

[EN] 08. Configuring Attribute Views

Attribute views facilitate editors' work – only those attributes are displayed in the [attribute editor](#) which are important for them. The attribute definition should be put in the XML file the URL for which has been configured in the [generalConfig.properties](#) file, in `attributesViews.url`.

The structure of the XML file which defines attribute views is presented in the example below:

```
<attributesViewsList>
  <attributesView fullView="true">
    <name lang="pl"><![CDATA[Wszystkie atrybuty]]></name>
    <name lang="en"><![CDATA[All attributes]]></name>
  </attributesView>

  <attributesView>
    <name lang="pl"><![CDATA[Tytu, autor i opis]]></name>
    <attribute required="true">Title</attribute>
    <attribute>Creator</attribute>
    <attribute descendantRequired="true" showAllDescendants="true">Description</attribute>
  </attributesView>
</attributesViewsList>
```



The word “attribute” can be understood as a basic ingredient of a bibliographic description or as an element added to the XML tag. For the sake of clarity, whenever the word is used in the second sense here, it will be expressly stated.

The main tag of the configuration file is `<attributesViewsList>`. Inside, there are `<attributesView>` tags which define one attribute view each. The user can enter the following information about the view:

- name: put in `<name>` tags with the `lang` XML tag, which defines the language (a two-letter symbol in accordance with the ISO 639 standard) of the displayed name; names should be put in the CDATA section, to make it possible to use any Unicode characters; for every view, a name must be defined in at least one language; it is recommended that names be entered for every language available in the “View” menu of the Editor Application; and
- visible attributes: for determining which attributes are to be displayed in a view; `<attribute>` tags should be used; a tag should contain the RDF name of the attribute; it is worth noting that `<attribute>` tags should not create a hierarchy which corresponds to the hierarchy of attributes in the system – all tags should be added directly inside the `<attributesView>` tag; optional XML attributes which change the behavior of the view can be used for every tag:
 - `required`: a `true/false` value, the default value is `false`; when value `true` is set, the attribute will be deemed to be required, and it will be marked with an asterisk; in such a case, when a standard and planned publication description is edited, the application will require that the user enters at least one attribute value in each available language or a language-independent value.
 - `descendantRequired`: a `true/false` value, the default value is `false`; when value `true` is set, it will be required that the user fill in the attribute or any of its **visible** subordinate attributes (at any depth of the hierarchy); such an attribute will be marked with two asterisks; just like above, the requirement will only concern standard and planned publications for all languages or, at least, a language-independent value; since the condition enforced by the `required` XML attribute is stronger than `descendantRequired`, there is no point in using both for one attribute; in such a case, `descendantRequired` will be ignored;
 - `showAllDescendants`: a `true/false` value, the default value is `false`; when value `true` is set, all subordinate attributes of the given attribute (at any depth of the hierarchy) will also be visible; subordinate attributes can still be added in separate XML tags, for example, to determine the required attributes;
- for a whole view, the `fullview` XML attribute can be added (a `true/false` value, the default value is `false`); when value `true` is set, all attributes will be visible in that view; it will still be possible to add attribute tags, for example, to determine the required attributes.