openBIS admin tutorial

28.10.2021

Table of Contents

[1 Overview 2](#_Toc75292580)

[2 Object types registration 3](#_Toc75292581)

[2.1 Registration of a new Object type 3](#_Toc75292582)

[2.1.1 Edit an Object type 3](#_Toc75292583)

[2.1.2 Create a vocabulary entry 4](#_Toc75292584)

[2.2 Registration of a second Object type 4](#_Toc75292585)

[3 Create Collection folders in ELN User Interface 5](#_Toc75292586)

[4 ELN Settings for Object types customization 5](#_Toc75292587)

[4.1 Enable spreadsheet component and rich text editor in object form 5](#_Toc75292588)

[4.2 Customization of parents and children sections in Object type form 6](#_Toc75292589)

[5 Dynamic Property plugin 7](#_Toc75292590)

[5.1 Register a dynamic property 7](#_Toc75292591)

[6 Entity validation plugin 8](#_Toc75292592)

[7 Overview of ELN Settings: 8](#_Toc75292593)

[7.1 Enabling Barcodes 8](#_Toc75292594)

[7.2 Enabling Zenodo Export 8](#_Toc75292595)

[7.3 Creating a new storage 8](#_Toc75292596)

[7.4 Additional Options in Settings 9](#_Toc75292597)

[8 Users management 9](#_Toc75292598)

[8.1 User management in the ELN Interface 9](#_Toc75292599)

[8.2 User management in the new admin interface 10](#_Toc75292600)

[9 Space management 10](#_Toc75292601)

# Overview

In this tutorial we will learn how to use the openBIS admin functionalities.

Topics covered:

1. Registration of Object types in the new admin UI
2. Dynamic properties
3. Validation plugins
4. ELN Settings
5. Users rights management in new admin UI
6. Space management in core UI

Experiments/Collection types and Dataset type are not covered, but registration of these types is similar to Object types.

# Object types registration

This part of the tutorial covers the registration of new Objects types in the new admin UI:

<https://openbis-training.ethz.ch/openbis/webapp/openbis-ng-ui/>

We will register 2 Object types: **Tissue** and **Patient**.

## Registration of a new Object type

We will now register a new Object type called **USERNAME\_TISSUE**, where **USERNAME** is the username you used to login to openBIS. E.g. BARILLAC\_TISSUE or PBHOUMIK\_TISSUE.

*We need to add the prefix to the Object type, because in this training we are working on a shared instance and we cannot create the same Object type with the same code more than once in openBIS. Codes are unique.*

1. Click on **Object Types** in the **Types** menu**.**
2. Click the blue **Add** button at the bottom.
3. A **New Object Type 1** tab opens.
4. Enter USERNAME\_TISSUE in the **Code** field
5. Enter TIS in the **Generated code prefix** field
6. **Save**

Note: Codes CANNOT be changed after registration. This statement is valid for codes anywhere in openBIS.

### Edit an Object type

We now want to edit the **USERNAME\_TISSUE** Object type to add some fields/properties, which constitute the metadata. For this Object type we want to have the following metadata: *Sample ID, CRO (=Contract Research Organization), Notes, Tissue type*. We will now add the first 3 properties:

1. Click on the **Edit** button in the **USERNAME\_TISSUE** object
2. Click on **ADD SECTION** and enter General Info in the **Name** field
3. Click on **ADD PROPERTY** and enter the Sample ID property as follows:
   1. Leave the **Scope** to **Local**. This can be *Local* or *Global* depending on whether this is a local property specific to that object or can be used for other objects as well.
   2. Enter SAMPLE\_IDin the **Code** field. There should be no spaces in the Code field.
   3. Select **VARCHAR** in the **Data Type** field.
   4. Enter Sample IDin the **Label** field.
   5. EnterIdentifier of Samplein the **Description** field.
4. Next add the following properties inside the same section:
   1. CRO(VARCHAR).
      1. Scope=local**.**
      2. Add Contract Research Organization in the **Description** field.
   2. Notes.
      1. Scope = **Global.** This property already defined in openBIS, so you can select this from the list of global properties (in the Code field).

### Create a vocabulary entry

In the exercise above we created a new Object type called **USERNAME\_TISSUE**, with the properties *Sample ID, CRO, Notes*.

We now want to create a Controlled Vocabulary that we can later assign to this Object type as a property of type Controlled Vocabulary. This Data Type gives us an option to create a pre-defined list of values for a particular field.

We will now create a Controlled Vocabulary called **Tissue Type**, which contains 3 options: *Lung, Liver, Kidney*. Also in this case we will add a prefix to the Code, so that this can be unique in the database.

1. In the Main tab **TYPES,** there is a **Vocabulary Types** section. Click on it. Select the **ADD** button at the bottom. A new tab called **New Vocabulary Type 1** opens.
2. In the **Code** field enter USERNAME\_TISSUE\_TYPES**. USERNAME** is the username you used to login to openBIS**.**
3. Click on the **ADD TERM** button and start adding the terms for the dropdown:
   * 1. Enter LUNG in the **CODE** field and Lungin the **Label.**
     2. Click on **ADD TERM** again & enter the next two terms: Liverand Kidney**.**
4. **Save.**
5. Now we have created a **Vocabulary Type** called USERNAME\_TISSUE\_TYPES**.**

Note: It is possible to change the **Label** in a **Controlled Vocabulary**. However, changing the **Code** is not possible.

Now we want to assign this vocabulary to the Object type **USERNAME\_TISSUE**:

1. Edit the Object type **USERNAME\_TISSUE.**
2. Select the **General info** section.
3. Add a new **local** property in this section called Tissue types of type **Controlled vocabulary** and assign the previously created vocabulary to this.
4. **Save**.

## Registration of a second Object type

Now we want to register a second Object type, called **Patient**.

Also in this case we will add the **USERNAME** prefix to the Object type’s Code.

Register a new Object type with **Code**= USERNAME\_PATIENT and **Generated Code Prefix** = PAT.

**Local properties** to add in a section called Patient info:

1. Patient ID (Varchar),
2. Previous treatments (Multiline Varchar)
3. Age (Integer)
4. Spreadsheet (XML)

**Add the following global properties**:

1. Initial Weight (Kg) (Real)
2. Final Weight (Kg) (Real)

# Create Collection folders in ELN User Interface

Now we will switch to the ELN User Interface: <https://openbis-training.ethz.ch/openbis/webapp/eln-lims/>?

We want to create a Collection for Tissues and one for Patients.

The first thing we need to do is enable the new Objects types we created to show in drop down menus, otherwise we will not see them anywhere in the ELN.

1. Go to the **Settings** under **Utilities** in the main menu and **Edit** it. In this case, we only have the General Settings. In the instances at Empa, usually there is one Settings for each group of a lab. The person who is appointed admin for a group can edit the group Settings. This is not the same as an Instance admin.
2. Scroll down till you reach **Object Type definitions Extension**.
3. Open your **Username Tissue** Object type
4. Click on **Show in drop downs** option.
5. Do the same for your Object type **Username Patient**
6. **Save**

Now we will create the Collection Folders under the **Materials** Space.

1. Click on the **Materials** space and create **+New Project**, called USERNAME\_SAMPLES & **Save.**
2. Click on the newly created Username Samples folder & select **+New Collection**.
3. Enter TISSUES in the **Code**, Tissues in the **Name** field & select your *Username Tissue* as the **Default Object type** from the dropdown**.**
4. **Save.**

We can follow similar steps to create a Collection folder for Patients in the **Materials** Space. In this case, we create a project folder called USERNAME\_PATIENTS.

# ELN Settings for Object types customization

Some components of the Object forms can be customized by an admin in the ELN Settings, in the ELN UI. This can be done by an Instance admin, but also by a group admin in the case of a multi-group openBIS instance.

## Enable spreadsheet component and rich text editor in object form

Click on **+new Username Patient** in your **Username** **Patients** folder. You will notice that:

1. **Previous treatment** is a simple large text field.
2. **Spreadsheet** is a simple large text field.

Two widgets are currently available for some property types and they need to be enabled in the Settings:

1. **Word Processor**: Rich Text Editor available for properties of type MULTILINE\_VARCHAR.
2. **Spreadsheet**: Spreadsheet component available for properties of type XML.

We want to have a rich text editor enabled for the property **Previous Treatment** and the spreadsheet component enabled for the property **Spreadsheet** in the Object type **USERNAME\_PATIENT**.

1. Go to the **Settings** under **Utilities** in the main menu and **EDIT**.
2. Go to the **Custom Widgets section**. There is a **+** sign and we can click on that to assign a new widget.
3. Scroll down the list and select your property and the component to add:

* **Username Patient.Previous treatments** -> **Word Processor**
* **Username Patient.Spreadsheet** -> **Spreadsheet**

1. **Save** the settings

Now refresh the browser page, navigate to your **Username** **Patients** Collection folder and open the form to register a new patient. The form now shows the rich text editor for the **Previous treatment** property and the Spreadsheet component for the **Spreadsheet** property.

We can now register one Patient with **Patient ID**=P176; **Age** = 55; **initial weight** = 75.0; **final weight**=67.5.

## Customization of parents and children sections in Object type form

When a new Object type is registered, the **Parents** and **Children** sections are always present by default in the Object form and they are as shown below:



You can see this if you go to your **Username** **Tissues** collection and open the form to register a new tissue.

These sections are however customizable from the **Settings**.

We will now customize the form for your Object type **Username Tissue** that was created before. We will remove the **Children** section from the form, rename the **Parents** section to Connectionsand customize the form to always show **Patients** as default parents.

1. Click on **Settings-EDIT**.
2. Scroll down to **Object Type definitions Extension** and click on your Object type **Username Tissue**.
3. In the **Settings for Parents & Children** section there is a **Disable Section** option. Disable this for Children.
4. Click on the **+** in the **Hints for** section**.** 
   1. Select **Parents** in the dropdown
   2. Select your Username Patient in the **Type.**
   3. Label = Patient.
5. **Save**.

If you want to make a parent mandatory, the **Min** field should have the value of 1. This means that you need to have at least one parent. Of course, if the minimum is more than 1, you should enter the appropriate number. If there is a maximum of parents of a give type that can be entered, you should enter the number in the **Max** field.

Now let’s go back to the **Username Tissues** collection and let’s open again the form to register a new Tissue. We can now see the difference in the form: there is no **Children** section, there is a section called **Connections** (instead of Parents) and **Patient** has been added to the form by default.

In this way you can customize all your Object types forms in openBIS.

You can now register a tissue, with **Sample ID**=674; **Tissue type**=lung. Add your previously registered patient as **parent**. Assigning this relationship means that the tissue we are registering was taken from that patient. In order to add the parent, you need to click on the select button, followed by clicking on Exports and.. column, where the dropdown will show **Add selected**. Click Save.

# Dynamic Property plugin

A dynamic property is a property that is calculated on the fly by openBIS. These properties are controlled by Jython scripts, in the **Dynamic Property Plugins.**

Dynamic property plugins can be created in the **new admin interface**: <https://openbis-training.ethz.ch/openbis/webapp/openbis-ng-ui/>

We now want to write a small script that calculates the difference between the **initial weight** and **final weight** of a **Patient**. We will the assign this script to a property called **weight difference** in the Object type **Username** **Patient**.

The script is already created in the training instance. These are the steps that were followed to register it:

1. Go to the **Tools** tab in the menu
2. Click on **Dynamic Property Plugins**
3. Click the **ADD** button
4. Enter PATIENT.weight\_difference in the **Name** of the script
5. Select **OBJECT** in **Entity Kind**. This is because we want to use it for a property in an Object type.
6. Insert the following content in the **Script** field.

def calculateValue():

initial\_weight = float(entity.propertyValue('INITIAL\_WEIGHT'))

final\_weight= float(entity.propertyValue('FINAL\_WEIGHT')) weight\_difference=0

if initial\_weight > final\_weight:

weight\_difference = initial\_weight - final\_weight

return weight\_difference

def calculate():

return calculateValue()

1. Evaluate the script: select the patient that was registered before from the list of objects to evaluate that the script runs correctly.

## Register a dynamic property

Now we want to register a new dynamic property, called **weight difference,** in the Object type **Username** **Patient**. This property will use the script above:

1. Edit the Object type **USERNAME\_PATIENT.**
2. In the **Patient info** section add a **local** property called Weight difference of type **REAL.**
3. Assign the **Dynamic Property Plugin** called **PATIENT**.**weight\_difference** to it.
4. **Save.**

Now we can go back to the ELN interface and open the **Username Patient** entry we registered before. We can see that the **weight difference** property is automatically filled in.

Note: dynamic properties are calculated properties, and as such they do not appear in Edit mode.

# Entity validation plugin

Entity validation plugins are scripts used to ensure metadata consistency. These are Jython scripts, similar to the script we saw before. A validation scripts runs every time an entity is registered or updated to check for specified inconsistencies.

In the ELN two default validation plugins are provided to check the consistency of start date and end date in an Experiment or Experimental Step. If a user enters by mistake an end date that is before the start date in such entries, openBIS throws an error.

# Overview of ELN Settings:

We will now go back to the ELN UI to look at some additional options in the Settings: <https://openbis-training.ethz.ch/openbis/webapp/eln-lims/>?

## Enabling Barcodes

openBIS offers the possibility to assign barcodes to samples and generate barcodes:

1. if you create your own Object types for which you want to use the barcode functionality, the **$Barcode** global property needs to be added to the type.
2. The Barcodes functionality needs to be enabled from the **Settings:**
   1. Click on Settings & then **EDIT**.
   2. In the list of Main menu items, click on **showBarcodes.**
   3. **Save**.
3. Refresh the browser page and the **Barcode Generator** option appears in **Utilities**.

## Enabling Zenodo Export

Zenodo is a data repository for sharing data: <https://zenodo.org/> . The connection to Zenodo is available in all openBIS instances, but needs to be enabled in the ELN UI if needed by the lab. This allows to export data and metadata stored in openBIS directly to Zenodo.

To enable the connection to Zenodo:

1. Go to the Settings in the main menu
2. Select the **Settings** and **Edit.** Then select s**howZenodoExportBuilder**.
3. After refreshing the web page, **Exports to Zenodo** will be shown under **Exports** in the **Utilities** main menu.

In order to use the Zenodo export functionality, users need to create a Zenodo API token in Zenodo and add it to their User Profile in openBIS, as documented here: <https://openbis.ch/index.php/docs/user-documentation/data-export/export-to-zenodo/>

## Creating a new storage

Each lab can customize their storages. We will now create a new -20°C freezer, which has 5 shelves and 4 racks per shelf. We also want to ensure that users fill in the information on box position when entering the storage information:

1. Click on **+New Storage**
2. Select **Show identification** info in the **More**.. dropdown
3. Enter USERNAME\_MINUS\_20FREEZER in the code field
4. Enter -20°C freezer in the **Name** field
5. Enter 5 in the **Number of rows**
6. Enter 4 in the **Number of columns**
7. Select Box position validation in **Validation level**
8. **Save**

We can see the storages in the **Storage Manager** under **Utilities** in the **main menu**.

## Additional Options in Settings

Additional functionalities in the Settings allow an admin to:

Customize what to see in the **Main menu.**

Create partly pre-filled **templates** for Object types.

Specify which Spaces should go in the **Inventory**. When Spaces are manually created, they are automatically assigned to the Lab Note book. In the Settings it is possible to specify that they should go in the Inventory.

Associated a file extension to a given **Dataset** type. E.g. files of type .ppt could always be saved in Datasets of type Attachment.

Enable the **Archiving button for datasets**. This should be enabled in openBIS instances for which archiving to Long Term Storage is set up.

Hide sections by default.

Visualization of **available storage space**.

# Users management

Users management can be done from the ELN interface in a limited way and more extensively from the admin interface.

Note: Please be aware that this does not apply to multi-group instances, where users are automatically registered and roles automatically assigned.

## User management in the ELN Interface

Users can be added via the **User Manager** under **Utilities** in the main menu.

The ELN UI supports registration of users with 3 types of authentication (if these are configured in the server):

1. LDAP
2. File based
3. Switch aai/edu id

Users registered from the ELN interface are assigned some default roles:

1. *Space user* roles for all inventory spaces
2. *Space admin* role for their own space in the lab notebook that is automatically created upon registration via the ELN.

The ELN does not provide an overview of the Roles, only of registered users.

*We will show an example of how a user can be registered from the ELN UI and see which roles are assigned to it.*

## User management in the new admin interface

In the new admin interface LDAP and Switch AAi authentication is supported since openBIS v20.10.2.

The roles management can be done from the new admin interface independently of the authentication system used.

To add a new user from the admin interface:

1. Select the **USERS** tab& click on **ADD button.** This opens a new tab called **New User 1.**
2. In the **User Id** field, the LDAP username of the user needs to be entered.
3. Click on **ADD ROLE** to add roles to a user**.**

It is also possible to create groups of users and assign roles to a whole group.

*We will show an example of adding a user to whom we assign SPACE\_OBSERVER role for the MATERIAL space. We will also create a group to which we will add two users and assign SPACE\_ADMIN rights for the DEFAULT space to this group.*

# Space management

Space management is currently only supported in the Core UI: <https://openbis-training.ethz.ch/openbis/>

It is planned that this will be added to the ELN UI.

*We will show how to add a new Space in the core UI:*

* 1. Go to the **Admin** tab
  2. Select **Spaces**
  3. Click **Entity: Add Space** at the bottom of the page
  4. Enter the **Code** of a Space, e.g EQUIPMENT
  5. **Save**