

[EN] 03. The Extensions of the Reader Application

The Extensions of the Reader Application

- [The Extension Mechanism of the Reader Application of the dLibra System](#)
- [The Structure and Operation of Content Presentation Extensions](#)
- [The Default Manner of Content Presentation](#)

The Extension Mechanism of the Reader Application of the dLibra System

Since version 5.0, the dLibra system has had an extension mechanism for presenting publication content. For some document formats (for example, for the DjVu format), the reader must install appropriate browser extensions. In the Reader Application of the dLibra system, there are mechanisms for presenting some file types in a user-friendly form. The mechanisms are based on software compatible with HTML and on the default solutions of Internet browsers.



W jaki sposób chciałbyś wyświetlić ten obiekt?

Pozostaw obsługę przeglądarce (pobierz plik główny)

DjVu - HTML5

Wbudowany applet DjVu

Zapamiętaj mój wybór

Zastosuj

On the basis of the type and name of the main publication file, the Reader Application determines if extensions for the manner of content presentation are available for the given publication. If they are, the application displays the list of the available extensions to the reader (see the figure above), before displaying the content. The user can use the player built into the dLibra system or leave the handling to the browser (if the plugin is already installed).

The Structure and Operation of Content Presentation Extensions

Every extension used for presenting content consists of two parts: files visible through the WWW, for example, JavaScript files or CSS sheets; those files are in the **dlibra-webapp-5.0/formats/extension-name** catalog; and files with extension configuration, for example, settings of the data formats which the given extension can display, VTL templates for generating HTML code, or a warning text; those files are in the **dlibra-webapp/WEB-INF/formats/extension-name** catalog.

Below, we describe the extension used for presenting objects in the Zoomify format. It consists of the following files:

- **settings.xml** – extension configuration,
- **message_pl.xml, message_en.xml** – text labels displayed in the warning text, and
- **content.vm, message.vm** – HTML code templates and the warning template.

Every content presentation extension must contain the five files listed above. The part visible from the WWW is optional.

The **settings.xml** file contains the most important settings related to the functioning of the extension, such as:

- **handled.mime.types** – a list of the MIME types supported by the extension, separated by commas; in the case of the discussed extension, there is only one text/xml type;
- **handles.secured** – that parameter can have the “true” or “false” value; if the given extension can protect the publication content against copying, then the value of that parameter can be set to “true”; in the case of the Zoomify extension, the value of that parameter is “false”;
- **handles.normal** – that parameter can have the “true” or “false” value; If the given publication supports presenting normal (unprotected) publications, that parameter has the “true” value (that is the case for the discussed extension); and
- **handled.fileNames** – that parameter is optional; the user should enter the admissible names for the main publication file here, separated with commas. In a publication in the Zoomify format, the main file is a simple XLM document. In order to be able to distinguish it from other file formats which also have an XML document as the main file, a requirement for consistency of the name of the main file can be introduced. In the discussed case, the value of that parameter is: ImageProperties.xml.

When the user has chosen an extension for presenting content, the Reader Extension generates a warning which pertains to* the format – the content of the message.vm file is rendered with the use of the text labels from the **messages_pl.xml** and **messages_en.xml** files. Next, the user goes to the content, and the Reader Application displays the rendered content.vm template. In that template, references to a) the elements of the extension for the manner of content presentation which are visible from the WWW and b) the file with the publication content. From the level of the content.vm template, two variables are available for the person creating extensions for presenting content:

1. **`\${completeContentUrl}`** – a complete link to the main publication file, and
2. **`\${pathToHandler}`** – a website address which indicates the catalog of the given extension, for example, http://dlibra.psn.pl/dlibra/formats/djvu_html5. That catalog contains the scripts of the given extension.

In the case of the presented extension, Zoomify, the page rendered while the publication content is being displayed contains the calling of the zoomifyViewer.swf script, where the arguments include the path to the main file of the publication. The box below contains the content of the content.vm template for the djvu_html5 extension:

```
<link type="text/css" rel="stylesheet" href="${pathToHandler}/Djvu_html5.css">
<script type="text/javascript" language="javascript"
    src="${pathToHandler}/djvu_html5/djvu_html5.nocache.js"></script>
<noscript>
    <div
        style="width: 22em; position: absolute; left: 50%; margin-left: -11em; color: red; background-
color: white; border: 1px solid red; padding: 4px; font-family: sans-serif">
        Your web browser must have JavaScript enabled in order for this
application to display correctly.</div>
</noscript>
<div id="djvuContainer" file="${completeContentUrl}"></div>
<script type="text/javascript">
var DJVU_CONTEXT = {
    parallelDownloads: 1,
};
</script>
```

The Default Manner of Content Presentation

There are three files (message_pl.xml, message_en.xml i settings.xml) in the **dlibra-webapp/WEB-INF/formats** catalog. That is the configuration of the default handling of the manner of content presentation. Any changes in those files may prevent the Reader Application from functioning correctly.