
guide template Documentation

Release 0.1.1

Open Microscopy Environment

May 10, 2023

Contents

1	Using OMERO.mde to edit ome metadata	3
1.1	Description	3
1.2	Setup	3
1.3	Step-by-Step	3
2	Customize OMERO.mde	7
2.1	Description	7
2.1.1	Customize user input form of OMERO.mde	8
2.1.2	Add custom objects to OMERO.mde	11
2.1.3	Predefine object values in OMERO.mde	15
3	Use input templates for OMERO.mde	19
3.1	Step-by-Step	19

OMERO.mde is an extension of OMERO.importer to get an overview of available metadata provided by the selected image container and annotate images at import step by standardized key-value input forms. Changes of metadata provided by image will be saved as key-value pairs.

Contents:

Using OMERO.mde to edit ome metadata

1.1 Description

By using OMERO.mde you can annotate your images with standardized key-values at the import step.

We will show:

- How to enable OMERO.mde feature for the OMERO.importer
- How to edit ome metadata
- How to modify the object tree

1.2 Setup

Install the OMERO.insight client as described in [import-desktop-client.html#setup](#)

Go to `OMERO.insight/config/` and edit `container.xml` and `containerImporter.xml`. To enable the OMERO.mde set `boolean` to `true` in:

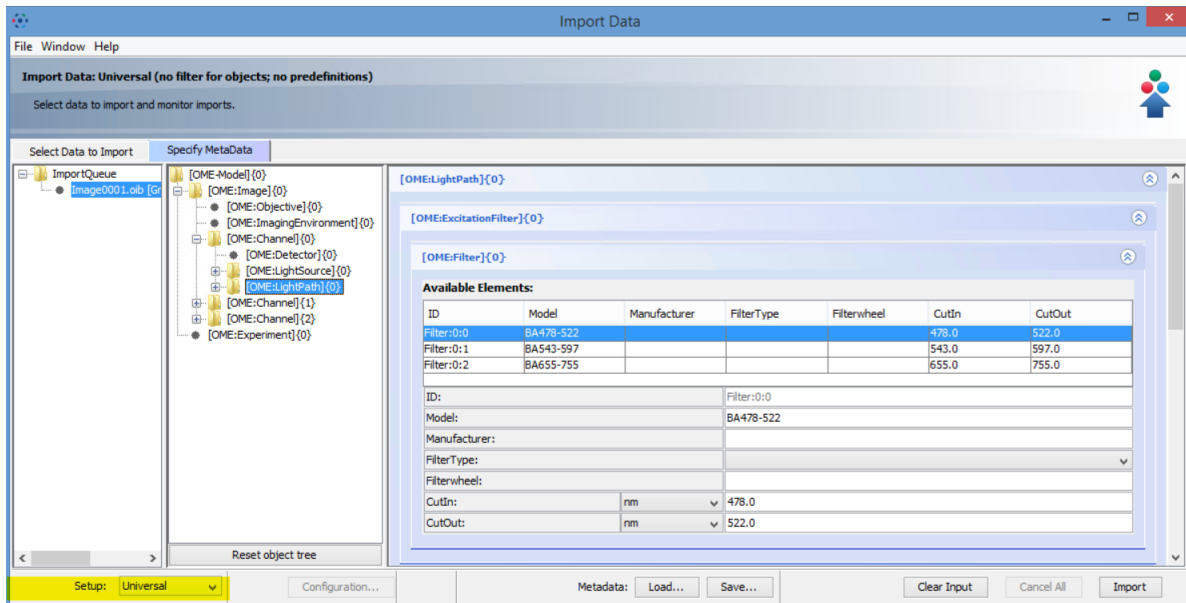
```
<!-- mde on/off -->  
  
<entry name="omero.client.import.mde.enabled" type="boolean">false</entry>
```

and save the files.

1.3 Step-by-Step

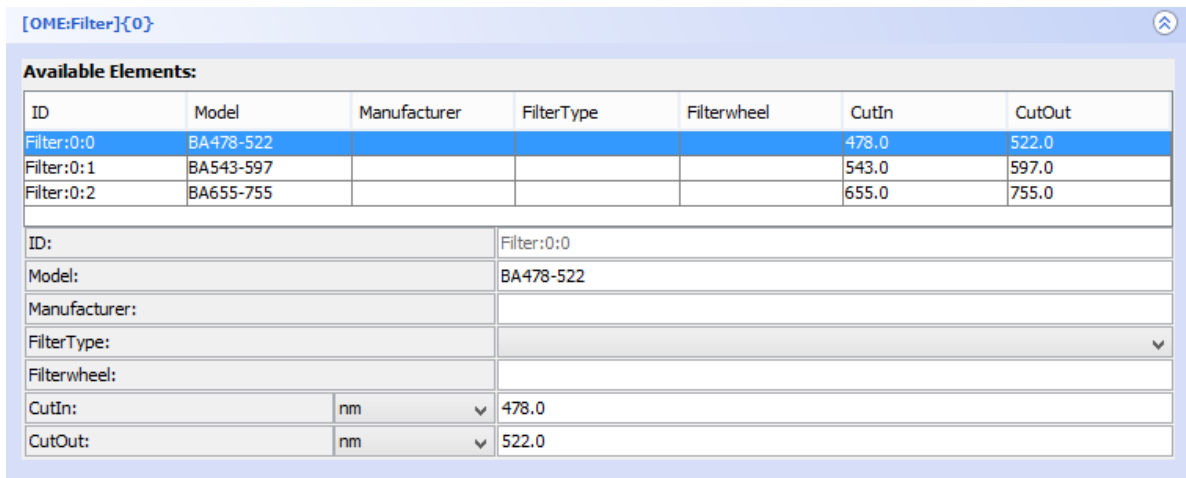
1. Start OMERO.importer and login.
2. Select your files for import and add it to the queue.

- Click on the MDE Button MDE >> at the right bottom pane.



On the left pane you find the import queue, on the middle pane the object tree and on the right pane the input forms.

- In the `Specify MetaData` window, select a file to show available metadata stored in the file container. After reading the metadata of the file by Bio-Formats you can find in the middle pane the object tree with standard OME objects like Image, Experiment, Detector etc. The properties of the selected object in the object tree and its subtree are displayed on the right pane as key-value pairs. For instrument objects there is a table with all objects of the same type as `available Elements` as defined in the file container.



Now you can editing the properties of the object.

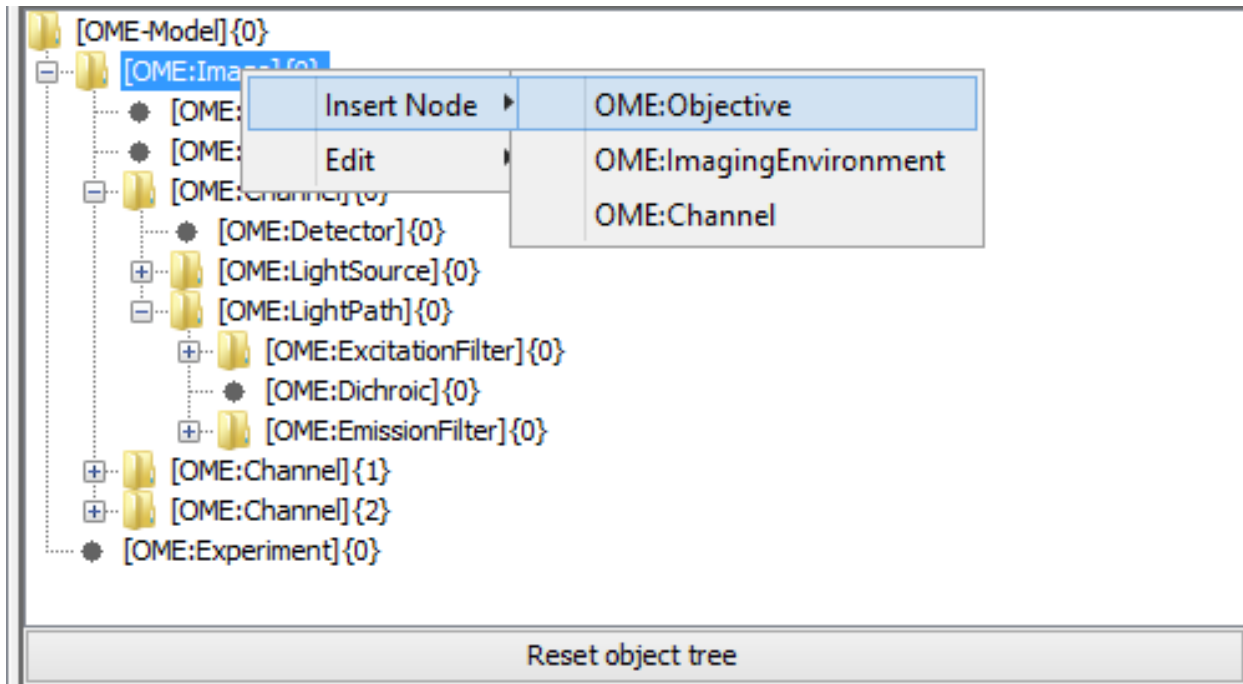
You can:

- change the unit
- add or change the value or select another value from the list of possible values
- replace all properties of an instrument object by selecting an element from the `available Element` list

Modifying the object tree

Add an additional object in the object tree:

1. Click with the right mouse button on the object for which a child object should be created.
2. Select an object from list under `Insert Node` to insert an object.



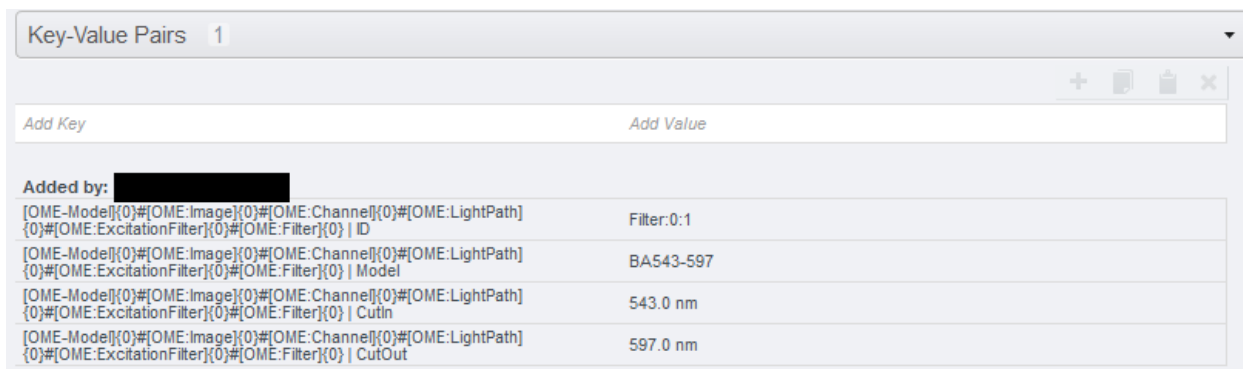
Copy and paste of a subtree:

1. Click with the right mouse button on the root of the subtree you want to copy.
2. Select `Edit > Copy` to copy this subtree.
3. Select the object where you want to insert the subtree (you can only insert the subtree where this type of rootobject is allowed as a child) and select by right mouse click `Edit > Paste`. Only the structure of the subtree will be inserted, not the values of the properties.

To reset to original object tree based on file information press the `Reset object tree` button. To reset all object and properties changes press the `Clear Input` button.

If you edit the properties and object for a folder, all files in the folder inherit the changes.

All changes will be add as key-value annotation to the file after import.



2.1 Description

You can configure and customize the OMERO.mde by defining and configuring elements in a `mdeConfiguration.xml` file. A template of this file can be found in the `config/` directory of the installed OMERO.insight installation or at `mdeConfiguration.xml`.

Structure of `mdeConfiguration.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
  <MDEConfiguration>
    <MDEPredefinitions> <!-- Predefinitions for properties of objects group by_
↳ setups--!>
      <SetupPre Name="SetupName">
        <ObjectPre>
          <TagData.../>
          ...
        </ObjectPre>
        ...
      </SetupPre>
      ...
    </MDEPredefinitions>
    <MDEObjects>
      <Definitions> <!-- Definitions of objects, properties and hierarchy--!>
        <ObjectDef Type="ObjectName">
          <TagData.../>
          ...
          <Parents.../>
        </ObjectDef>
        ...
      </Definitions>
      <Configurations> <!--Definition of setups and configuration of objects_
↳ and properties belonging to the setup --!>
        <SetupConf Name="SetupName">
```

(continues on next page)

(continued from previous page)

```

        <ObjectConf Type="ObjectName">
            <TagDataProp.../>
            ...
        </ObjectConf>
        ...
    </SetupConf>
    ...
</Configurations>
</MDEObjects>
</MDEConfiguration>

```

The location of the configuration file should be in the `config/` folder or in `<userHome>/omero/`. Please specify the location you will use by editing `container.xml` and `containerImporter.xml`. To use `config/` as location:

```

<!-- mde config file location (. for in config dir; omero for local user omero dir) --
↔>
    <entry name="omero.client.import.mde.path">.</entry>

```

or for `<userHome>/omero/` as location:

```

<!-- mde config file location (. for in config dir; omero for local user omero dir) --
↔>
    <entry name="omero.client.import.mde.path">omero</entry>

```

and save the files. Here we will use `config/` as location for `mdeConfiguration.xml`. Save the configuration file under `config/` and restart the OMERO.mde to load the custom configurations.

You can customize the OMERO.mde in following point:

2.1.1 Customize user input form of OMERO.mde

Description

You can generate a filter view of the sets of objects. Also you can hide properties for objects and change the default unit. You can mark fields as required to restrict your input forms to objects with required metadata (OMERO.insight version $\geq 5.5.15$). We will show in this section how to create a new setup and how to generate customized input forms of available objects in this setup.

Step-by-Step

1. Open the `mdeConfiguration.xml` file under `config/`.
2. Insert a new `<SetupConf>` element under `<Configuration>` and specify the name for the setup (here `MyCustomSetup`):

```

<MDEObjects>
    <Configurations>
        <SetupConf Name="MyCustomSetup"/>
    </Configurations>
</MDEObjects>

```

3. Define now the object elements `<ObjectConf>` that should be enabled (OME standard objects) or visible/choosable (other). Please pay attention of the hierarchy (the parent of and object has also to be defined

here). If an object is not mentioned, it is disabled and also all child objects (OME standard objects) or not visible (defined custom objects):

```
<MDEObjects>
  <Configurations>
    <SetupConf Name="MyCustomSetup">
      <ObjectConf Type="OME:Image"/>
      <ObjectConf Type="OME:Objective"/>
    </SetupConf>
  </Configurations>
</MDEObjects>
```

For this objects also add <TagDataProp> elements for the properties where you want to change the unit or you want to hide this property:

```
<MDEObjects>
  <Configurations>
    <SetupConf Name="MyCustomSetup">
      <ObjectConf Type="OME:Image"/>
      <TagDataProp Name="Description" Unit="" Visible="false"/>
      <ObjectConf Type="OME:Objective"/>
    </SetupConf>
  </Configurations>
</MDEObjects>
```

4. Save the file and restart OMERO.importer to load the new configuration for OMERO.mde. Select your images or dataset for import. Switch to the OMERO.mde pane and select your setup to load your customize forms.

Setup: Universal

Import Data: Universal (no filter for objects; no predefinitions)

Select data to import and monitor imports.

Select Data to Import | Specify MetaData

ImportQueue

- 0016_Amp_med_GB
 - [OME-Model]{0}
 - [OME:Image]{0}
 - [OME:Objective]{0}
 - [OME:ImagingEnvironment]{0}
 - [OME:Channel]{0}
 - [OME:Experiment]{0}

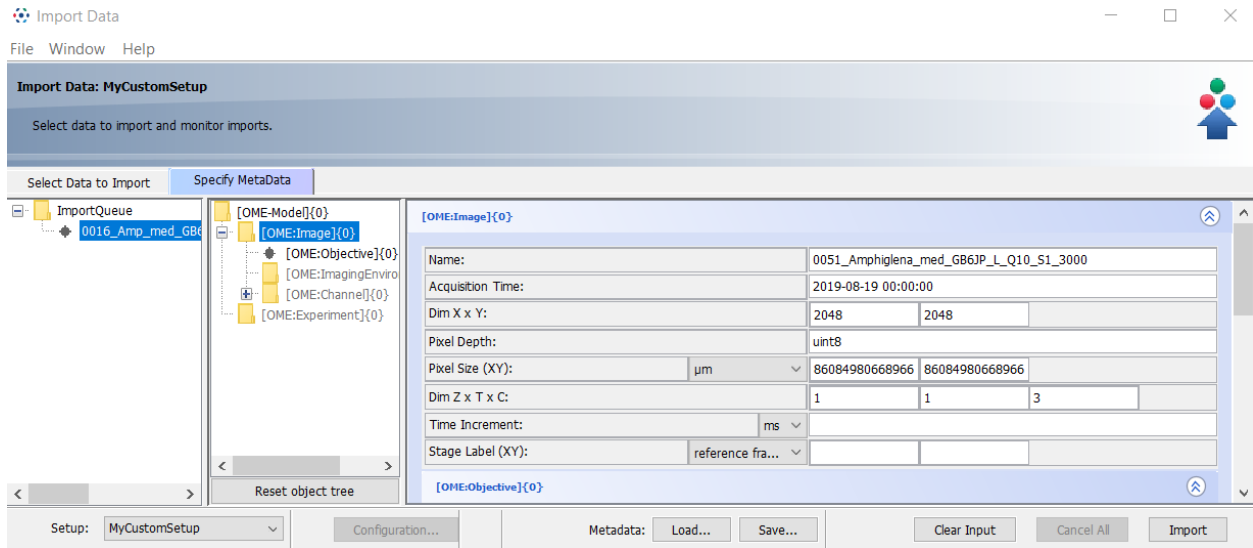
Reset object tree

[OME:Image]{0}

Name:	0051_Amphiglena_med_GB6JP_L_Q10_S1_3000		
Description:			
Acquisition Time:	2019-08-19 00:00:00		
Dim X x Y:	2048	2048	
Pixel Depth:	uint8		
Pixel Size (XY):	µm	86084980668966	86084980668966
Dim Z x T x C:	1	1	3
Time Increment:	ms		
Stage Label (XY):	reference fra...		

Setup: Universal | Configuration... | Metadata: Load... Save... | Clear Input | Cancel All | Import

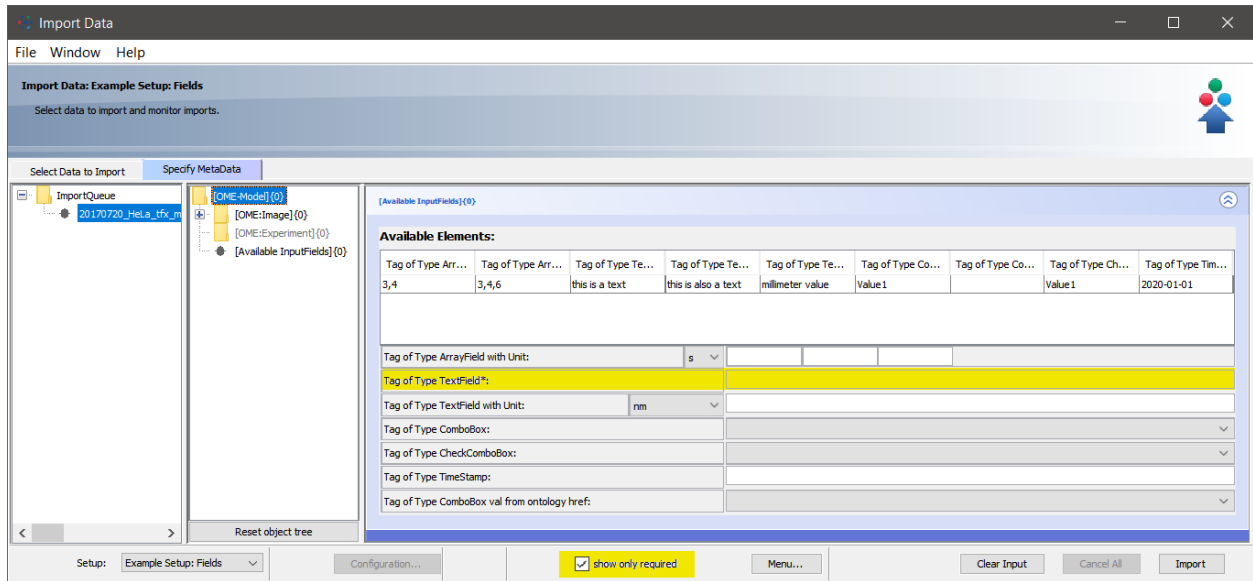
Setup: MyCustomSetup



You can mark any field as required by adding `Required = "true"` as an attribute of `TagDataProp` (OMERO.insight version $\geq 5.5.15$). By selecting **show only required**, you can restrict the displayed objects of the selected setup to those that contain at least one required metadata field:

```
<MDEObjects>
  <Configurations>
    <SetupConf Name="Example Setup: Fields">
      <ObjectConf Type="OME:Image"/>
      <TagDataProp Name="Name" Unit="" Visible="true" Required="true"/>
      <ObjectConf Type="OME:Objective"/>
    </SetupConf>
  </Configurations>
</MDEObjects>
```

Setup: Example Setup: Fields



2.1.2 Add custom objects to OMERO.mde

Description

By using OMERO.mde you can create standardized key-value input forms with certain values and units. You can group key-values by creating an object module with the key-values as properties. We will show in this section how to create new objects for OMERO.mde and how you could create new key-values as properties of objects.

Step-by-Step

1. Open the `mdeConfiguration.xml` file under `config/`.
2. Insert a new `<ObjectDef>` element under `<Definitions>` and specify the name for the object (here `MyCustomObject`):

```
<MDEObjects>
  <Definitions>
    <ObjectDef Type="MyCustomObject"/>
  </Definitions>
  <Configurations.../>
</MDEObjects>
```

3. Define now the properties (key-values) for your new object. For every property you have to add an `<TagData>` element. See below section `Type` of input fields.
4. Define now in which hierarchy the object should be insert. For that define the parent element:

```
<MDEObjects>
  <Definitions>
    <ObjectDef Type="MyCustomObject">
      <TagData.../>
      ...
    <Parents.../>
    </ObjectDef>
  </Definitions>
  <Configurations.../>
</MDEObjects>
```

At this point we have specified an insertable object. You can insert this object with right-click on the specified parent node. If you want this object will be automatically inserted to the object tree, you have to specify this also in the `SetupConf` section of the corresponding setup:

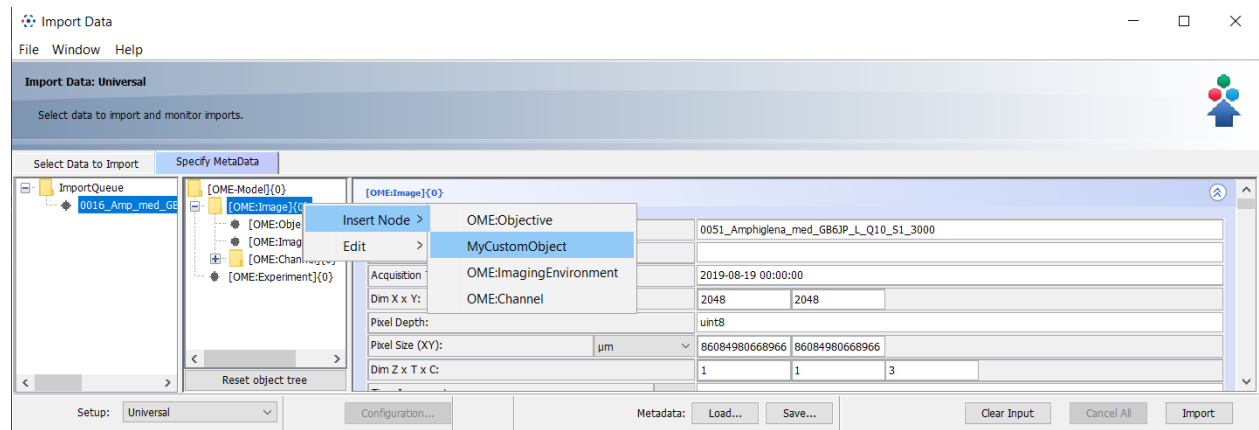
```
<MDEObjects>
  <Configurations>
    <SetupConf Name="MyCustomSetup"
      <ObjectConf Type="OME:Image" ...>
      <ObjectConf Type="OME:Objective"...>
      <ObjectConf Type="MyCustomObject" Insert="true" InsertPoint="OME:Image" ..
↔.>
    </SetupConf>
  </Configurations>
</MDEObjects>
```

Example

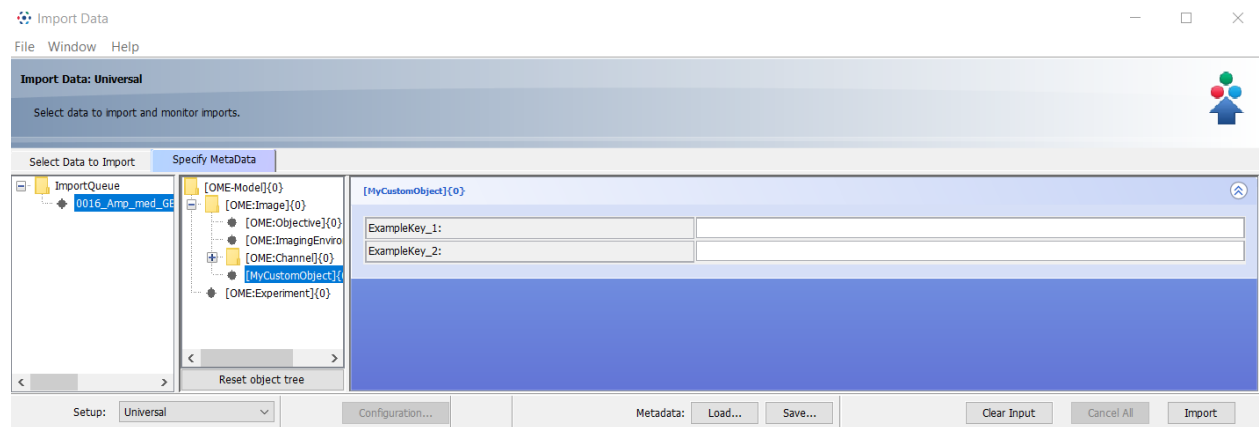
Specification like:

```
<MDEConfiguration>
  <MDEPredefinitions...>
  <MDEObjects>
    <Definitions>
      ...
      <ObjectDef Type="MyCustomObject">
        <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
          Unit="" Value="" Visible="true" />
        <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
          Unit="" Value="" Visible="true" />
        <Parents Values="OME:Image" />
      </ObjectDef>
    </Definitions>
  </MDEObjects>
</MDEConfiguration>
```

will create an insertable object at OME : Image object.



that has input form for two specified key-value pairs



With the additional specification in *Configuration*:

```
<MDEConfiguration>
  <MDEPredefinitions...>
  <MDEObjects>
    <Definitions>
```

(continues on next page)

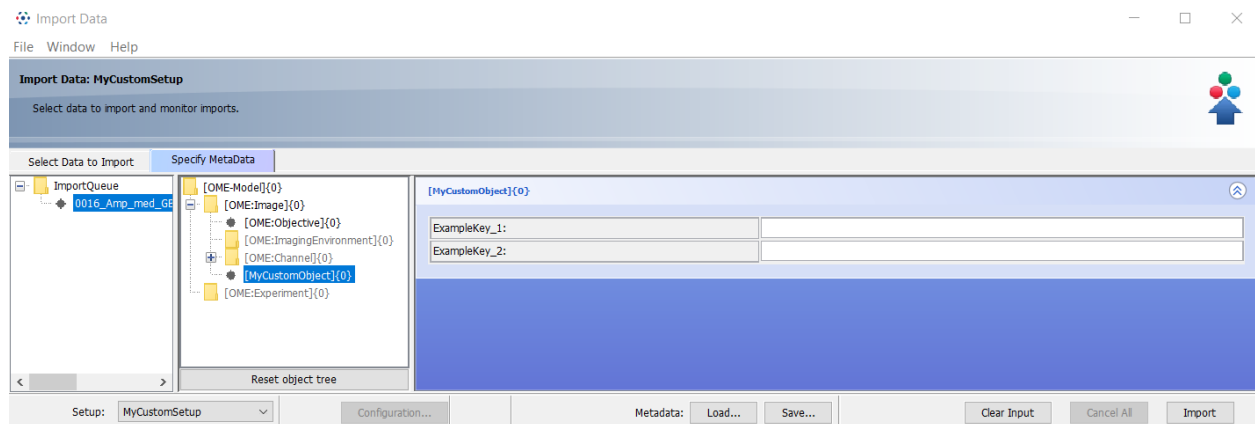
(continued from previous page)

```

...
<ObjectDef Type="MyCustomObject">
  <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
    Unit="" Value="" Visible="true" />
  <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
    Unit="" Value="" Visible="true" />
  <Parents Values="OME:Image" />
</ObjectDef>
</Definitions>
<Configuration>
  <SetupConf Name="MyCustomSetup">
    <ObjectConf Type="OME:Image"...>
    <ObjectConf Type="OME:Objective"/>
    <ObjectConf Type="MyCustomObject" Insert="true" InsertPoint="OME:Image
↩"/>
  </SetupConf>
</Configurations>
</MDEObjects>
</MDEConfiguration>

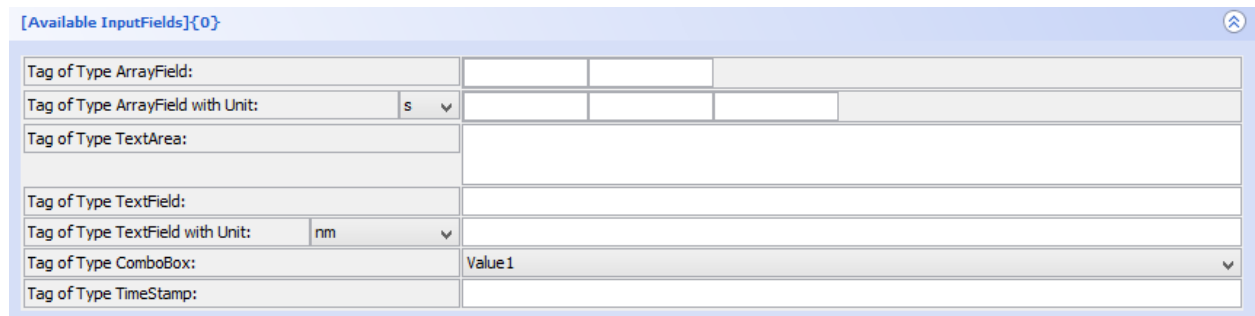
```

will lead into following object tree if you select the setup *MyCustomSetup*



Type of input fields

There are different editor input field types for the element `<TagData>`. You can find this example by using the example `mdeConfiguration.xml` and insert a *Available InputFields* object by right-clicking on OME-Model node or it is inserted automatically when you select the setup `Example Setup:Fields`.



You can specify the different types like:

TextField define like:

```
<TagData DefaultValues=""
  Name="Tag of Type TextField"
  Type="TextField"
  Unit=""
  Value=""
  Visible="true" />
```

TextField with unit define like:

```
<TagData DefaultValues=""
  Name="Tag of Type TextField with unit"
  Type="TextField"
  Unit="nm"
  Value=""
  Visible="true" />
```

TextArea define like:

```
<TagData DefaultValues=""
  Name="Tag of Type TextArea"
  Type="TextArea"
  Unit=""
  Value=""
  Visible="true" />
```

ArrayField define like (for an array of 2 elements):

```
<TagData DefaultValues="2"
  Name="Tag of Type ArrayField"
  Type="ArrayField"
  Unit=""
  Value=""
  Visible="true" />
```

ArrayField with unit define like (for an array of 3 elements):

```
<TagData DefaultValues="3"
  Name="Tag of Type ArrayField with unit"
  Type="ArrayField"
  Unit="s"
  Value=""
  Visible="true" />
```

ComboBox define like:

```
<TagData DefaultValues="Value1,Value2,Value3"
  Name="Tag of Type ComboBox"
  Type="ComboBox"
  Unit=""
  Value=""
  Visible="true" />
```

ComboBox with values from ontology (OMERO.insight version >= 5.5.15):

```
<TagData DefaultValues=""
  Name="Tag of Type ComboBox val from ontology href"
```

(continues on next page)

(continued from previous page)

```

    Type="ComboBox"
    Unit=""
    Value=""
    Visible="true">
    <Ontology URL_restapi="http://data.bioontology.org" Acronym="BAO" ID_
    ↪href="http://www.bioassayontology.org/bao#BAO_0150008" />
  </TagData>

```

CheckComboBox for multiple selection (OMERO.insight version $\geq 5.5.15$):

```

<TagData DefaultValues="Value1,Value2,Value3"
  Name="Tag of Type CheckComboBox"
  Type="CheckComboBox"
  Unit=""
  Value="Value1"
  Visible="true" />

```

TimeStamp define like:

```

<TagData DefaultValues=""
  Name="Tag of Type TimeStamp"
  Type="TimeStamp"
  Unit=""
  Value=""
  Visible="true" />

```

2.1.3 Predefine object values in OMERO.mde

Description

By using OMERO.mde you can generate a set of predefined values for the available objects. You can predefine different values for same objects by group it in setups.

Step-by-Step

1. Open the `mdeConfiguration.xml` file under `<userHome>/omero/`.
2. Insert a new `<SetupPre>` element under `<MDEPredefinitions>` with the name of the setup for that you want to specify the values:

```

<MDEConfiguration>
  <MDEPredefinitions>
    <SetupPre Name="MyCustomSetup">
  </MDEPredefinitions>
</MDEConfiguration>

```

3. Insert under this `<SetupPre>` element an `<ObjectPre>` element with the name of the object for that you want to specify the values:

```

<MDEConfiguration>
  <MDEPredefinitions>
    <SetupPre Name="MyCustomSetup">
      <ObjectPre ID="" Type="MyCustomObject"/>

```

(continues on next page)

(continued from previous page)

```

    </SetupPre>
  </MDEPredefinitions>
</MDEConfiguration>

```

4. Now you can copy - paste the object properties defined under <ObjectDef> for this object and then specify the Values.

NOTE: You can specify different predefined values for the same object by generate multi <ObjectPre> elements for this object. OMERO.mde than shows you a table of available Elements Predefinitions) for this object.

Example

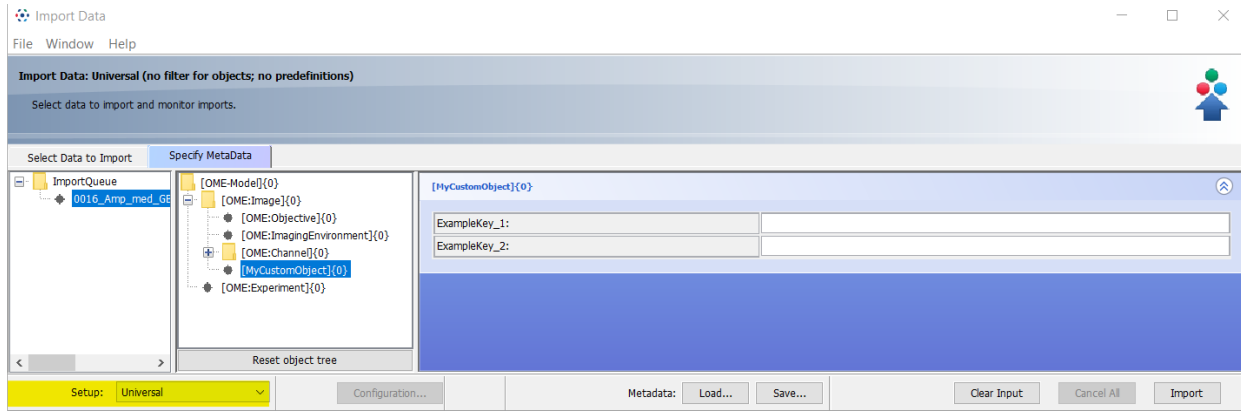
Specification like:

```

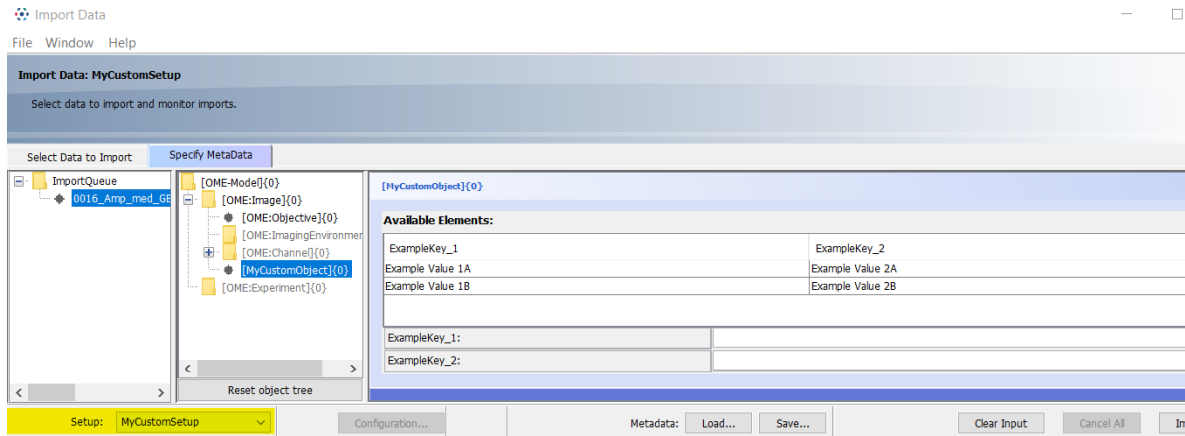
<MDEConfiguration>
  <MDEPredefinitions>
    <SetupPre Name="MyCustomSetup">
      <ObjectPre Type="MyCustomObject">
        <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
          Unit="" Value="Example Value 1A" Visible="true" />
        <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
          Unit="" Value="Example Value 2A" Visible="true" />
      </ObjectPre>
      <ObjectPre Type="MyCustomObject">
        <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
          Unit="" Value="Example Value 1B" Visible="true" />
        <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
          Unit="" Value="Example Value 2B" Visible="true" />
      </ObjectPre>
    </SetupPre>
  </MDEPredefinitions>
  <MDEObjects>
    <Definitions>
      ...
      <ObjectDef Type="MyCustomObject">
        <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
          Unit="" Value="" Visible="true" />
        <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
          Unit="" Value="" Visible="true" />
        <Parents Values="OME:Image" />
      </ObjectDef>
    </Definitions>
    <Configuration>
      <SetupConf Name="MyCustomSetup">
        <ObjectConf Type="OME:Image" />
        <ObjectConf Type="MyCustomObject" />
      </SetupConf>
    </Configuration>
  </MDEObjects>
</MDEConfiguration>

```

will create input form like



Setup:Universal



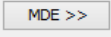
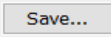
Setup:MyCustomSetup

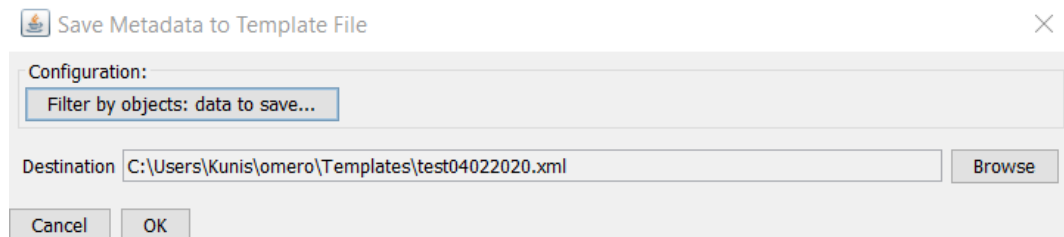
Use input templates for OMERO.mde

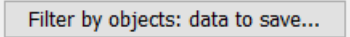
** only available from OMERO.insight v5.5.11** You can save your inputs for MDE objects in a template file for reused the inputs for another import.

3.1 Step-by-Step

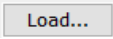
Save input:

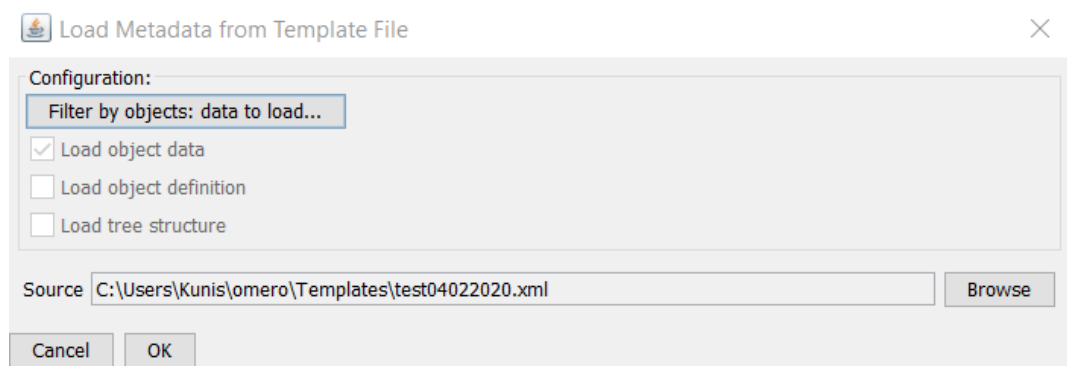
1. Start OMERO.importer and login.
2. Select your files for import and add it to the queue.
3. Click on the MDE Button  at the right bottom pane.
4. Add your inputs.
5. Select in the middle bottom pane  to get the *Save Template* dialog

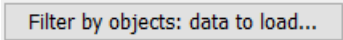


6. Select the object types for which the input should be saved. To do this press the button  and select the types.
7. Choose the location and file name for your template file.
8. Press ok to save the file.

Loading the template file for another import:

1. In the MDE Pane select the dir or file for which the input should load.
2. Select in the middle bottom pane  to get the *Load Template* dialog



3. Select the object types for which the input should be loaded. To do this press the button  and select the types.
4. Choose the location and file name of your template file.
5. Press ok to load the values.