1 Preview

If you associate an XML Object with an adequate XSLT document, the Preview will show the HTML resulting from this operation. To associate the XSLT file to a XML Document, add a stylesheet declaration to your XML:

```
<?xml-stylesheet type="text/xsl" href="URI of the Stylesheet"?>
```

"URL of the Stylesheet" can also be the URI of the XSLT object in TextGrid (right click the stylesheet in the navigator and select *Copy URI* to copy its URI to the clipboard). The context menu of the preview window is the same as in your operating system.

The TextGridLab bundles a version of Saxon's Home Edition, so you can use XSLT 2.0 stylesheets here.

TEXTGRIDLAB 2.3 When there is no `<?xml-stylesheet?>` declaration and you are editing a TEI document, the preview page will try to render your XML document using a default stylesheet. This will use the TEI's HTML stylesheets (by Sebastian Rahtz) by default, but you can configure it in the Preferences at *XML* / *Experimental Features*. You can use https://www.textgridlab.org/1.0/aggregator/stylesheets/preview.xsl to render an approximate simulation how your document would look like when published to textgridrep.de.

1.1 Accessing TextGrid resources from within your stylesheet

The XSLT processor that produces the HTML is able to resolve `textgrid:` URIs and thus to access TextGrid documents directly with the permissions of the user currently signed into the TextGridLab. This means that you can for example use the `document()` XPath function using TextGrid URIs, e.g., assuming `@url = 'textgrid:4711.0'`,

```
<xsl:apply-templates select="document(@url)"/>
```

will work.

TEXTGRIDLAB > 2.1.1 However, the embedded browser that will display the resulting HTML is *not* able to handle `textgrid:` URIs directly. If you would like, e.g., the browser to display the images that are referred to in your source document using something like `<tei:pb facs="textgrid:4711.0"/>`, you will first have to translate that into a http(s) URL. To help you with that, the preview page will pass the XSLT processor a parameter called `graphicsURLPattern`. This parameter's value will contain a string containing the special substring `@URI@`, and when you replace `@URI@` with the textgrid: URI of the image, you will get a https URL that the browser can use (and that already contains authentication information). So, the following XSLT fragment an handle the pb case mentioned above:

```
<xsl:param name="graphicsURLPattern"/>

<xsl:template match="tei:pb[@facs]">
  <img class="facsimile" src="{replace($graphicsURLPattern, '@URI@', @facs)}"/>
</xsl:template>
```