How to create an enjoyable authoring experience

George Bina

george@oxygenxml.com

@georgebina @ #oxygenxml meetup @ #xmlprague

Lightweight DITA

- Placeholders
- Hints
- Inline actions
- Inline editors for attribute information
- Remapped element names
- Add usual content automatically to elements
- Suggest attribute values

Editing Lightweight DITA topics

Demo editing a LW-DITA topic!

How LW-DITA is implemented?

- Extension of the DITA framework
  - to inherit the default DITA support
- Provide specific CSS for Lightweight DITA
  - we use a LESS file that is converted to CSS
  - provides placeholders, hints, inline actions and editors
- Define a content completion configuration file
  - provides element remapping, controlled attributes and default element content

LESS vs CSS

LESS is a superset of CSS

LESS can be used directly or compiled to CSS

Adds:

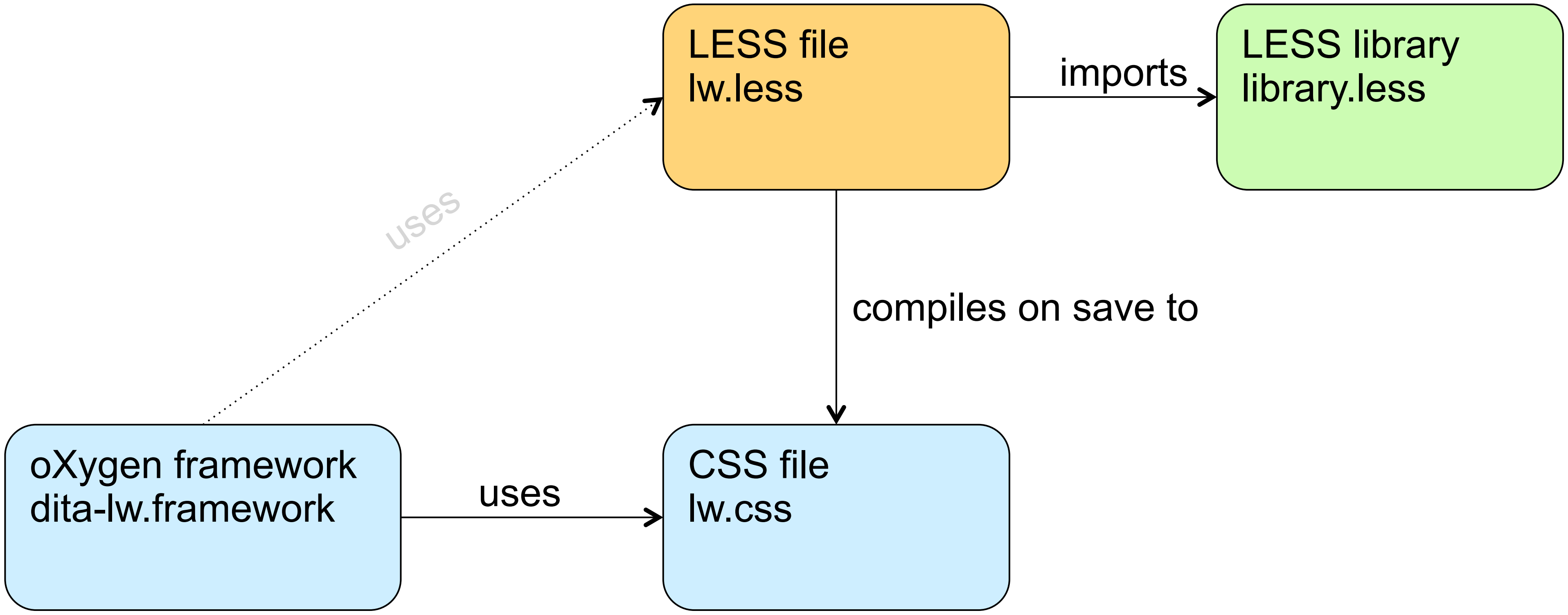
- Variables
- Mixed-ins
- etc.

Styles library

LESS library defines mixed-ins:

- .placeholder("content");
- .hint();
- .markSection("Section name");
- .deleteAction();
- .addActions({.action('actID1');.action('actID2');});
- .editable();
- .textfield("Label", "attributeName");
- .urlChooser(”Label", ”attributeName");
- etc.

LW-DITA LESS to CSS

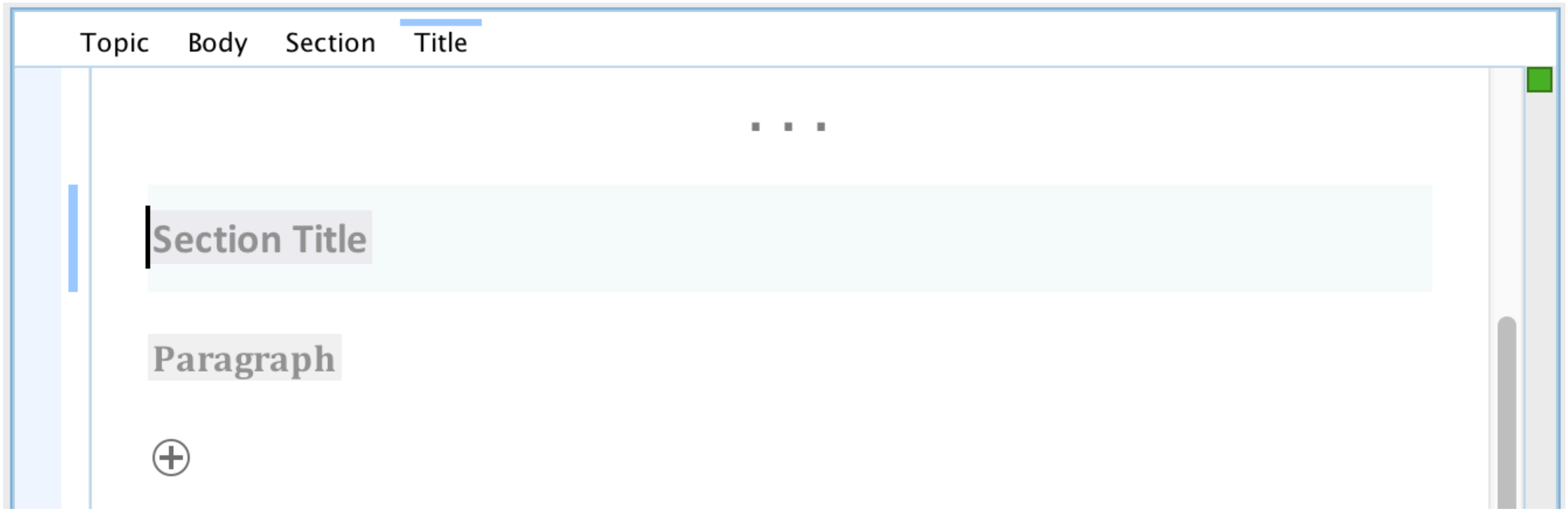


LESS file

- Placeholders
- Hints
- Inline actions
- Inline editors for attribute information

Placeholders

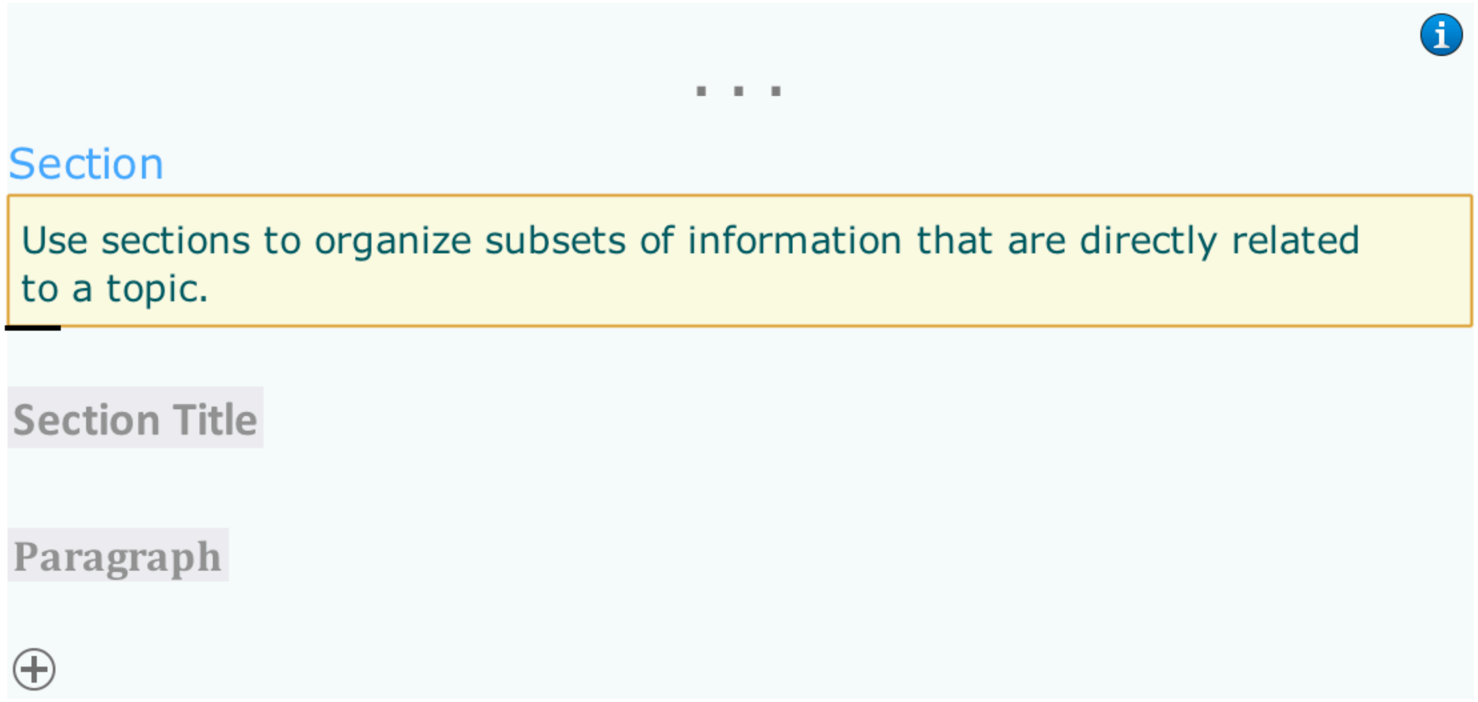
section > title {.placeholder("Section Title");}



Hints

```
section          {.hint();.markSection("Section");}
```

```
<div id="hints-section">
  <p>Use sections to organize subsets of information that are directly related to a topic.</p>
</div>
```



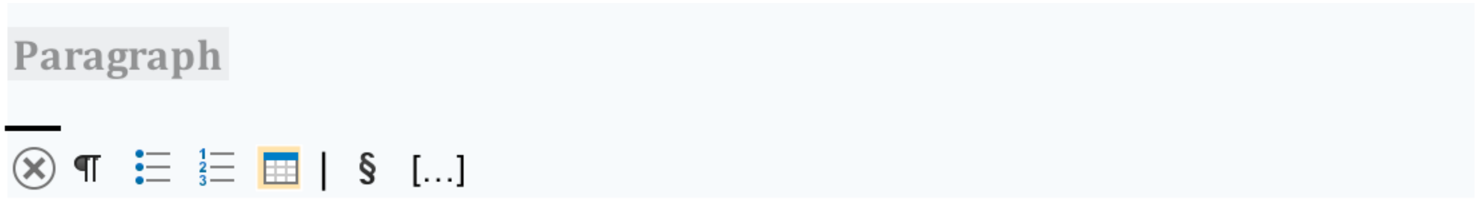
. . .

Section

Use sections to organize subsets of information that are directly related to a topic.

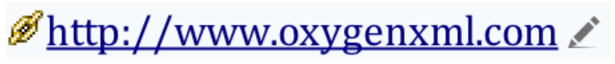
Inline actions

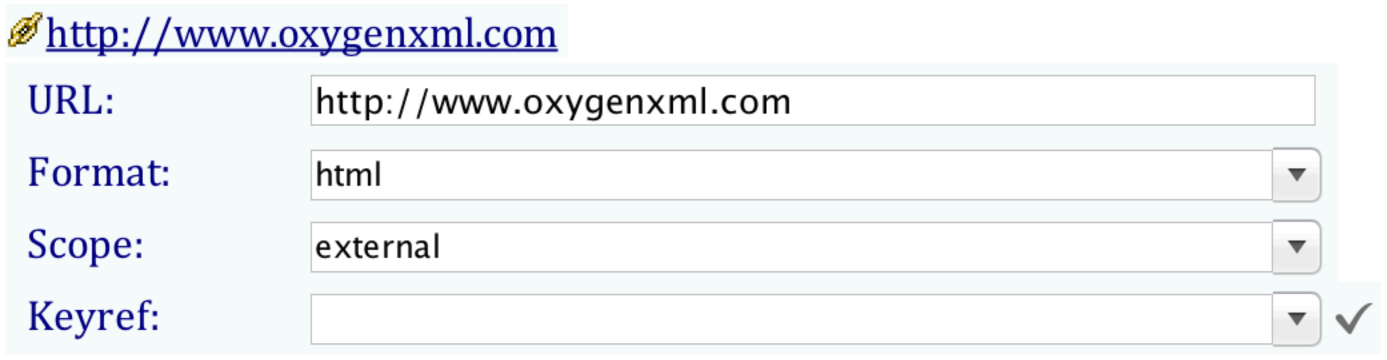
```
section {
    .actions-after({
        .action(‘paragraph');
        .action(‘unorderedList');
        .action(‘orderedList');
        .action(‘insert.table');
        .actionSeparator();
        .action(‘section');
        .actionGroup(‘pre, dl, fig, object');
    });
}
```



Inline editors

```
xref {.editableInline();}
xref:-oxy-edit:after(5) {
    .textfield("  URL:", "href");
    .combobox( "  Format:", "format");
    .combobox( "  Scope:", "scope");
    .combobox( "  Keyref:", "keyref");
}
```


http://www.oxygenxml.com


http://www.oxygenxml.com

URL:	http://www.oxygenxml.com
Format:	html
Scope:	external
Keyref:	

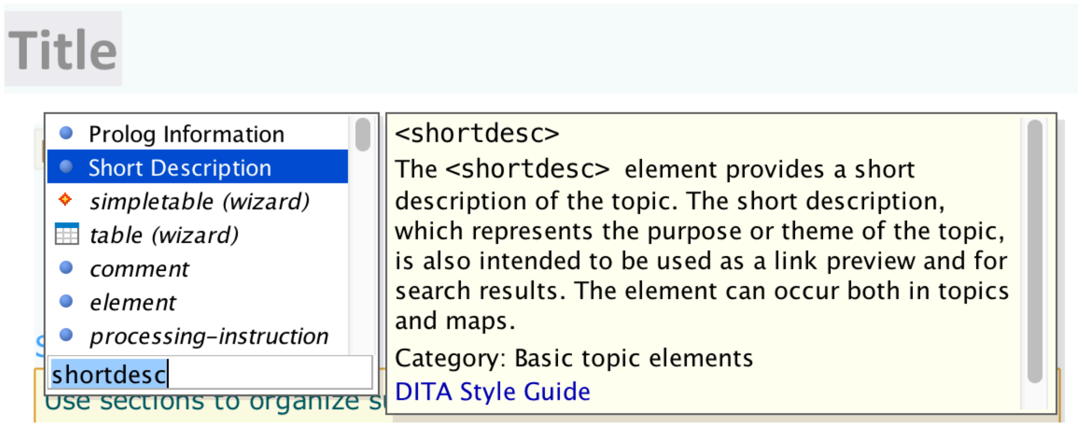
Configuration file

- Remapped element names
- Add usual content automatically to elements
- Suggest attribute values
- What elements and attributes are offered or excluded in a particular context

Remap element names

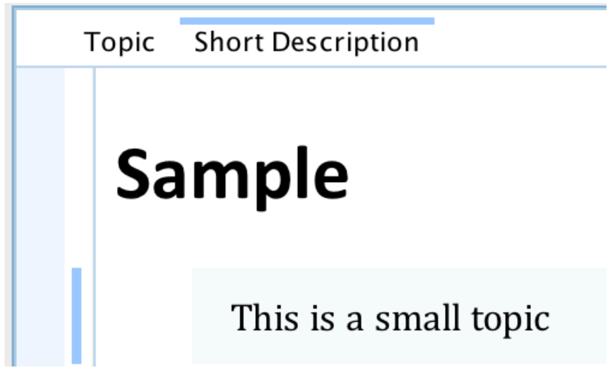
Render the “shordesc” element as “Short Description”:

```
<render
    element="shortdesc"
    as="Short Description"/>
```



Title

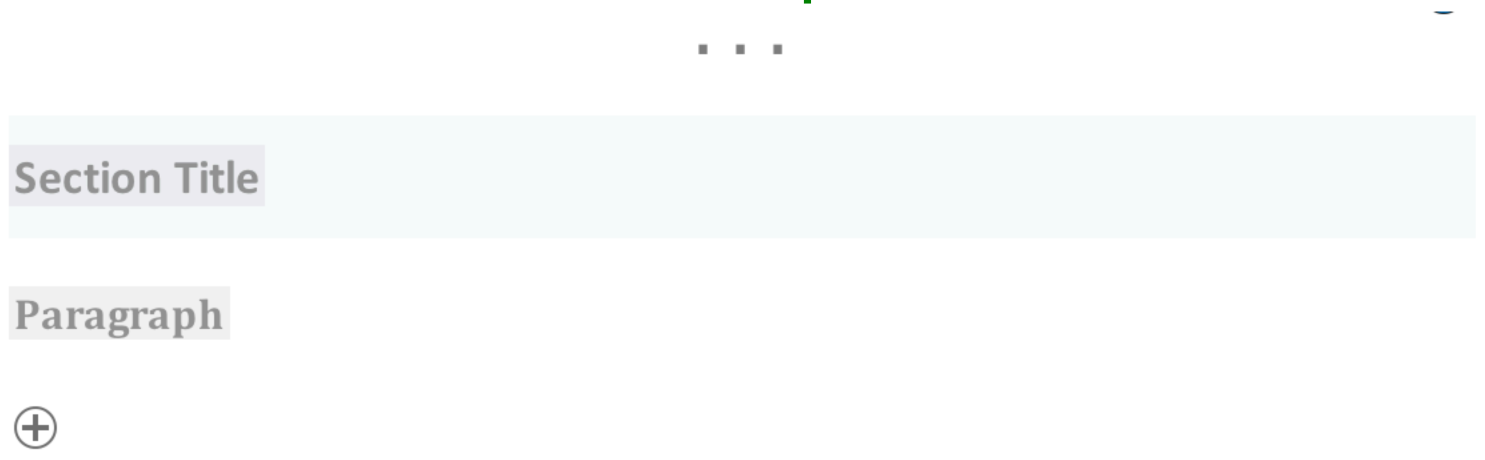
<shortdesc>
The <shortdesc> element provides a short description of the topic. The short description, which represents the purpose or theme of the topic, is also intended to be used as a link preview and for search results. The element can occur both in topics and maps.
Category: Basic topic elements
DITA Style Guide



Control inserted elements content

Automatically add a title and a paragraph in a newly inserted section:

```
<elementProposals
    path="section"
    insertElements="title p"/>
```

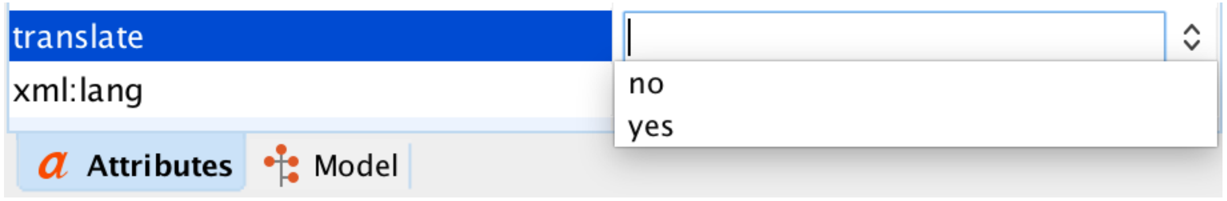

Section Title

Paragraph

Provide attribute values

Offer “yes” and “no” as options for the “translate” attribute:

```
<match attributeName="translate">
  <items action="addIfEmpty">
    <item value="yes"/>
    <item value="no"/>
  </items>
</match>
```



Take-aways

Explore what the content completion configuration file can provide, it may easily solve many common requirements

It is really useful to separate rules in two:

- styles library
- actual file that uses the available styles

→

we can develop the styles in a declarative way

Thank you

Questions?

george@oxygenxml.com

@georgebina

http://www.oxygenxml.com