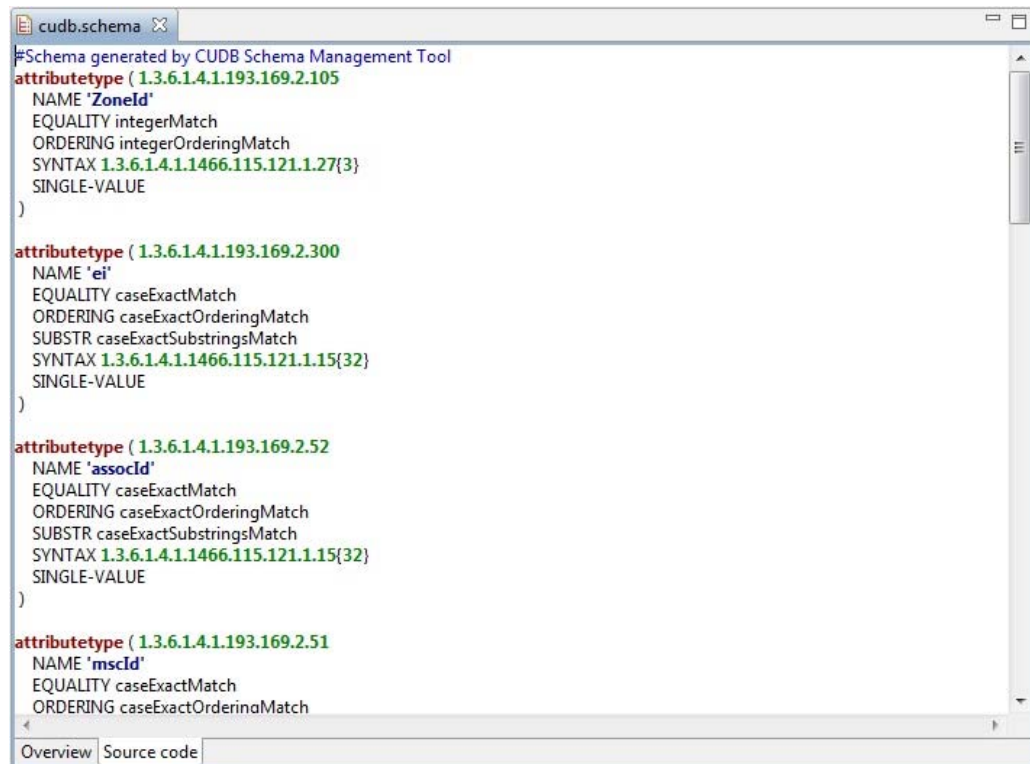


Figure 2 Schema Editor - Overview Tab

### 3.3.1.2 Source Code Tab

The Source Code Tab is shown in Figure 3. This tab shows the schema information as it is stored in the schema file, using highlighted syntax. The code shown in the Source Code Tab is not editable.



```

#Schema generated by CUDB Schema Management Tool
attributetype ( 1.3.6.1.4.1.193.169.2.105
  NAME 'ZoneId'
  EQUALITY integerMatch
  ORDERING integerOrderingMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27{3}
  SINGLE-VALUE
)

attributetype ( 1.3.6.1.4.1.193.169.2.300
  NAME 'ei'
  EQUALITY caseExactMatch
  ORDERING caseExactOrderingMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32}
  SINGLE-VALUE
)

attributetype ( 1.3.6.1.4.1.193.169.2.52
  NAME 'assocId'
  EQUALITY caseExactMatch
  ORDERING caseExactOrderingMatch
  SUBSTR caseExactSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32}
  SINGLE-VALUE
)

attributetype ( 1.3.6.1.4.1.193.169.2.51
  NAME 'mscId'
  EQUALITY caseExactMatch
  ORDERING caseExactOrderingMatch

```

Figure 3 Schema Editor - Source Code Tab

## 3.3.2 Attribute Editor

The Attribute Editor provides information about a particular attribute of a schema, and allows the modification of the attribute properties.

The Attribute Editor consists of the Overview Tab (see Section 3.3.2.1 on page 8) and the Source Code Tab (see Section 3.3.2.2 on page 9).

### 3.3.2.1 Overview Tab

The Overview Tab is shown in Figure 4.



CUDBNode

General information

Attribute name: CUDBNode

OID: 1.3.6.1.4.1.193.169.2.101

Schema: cudb.schema

Description:

Supertype: None

Syntax: Other 1.3.6.1.4.1.1466.115.121.1.26

Syntax length: 0

Properties: ☐ Obsolete ☒ Single-valued ☐ Collective ☒ Modifiable by user

Matching rules

Equality: caseIgnoreIA5Match 1.3.6.1.4.1.1466.109.114.2

Ordering: None

Substring: None

Overview Source code

Figure 4 Attribute Editor - Overview Tab

The Overview Tab shows detailed information about the selected attribute, and consists of the following two sections:

- The “General information” section contains the name, OID, schema, description, supertype, syntax information, and properties of the attribute.
- The “Matching rules” section shows the matching rules for equality, ordering and substring.

### 3.3.2.2

#### Source Code Tab

The Source Code Tab is shown in Figure 5.

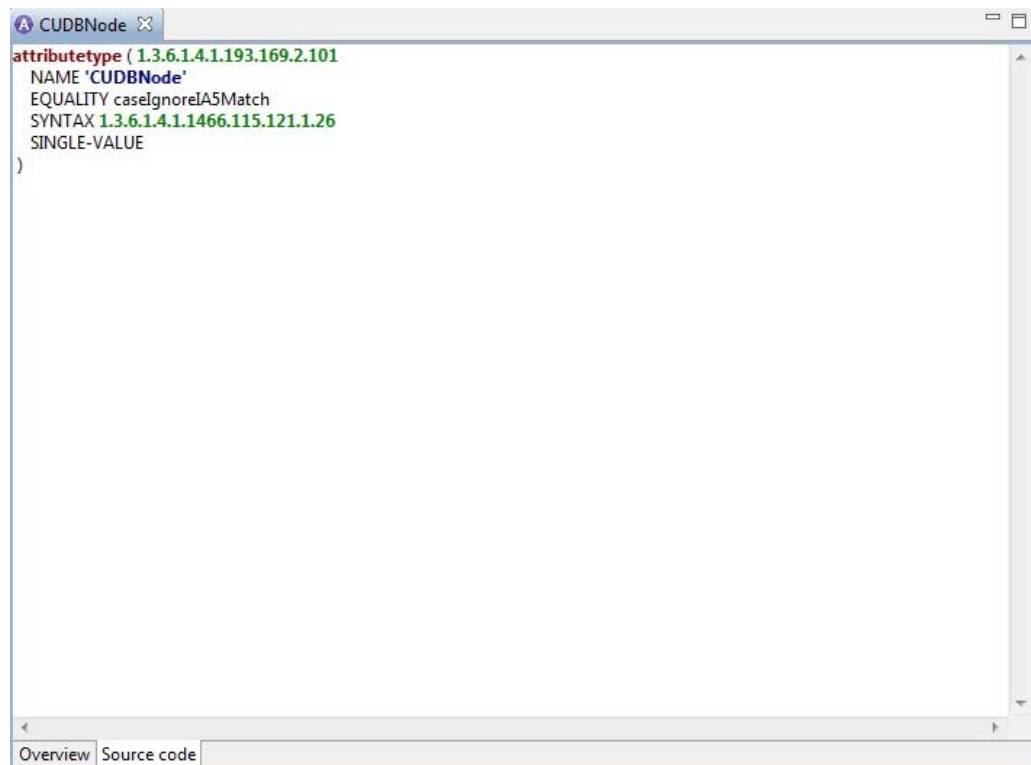


Figure 5 Attribute Editor - Source Code Tab

The Source Code Tab shows the source code of the attribute as it is stored in the schema file. The code is not editable in this window. When the attribute is edited in the Overview Tab, the new information appears automatically in the Source Code Tab, even if the attribute has not been saved yet.

### 3.3.3 Object Class Editor

The Object Class Editor provides information about a particular object class in a schema, and allows modifying the object class properties.

The Object Class Editor consists of the Overview Tab (see Section 3.3.3.1 on page 10) and the Source Code Tab (see Section 3.3.3.2 on page 11).

#### 3.3.3.1 Overview Tab

The Overview Tab is shown on Figure 6.



CUDBAssociation

General information

Name: CUDBAssociation

OID: 1.3.6.1.4.1.193.169.1.12

Schema: cudb.schema

Description:

Superior classes: top [Add] [Remove]

Class type: Structural

Mandatory attributes: assocId [Add] [Remove]

Optional attributes: ZoneId, DSUnitGroup [Add] [Remove]

Overview | Source code

**Figure 6** Object Class Editor - Overview Tab

The Overview Tab shows detailed information about the selected object class, and consists of the following three sections:

- The “General Information” section includes the name, OID, schema, description, superior classes, and class type of the selected object class.
- The “Mandatory attributes” section lists the mandatory attributes of the object class.
- The “Optional attributes” section lists the optional attributes of the object class.

### 3.3.3.2 Source Code Tab

The Source Code Tab is shown on Figure 7.



Figure 7 Object Class Editor - Source Code Tab

The Source Code Tab shows the source code of the object class as it appears in the schema file. The code is not editable in this window. When the object class is edited in the Overview Tab, the new information appears automatically in the Source Code Tab, even if the object class has not been saved yet.

## 3.4 Log View

The Log View (shown in Figure 8 below) provides information about the progress of the operations performed in the Schema Management GUI. The Log View shows messages only with severity equal or above the defined log level (see Section 2 on page 3 for more information).

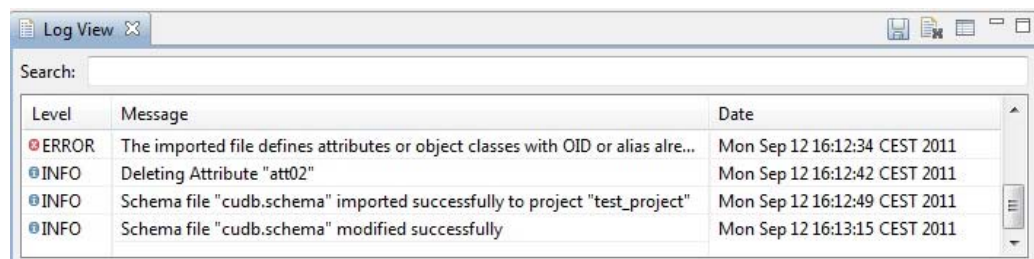



Figure 8 Log View



The Log View displays the following information about every event:

- “Level”: The related severity level of the event (DEBUG, INFO, WARNING, ERROR or FATAL).
- “Message”: The description of the event.
- “Date”: The date and time of logging the event.

The icons in the upper right side of the Log View window allow performing the following actions (from left to right):

- **Save** (

Use the **Search** textbox at the top of the log message list to search for specific messages in the log list. When something is typed in the textbox, the messages containing the typed text are listed.

Log messages can also be hidden: to do so, right-click on the header of the columns, and uncheck the log levels, messages, or dates to hide.





## 4 Managing Projects

Projects are the main elements of the Schema Management GUI. The Project View shows all projects contained in the workspace, but only one project can be opened at a time.

### 4.1 Creating a Project

To create a new project, right-click on the Project View, and select the `New > Project` command from the shortcut menu. Alternatively, select the same command from the application menu. After that, a wizard window is shown as seen in Figure 9 below.

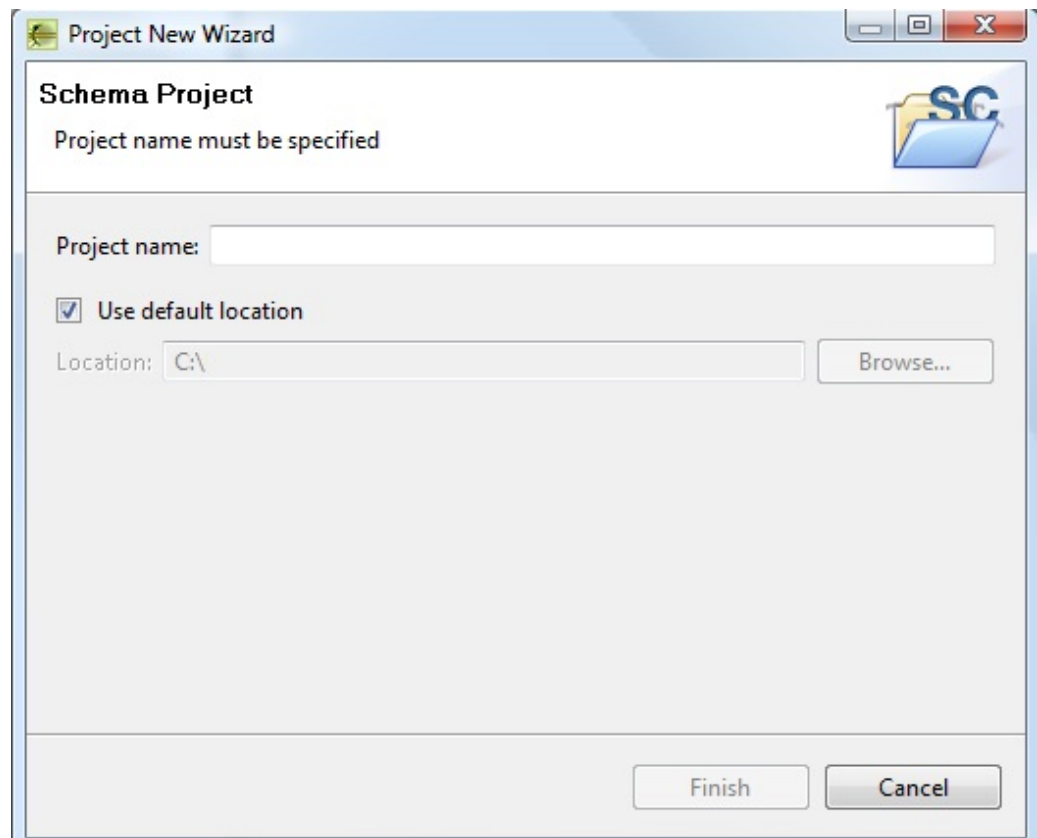


Figure 9 New Project Wizard

The wizard allows to specify the project name and location. By default, the new project is stored in a folder named `workspace`, located in the launching directory of the Schema Management GUI. After clicking the `Finish` button, a new folder is created in the specified location, and the project is shown in the Project View as an open project. Only one project can be opened at a time, so other projects and the editors open for them will be closed.

## 4.2 Deleting a Project

Perform one of the following steps to delete an existing project:

- ☐ Right-click on it in the Project View, and select the `Delete` command from the shortcut menu.
- ☐ Select the project to delete in the Project View, then select the `Edit > Delete` command from the application menu.
- ☐ Select the project to delete in the Project View, then click on the `Delete` button of the toolbar.

If any of the above steps is performed, a pop-up window appears as shown in Figure 10, asking to confirm the operation.

**Note:** Once the project is deleted, it cannot be recovered. If a project is deleted, the project folder and all related schema files are also deleted from the file system.

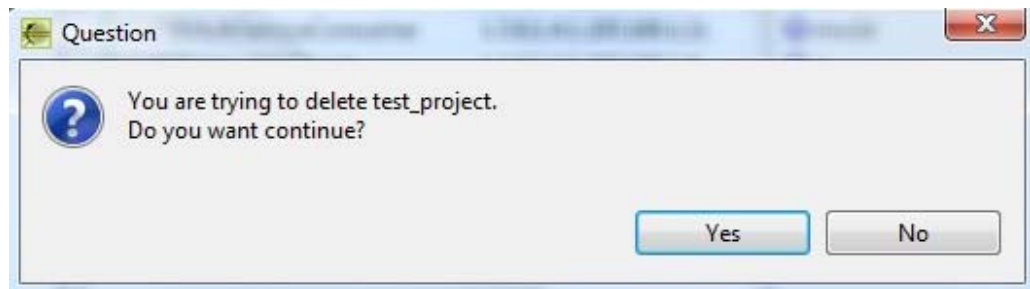


Figure 10 Confirmation of Deleting a Project



## 5 Managing Schemas

A schema is an LDAP schema file defining a list of attributes and object classes. The Schema Editor allows to create new schemas, or to import existing schema files.

### 5.1 Creating an Empty Schema

To create a new schema, open the project where the schema will be created.

Perform one of the below procedures to create a new empty schema:

- ☐ Right-click on the Project View, and select the **New > Schema** option from the shortcut menu.
- ☐ Click on the **File > New > Schema** option from the application menu.
- ☐ Click on the **New Schema** button on the toolbar.

Once one of the above actions is performed, the New Schema Wizard is launched, as shown on Figure 11.

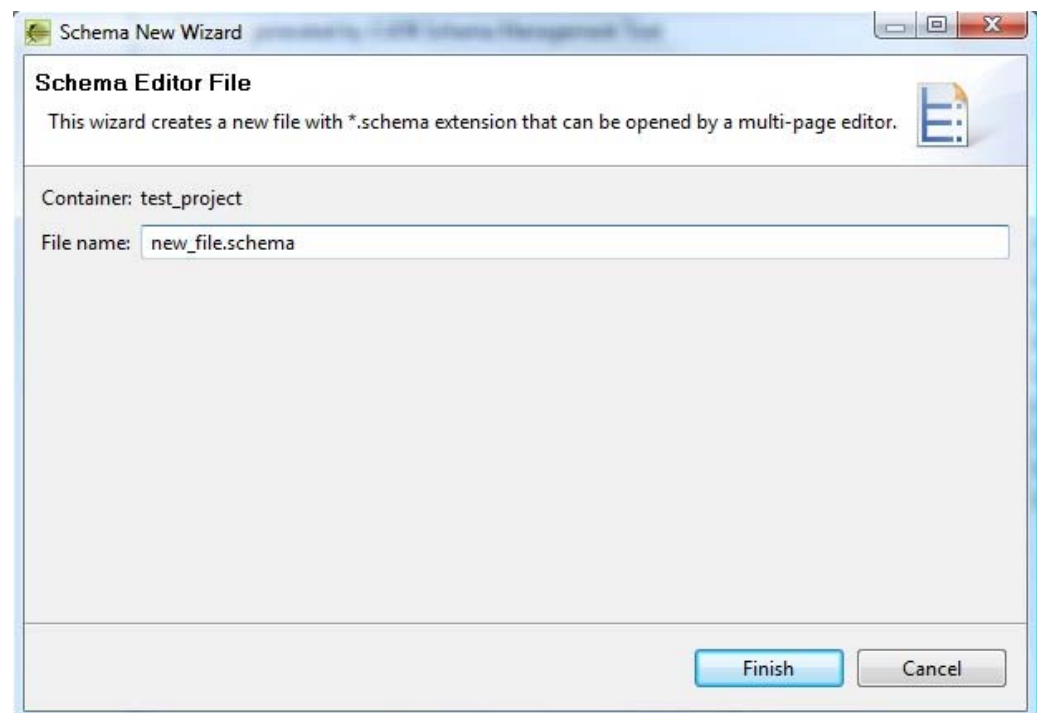


Figure 11 New Schema Wizard

## 5.2 Importing an Existing Schema

Perform one of the below procedures to import an existing schema:

- ☐ Right-click on the Project View, and select the `Import Schema` option from the shortcut menu.
- ☐ Select the `File > Import Schema` option from the application menu.

Once one of the above actions is performed, the Schema Import Wizard is launched, as shown on Figure 12.

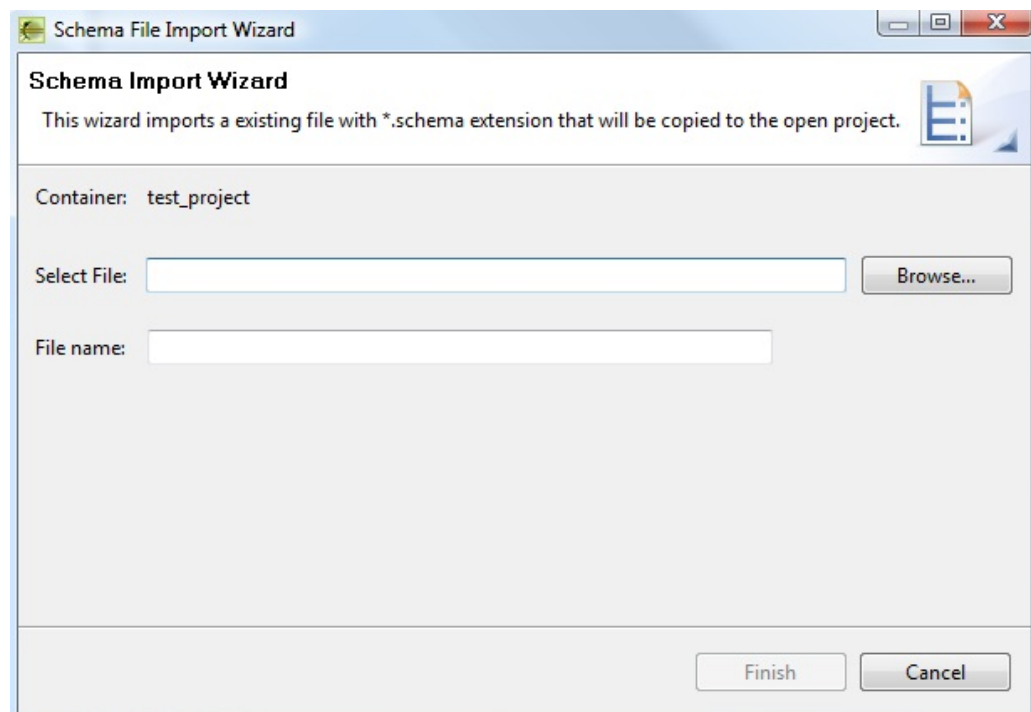


Figure 12 Schema Import Wizard

The wizard allows to select a schema file from the file system and to change the name of the selected schema in the project. The wizard also shows the name of the project that will contain the new schema.

## 5.3 Commenting Schemas

In a schema file, comments are lines starting with a number sign (#). Comment lines can be modified in the Overview Tab of the Schema Editor. The Schema Management Tool can only manage comments that are located on the top of schema files, before any non-comment line is found (either blank lines, or lines containing actual schema definition information), as shown in Figure 13.



```

1  # example.schema
2  # This is an example.
3  # This block of comments will be loaded in the Schema Editor
4
5  # This comment will be lost
6  attributetype ( 1.2.3.4.5.6 NAME 'attribute01'
7      |          SYNTAX 1.2.3.4.5.6{32}
8      |          SINGLE-VALUE)
9
10 # These comments will be lost too.
11 # Sample comment.
12 # Sample comment.
13
14 attributetype ( 1.2.3.4.5.6.7 NAME 'attribute03'
15      |          SYNTAX 1.2.3.4.5.6.7{32}
16      |          SINGLE-VALUE)

```

Figure 13 Comments in the Schema File

**Note:** If a schema file is imported to a project, and it is modified, all comments contained in the file will be lost, except the comments located at the beginning of the file before the actual schema definition content.

The schema shown in Figure 13 is loaded into the Schema Editor as shown in Figure 14:

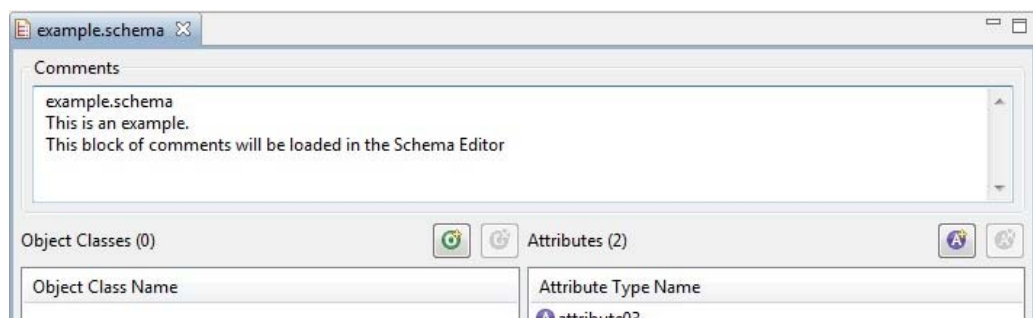


Figure 14 Schema File Loaded into Schema Editor

Due to the comment removal logic described at the beginning of this section, the comments at line 5, 10, 11 and 12 in Figure 13 are lost after the file is loaded into the Schema Editor. This is because these comments are preceded by empty non-commented lines, and actual schema definition information.

When the comments of a schema are modified, an asterisk appears at the beginning of the schema name as shown in Figure 15. The asterisk indicates that the schema contains unsaved changes.

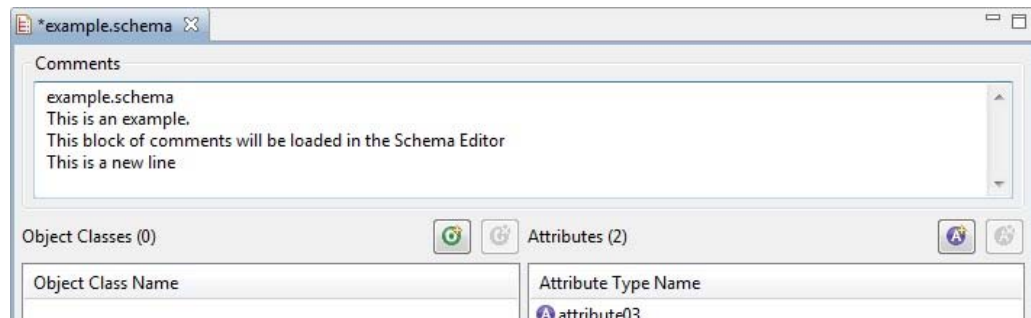


Figure 15 Schema Containing Unsaved Changes

## 5.4 Renaming a Schema

To rename a schema, perform one of the below procedures:

- ☐ Right-click on the specific schema in the Project View, then select the **Rename** option from the shortcut menu.
- ☐ Select the specific schema in the Project View, and select the **Edit > Rename** option from the application menu.

Once one of the above actions is performed, the pop-up window shown in Figure 16 appears, asking for the new schema name.

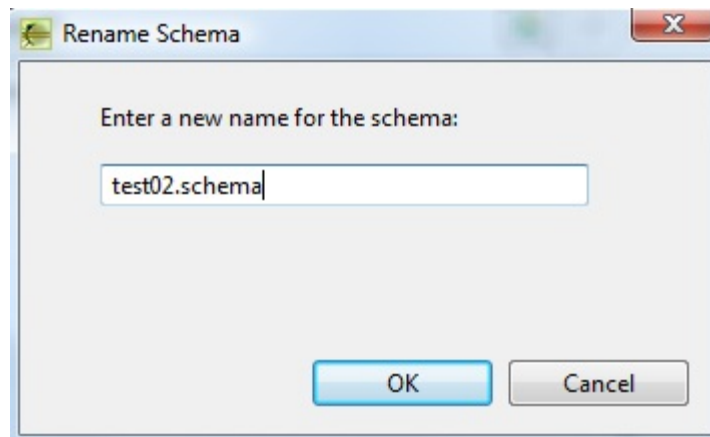


Figure 16 Rename Schema Window

## 5.5 Deleting a Schema

Perform one of the below procedures to delete an existing schema:

- ☐ Right-clicking on the selected schema in the Project View, then select the **Delete** option from the shortcut menu.
- ☐ Select the schema in the Project View, then select the **Edit > Delete** option from the application menu.



- ☐ Select the schema in the Project View, then click on the **Delete** button of the toolbar.

Once one of the above actions is performed, the schema file is deleted from the system. In case the schema to delete contains any attributes and object classes used in other schemas of the project, inconsistencies can occur: the GUI indicates this with a pop-up window shown in Figure 17. The window lists all attributes and object classes that are used in other schemas of the same project, and that depend on the data in the schema to be deleted.

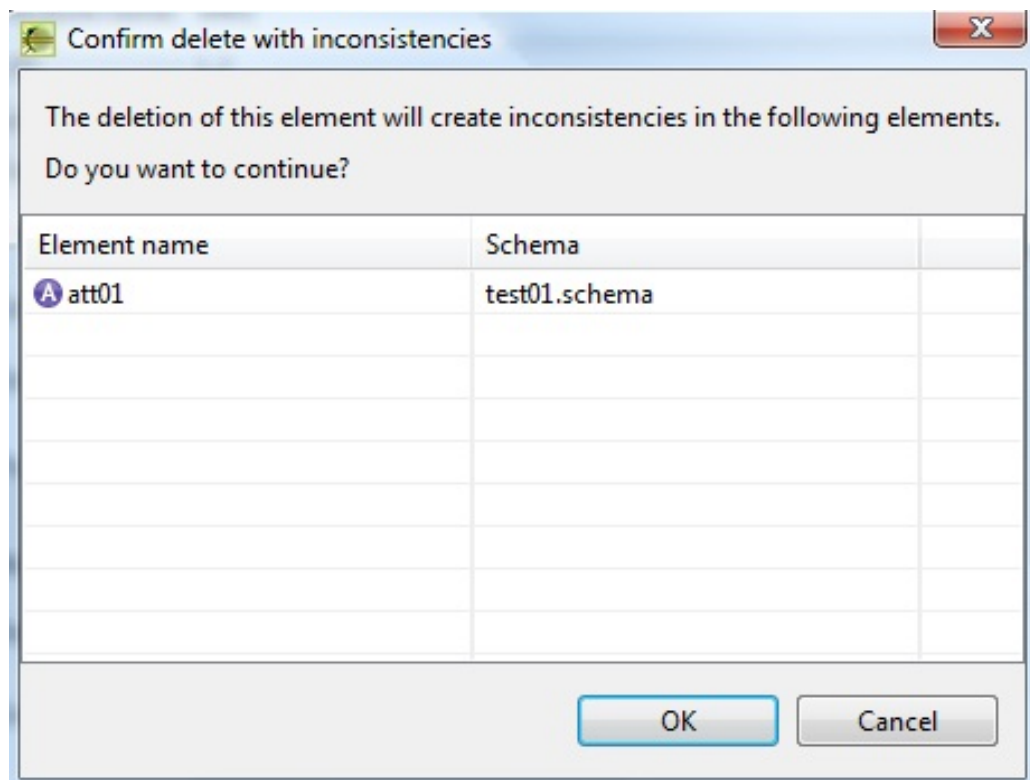


Figure 17 Potential Inconsistencies of Deleting a Schema






## 6 Managing Attributes

This section describes how to manage attributes in a schema.

### 6.1 Creating an Attribute

Perform one of the below procedures to create a new attribute in a schema:

- ☐ Right-click on a schema in the Project View, then select the **New > Attribute** option from the shortcut menu.
- ☐ Select a schema in the Project View by clicking on it, then click on the  button on the toolbar (as shown in Figure 18).

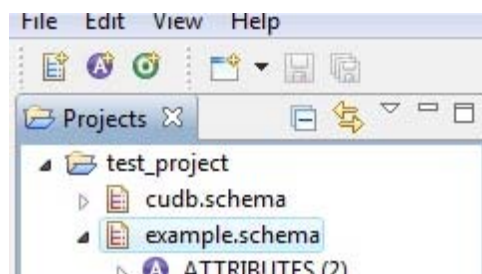



Figure 18 A+ Button in the Toolbar

- ☐ Select a schema in the Project View by clicking on it, then select the **File > New > Attribute** command from the application menu.
- ☐ Click on the  button located on the top of the Attribute list in the Schema Editor (as shown in Figure 19).

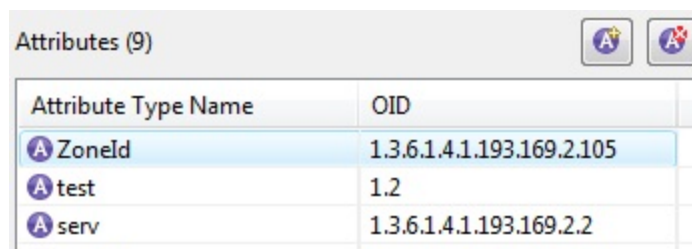
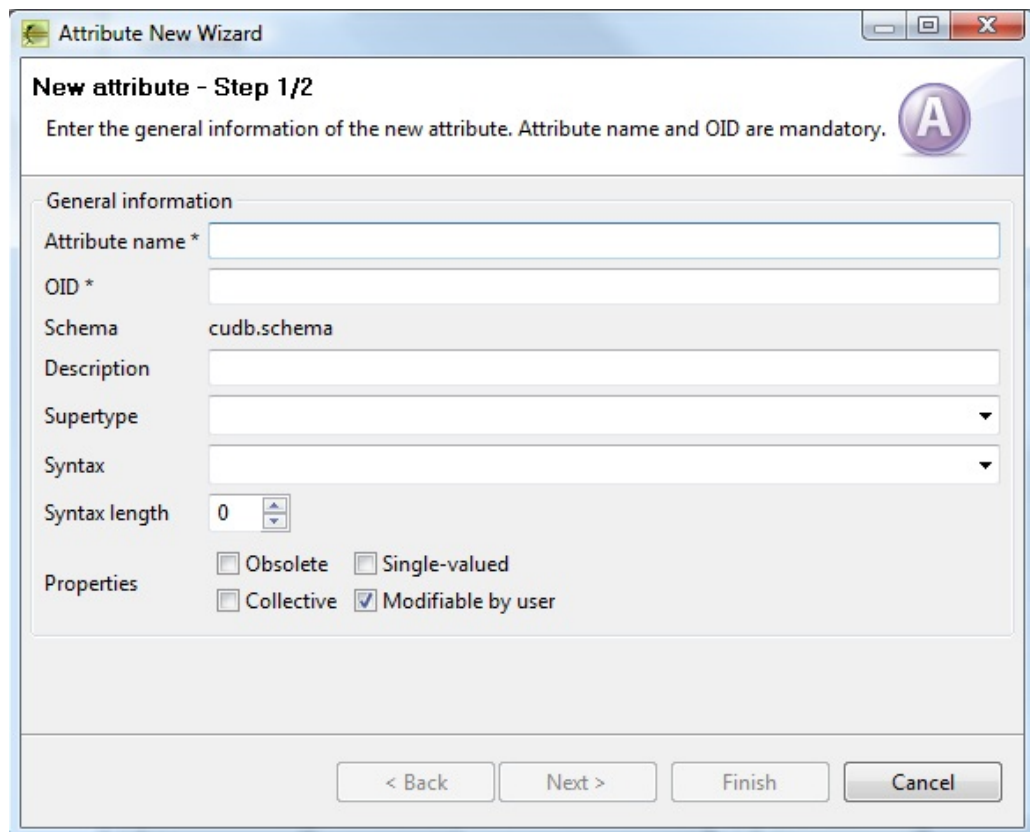


Figure 19 A+ button in an Schema Editor

Once one of the above actions is performed, the New Attribute Wizard opens, where the new attribute can be created in two steps. In Step 1 (shown in Figure 20), the basic information of the attribute must be added. This includes the following:

- **Attribute name:** Name of the attribute. Mandatory field. If the selected name is incorrect or used by another element of the project, the window will display an error.
- **OID:** OID of the attribute. Mandatory field. If the selected name is incorrect or used by another element of the project, the window will display an error.
- **Schema:** The name of the schema that will contain the new attribute. The schema name is not editable in this window.
- **Description:** Describes the schema.
- **Supertype:** An attribute within the project which is set as the supertype of the current attribute.
- **Syntax:** The syntax of the attribute.
- **Syntax length:** The allowed length of the syntax.
- **Properties:** The properties of the attribute. The available attributes are Obsolete, Collective, Single-valued, or Modifiable by user.



**Attribute New Wizard**

**New attribute - Step 1/2**

Enter the general information of the new attribute. Attribute name and OID are mandatory.

**General information**

Attribute name \*

OID \*

Schema cudb.schema

Description

Supertype

Syntax

Syntax length 0

**Properties**

☐ Obsolete ☐ Single-valued

☐ Collective ☒ Modifiable by user

< Back Next > Finish Cancel

Figure 20 Create New Attribute Wizard, Step 1

Step 2 of the wizard (shown in Figure 21) is used to introduce the matching rules of the new attribute.



Figure 21 Create New Attribute Wizard, Step 2

Once all the information of the new attribute is filled in, click on the **Finish** button. After that an **Attribute Editor** window opens (see Section 3.3.2 on page 8), showing the properties of the new attribute. The new attribute is also added to the **Project View**. When a new attribute is created, the schema file is automatically updated: therefore, it is not necessary to save the schema after creating an attribute.

## 6.2 Modifying an Attribute

The properties of an attribute can be modified through the corresponding sections of the **Overview Tab** in the **Attribute Editor** (see Figure 4). The properties available for modification are as follows:

- **Attribute name:** The name of the attribute. The name can be modified or a new one can be added (separated from the other by comma), as OpenLDAP allows attributes to have more than one name. However, each name must be unique within the project: if a selected name is already in use, the window will display an error.
- **OID:** The OID of the attribute, in the format of dot-separated numbers. The OID of the attribute must be unique within the project.

- **Schema:** The name of the schema. The schema name cannot be modified in this window, but it is possible to move the selected attribute to another schema within the project. For more information, see Section 6.3 on page 26.
- **Description:** Describes the schema. The existing text can be modified.
- **Supertype:** The drop-down list can be used to select an attribute within the project to be set as the supertype of the attribute.
- **Syntax:** The syntax of the attribute. For value modification use the drop-down list or write a valid OID directly into the textbox. In case the OID is not valid, the window displays an error. If a new OID is introduced, it must be unique within the project.
- **Syntax lenght:** The allowed length of the syntax can be modified by either typing a new number or by increasing/decreasing the current value with the arrows on the right.
- **Properties:** Used to modify attribute properties (Obsolete, Collective, Single-valued, or Modifiable by user). Change the properties by checking or unchecking the available options.

The window can also be used to modify the Matching Rules (Equality / Ordering / Substring) by selecting the desired rule from the drop down list, or by directly writing a valid OID to the textbox on the right. If the typed OID is not valid, the window will display an error. If a new OID is introduced, it must be unique within the project.

## 6.3 Moving an Attribute

Use the **Move** option to move an attribute from its current schema to another schema within the project. When clicking on the option, a pop-up window appears (shown in Figure 22), allowing to select the new schema where the attribute will be moved to.

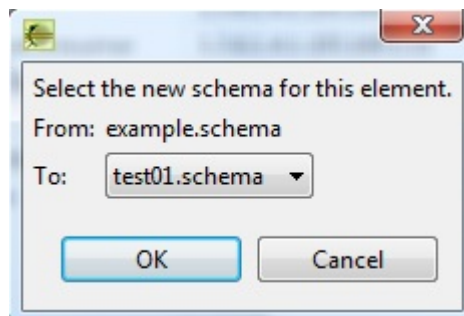



Figure 22 Move Attribute Dialog



## 6.4 Deleting an Attribute

Perform one of the below procedures to delete an attribute in a schema:

- ☐ Right-click on the selected attribute in the Project View, then select the **Delete** option from the shortcut menu.
- ☐ Select the affected attribute in the Project View by clicking on it, then select the **Edit > Delete** command from the application menu.
- ☐ Select the affected attribute in the Project View by clicking on it, then click on the **Delete** button of the toolbar.
- ☐ Select the desired attribute in the Attribute List of the Schema Editor, then click on the  button.

If the attribute to delete is used in other schemas of the project, the project can end up in an inconsistent state. To avoid this situation, a warning pop-up window appears listing the attributes and object classes used in others schemas of the same project.

Once one of the above actions is performed, the attribute is deleted. In case the attribute to delete is used in other schemas of the project, inconsistencies can occur: the GUI indicates this with a pop-up window shown in Figure 23. The window lists all attributes and object classes that are used in other schemas of the same project, and that depend on the attribute to be deleted.

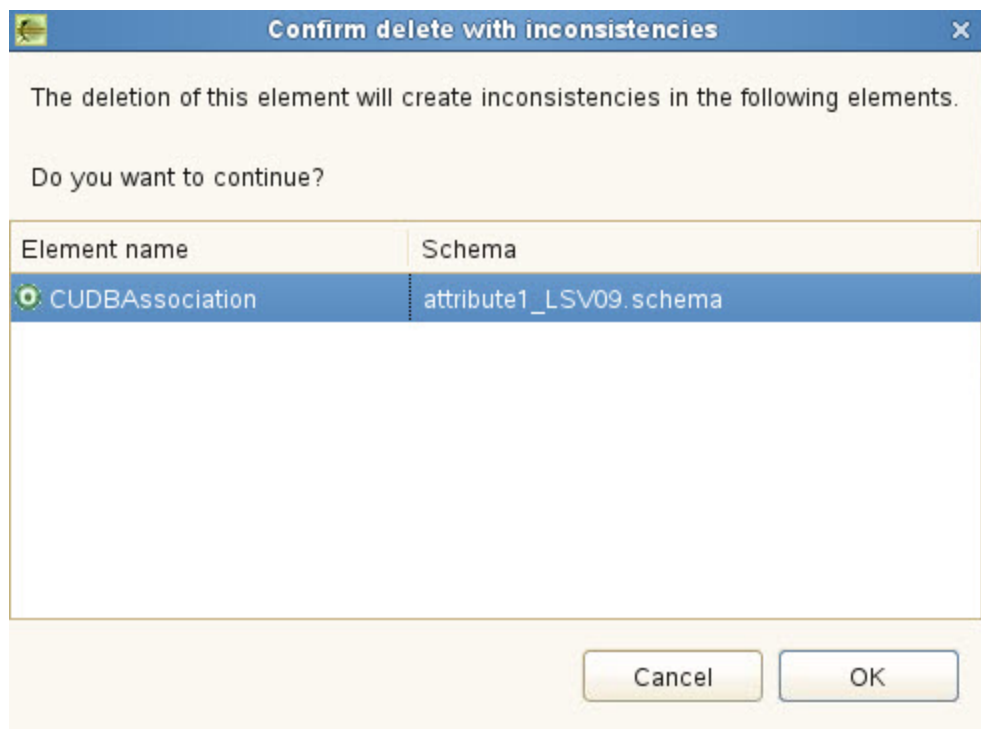


Figure 23 Potential Inconsistencies of Deleting an Attribute






## 7 Managing Object Classes

This section describes how to manage object classes with the Schema Management GUI.

### 7.1 Creating an Object Class

Perform one of the below procedures to create a new object class:

- ☐ Right-click on a schema in the Project View, then select the **New > Object Class** command from the shortcut menu.
- ☐ Select a schema in the Project View by clicking on it, then click on the  button on the toolbar (see Figure 24).

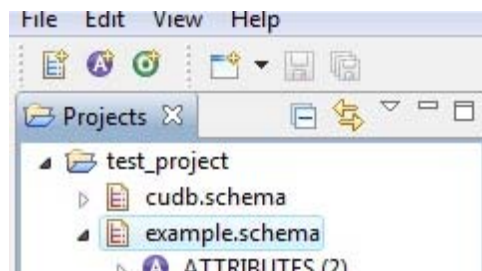



Figure 24 O+ Button in the Toolbar

- ☐ Select a schema in the Project View by clicking on it, then select the **File > New > Object Class** command from the application menu.
- ☐ Click on the  button located on top of the object class list in the Schema Editor (see Figure 25).

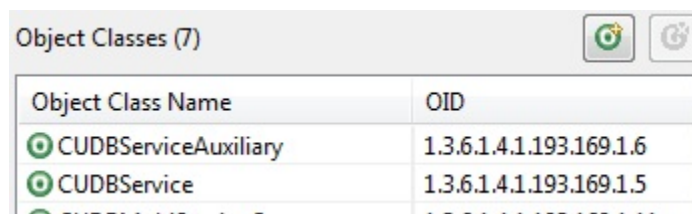


Figure 25 O+ Button in the Schema Editor

Once one of the above actions is performed, the New Object Class Wizard opens, where the new object class can be created in two steps. In Step 1 (shown in Figure 26) the basic information of the object class must be added. This includes the following:

- **Object class name:** The name of the object class. If the selected name is incorrect or already in use by another element of the project, an error will appear.
- **OID:** The OID of the object class. If the selected OID is incorrect or already in use by another element of the project, an error will appear.
- **Schema:** The schema that includes the object class. This field is not editable, it just shows the name of the corresponding schema.
- **Description:** The textual description of the object class.
- **Superior classes:** The object class or list of object classes which are superior to the current object class. Use the **Add** button to add new superior object classes to the list, or remove existing superior object classes with the **Remove** button.
- **Class type:** The type of the object class. Use the drop-down list to select the desired class type of the object class.

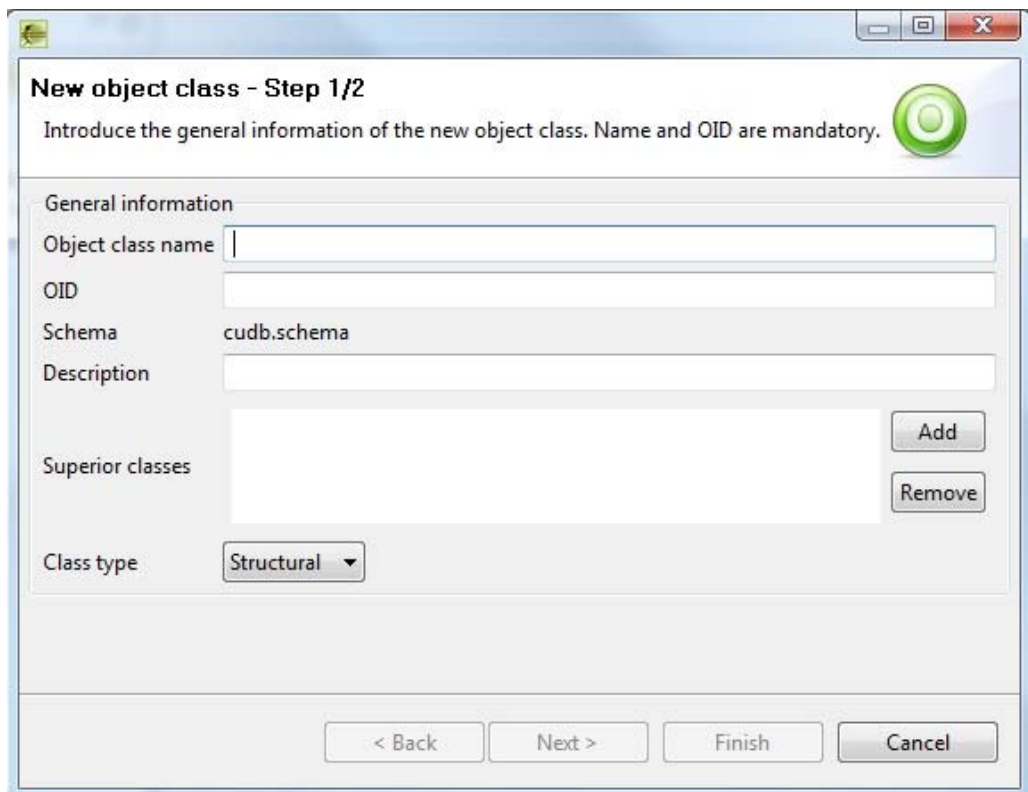


Figure 26 New Object Class Wizard, Step 1

In Step 2 of the wizard, mandatory and optional attributes can be defined. Use the **Add** buttons to open the list of possible attributes, and add them to the list of mandatory and optional attributes.

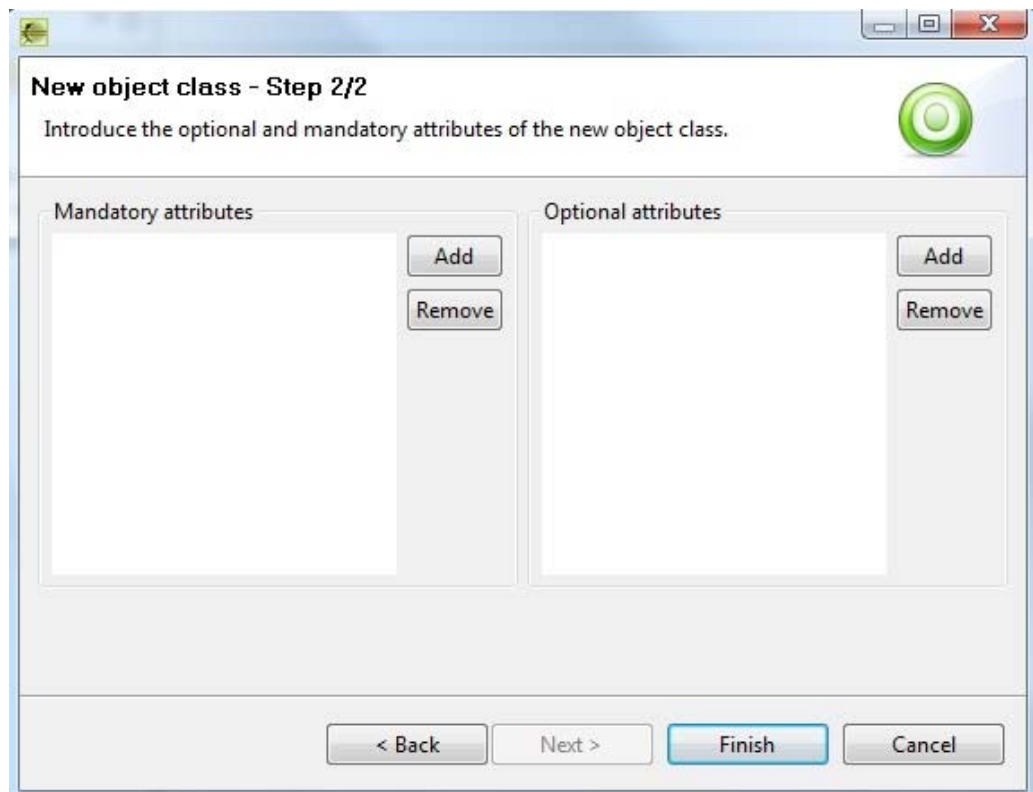


Figure 27 New Object Class Wizard, Step 2

Once all the information of the new object class is added, click on the **Finish** button. This opens the Object Class Editor showing the new object class (see Section 3.3.3 on page 10 for more information). The new object class is also included in the Project View. When a new object class is created, the schema file is automatically updated, so it is not necessary to save the schema after creating an object class.

## 7.2 Modifying an Object Class

The properties of an object class can be modified through the corresponding sections of the Overview Tab in the Object Class Editor (see Figure 6). The properties available for modification are as follows:

- **Name:** The name of the object class. The name can be modified or a new one can be added (separated with comma), as OpenLDAP allows object classes to have more than one name. Each name must be unique within the project: if an existing name is defined, an error appears.
- **OID:** The OID of the object class. The OID must be unique and must consist of dot-separated numbers. Use the textbox to modify the OID.
- **Schema:** The name of the schema to which the object class belongs. The schema name cannot be modified in this window.



- **Description:** The description of the object class. Use the textbox to change the textual description of the object class.
- **Superior classes:** The object class or list of object classes which are superior to the current object class. Use the **Add** button to add new superior object classes to the list, or remove existing superior object classes with the **Remove** button.
- **Class type:** The type of the object class. Use the drop-down list to select the desired class type of the object class.

The window can also be used to configure the mandatory and optional attributes of the object class in separate textboxes as follows:

- **Mandatory attributes:** This part of the window lists the mandatory attributes configured for the object class. To add new mandatory attributes, click on the **Add** button to the right of the textbox. This brings up the list of attributes within the project. To add any of them to the object class, double-click on them. A confirmation message is then shown on the top of the list each time a new mandatory attribute is successfully added. OpenLDAP allows to set the same attribute one or several times as mandatory attribute.

To delete a mandatory attribute, select it on the **Mandatory attributes** list, then click on the **Remove** button.

- **Optional attributes:** This part of the window lists the optional attributes configured for the object class. To add new optional attributes, click on the **Add** button to the right of the textbox. This brings up the list of attributes within the project. To add any of them to the object class, double-click on them. A confirmation message is then shown on the top of the list each time a new optional attribute is successfully added. OpenLDAP allows to set the same attribute one or several times as optional attribute.

To delete an optional attribute, select it on the **Optional attributes** list, then click on the **Remove** button.

## 7.3 Moving an Object Class

An object class can be moved from its current schema to another schema within the project with the **Move** option. When clicking on the **Move** button, a pop-up window appears (shown in Figure 28), allowing to select the new schema where the object class will be moved to.

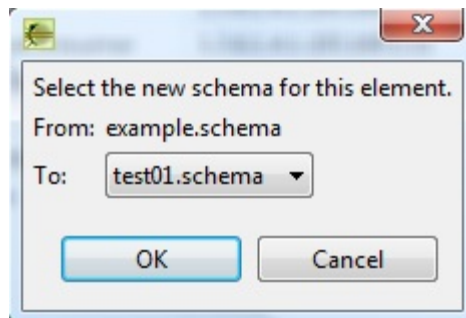



Figure 28 Move Object Class Window

## 7.4 Deleting an Object Class

Perform one of the below procedures to delete an object class from a schema:

- ☐ Right-click on the specific object class in the Project View, then select the `Delete` command from the shortcut menu.
- ☐ Click on the specific object class in the Project View, then select the `Edit > Delete` command from the application menu.
- ☐ Click on the specific object class in the Project View, then click on the `Delete` button of the toolbar.
- ☐ Select the specific object class in the object class list of the Schema Editor, then click on the  button.

Once one of the above actions is performed, the object class is deleted. In case the attribute to delete is used in other schemas of the project, inconsistencies can occur. To avoid this situation a warning pop-up window appears listing all the object classes used in other schemas of the same project.





## Glossary

For the terms, definitions, acronyms and abbreviations used in this document, refer to *CUDB Glossary of Terms and Acronyms*, Reference [1].





## Reference List

### **CUDB Documents**

- [1] *CUDB Glossary of Terms and Acronyms*