ITS for LO
The Internationalisation Extension for Libre Office

]init[Services for the eSociety

ILO Extension Developer Guide

Sebastian Sklarß, Matthias Kandora

TABLE OF CONTENT

Overview

This document provides a short overview of the code and class structures used inside the ILO Extension.

Managing ITS State/Elements

As LibreOffice does not allow the direct lookup of content and as it saves the respective data into tree like structures, the configured ITS data categories need to be hold in a tree like structure.

`De.init.its.api.ds.IitsTagElement` defines a DataCategory Node and `de.init.its.api.spi.IitsElementManager` provides a manager class to retain the structural information for all `IitsTagElement(s)`.

The `IitsElementManager` creates an arbitrary ID, which is used to keep the structural information persistent.

Notice: This extension relies on the LibreOffice mechanism of setting and reading bookmarks. The amount of characters available as labels for a bookmark is restricted by the LibreOffice API, the whole structure is stored and serialized in a file for further processing of a currently modified ODT document.

The relevant class for serialising a configured structure to disk is `de.init.its.util.ItsSerializerService.`

Note: Please be aware, that the current implementation has implemented a so called roundtripping (save and reload without losing information) only for the XLIFF Import and Export:
saving a document and re-open it with the LibreOffice file open and file save dialog will not reload previously configured ITS data categories.
In order to preserve the configured ITS structure in an ODT document, the user has to work with the export and import functionality. The XLIFF format represents the ITS data categories and therefore guarantees a lossless storage of this information on disk. Adding an ODT import and export including ITS markup as an ODT-ITS dialect might be a feature for further development.

Processing of Data

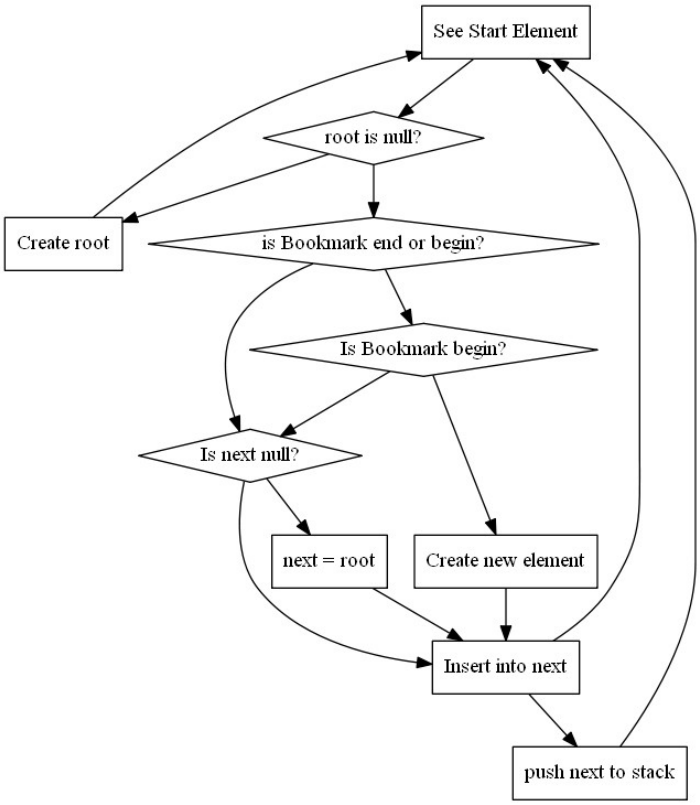
This section explains concepts on how to convert different data formats between each other.

An integral element for all operations, that will be described in more detail, is an abstract tree implementation (`de.init.its.process.support`)

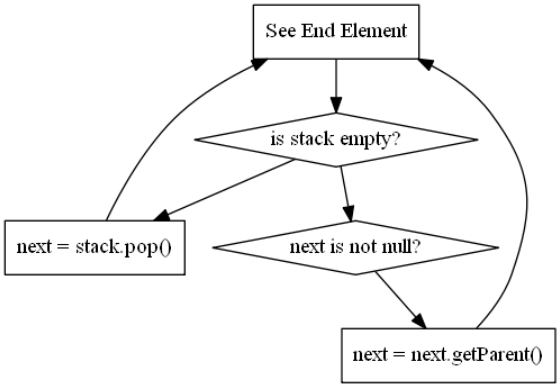
Converting ODT Bookmarks to ITS Documents

The most appropriate means to markup some content inside an ODT Document is the bookmark mechanism. A bookmark contains a range and a name.

Unfortunately, the internally used bookmark structure element does not contain content that it marks up - but encloses it as empty element: The content is surrounded by an attribute to show the beginning and the end of a present bookmark. In order to convert this 'flat' XML structure into a real hierarchical deep XML structure, the following algorithm is used for the transform:



In this schema for startElement, (root) is the (root) node, (next) is the current working node, (stack) is an internally used stack to keep track of ITS elements.



In this schema of EndElement, (node) is the current working node, (stack) is an internally used stack to keep track of ITS elements.

The main incorporating class is **`de.init.transform.re.ItsProcessor`** and it's method :**`#forward()`**

Converting ITS Documents to XLIFF

To export ITS annotated ODT documents, the XLIFFWriter class from OKAPI has been used. The main entry point for all XLIFF processing is the class **`de.init.its.process.re.XLIFFProcessor`**. We name the conversion from ITS to XLIFF a forward process, hence the name of the implementing method is **`#forward()`**.

Please refer to the OKAPI Documentation on how to convert ITS annotated content to the XLIFF Format.

Converting XLIFF to ITS Documents

Simply speaking, an XLIFF document is a file format based on the XML standard.

This section will explain shortly, what sections are of interest and how they are transformed. For a more consise overview of the XLIFF-Format please refer to the reference at the end of this document.

`<Trans-unit>`:	translation units. Will be used to indicate a certain range of text for translation
`<g>`:	general groupings. Will be used to subgroup certain segments of text
`<mrk>`:	marker blocks. Will be used to annotate specific content. The mrk blocks make use of certain attribute to denotate ITS relevant content. For example `mtype="protected"` indicates a text segment, that should not be translated, but must be retained for later processing.
`<source>`:	source block. Indicates the translation source . Inside trans-unit element
`<target>`:	target block. Indicates the the translation target. Inside trans-unit element.

(Example of XLIFF File)

```
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2"
xmlns:okp="okapi-framework:xliff-extensions"
xmlns:its="http://www.w3.org/2005/11/its"
xmlns:itsxlf="http://www.w3.org/ns/its-xliff/" its:version="2.0">
<file original="unknown" source-language="de" target-language="en" datatype="x-
undefined">
<body>
<trans-unit id="1" restype="x-text:p">
<source xml:lang="de">select the content <g id="1"><mrk
mtype="protected">marked</mrk></g> with a specific data category and open the
dialog from the toolbar (in this example 'gilla vel, aliquet nec' is
selected)</source>
<target xml:lang="en">select the content <g id="1"><mrk
mtype="protected">marked</mrk></g> with a specific data category and open the
dialog from the toolbar (in this example 'gilla vel, aliquet nec' is
selected)</target>
</trans-unit>
</body>
</file>
</xliff>
```

The conversion from XLIFF to ODT documents is implemented in the class `de.init.its.process.re.XLIFFProcessor#backward()`.

The transformation process respects the appropriate ITS to XLIFF mapping in order to convert specially configured mrk blocks into ODT elements.

Converting ITS Documents to ODT Documents

The conversion process from the deep structured ITS document to a flat XML ODT document structure is almost identical to the reversed process, except for a few things:

•The styling for ITS Data Categories needs to be restored, since the intermediary format has all styling removed

•the bookmarks are flat structured, meaning that its data has to be preserved using the ItsElementManager class and an external serialization file

Extending the source code

This extension implements only four data categories of ITS 2.0. Namely: “Translate”, “Localization Note”, “Locale Filter”, and “Terminology”. In order to extend the source to support more data categories, certain changes have to be done:

- Extend the enumeration ItsAttributes located under **`de.init.its.ds.ItsAttributes`** to support more data categories and their respective local (!) attributes.
- *Optional*: extend the validator method (validate(...)) of class de.init.its.spi.ItsAttributeManager to support more data categories and their respective attributes.
- The main entry point of the extension is **`org.openoffice.demo.IloMain`**. Look for a method called **`dispatch(...)`**. This will be called, whenever a preconfigured Button and the respective URL is being activated. (See LibreOffice Documentation on how to extend the Toolbar Interface).
- To display a dialog for a new data category, simply extend **`de.init.loext.its.dialogs.ui.ItsDialogs.ItsDialogBase`** and move the implementation to **`de.init.loext.its.dialogs.ui.ItsDialogs`**. as nested top-level class.
- Extend **`de.init.its.process.re.XLIFFReader`**, specifically the method **`#convert(...)`** to extend, or configure the currently implemented ITS to XLIFF mapping,

Styling Issues

LibreOffice Writer uses custom and automatic styles to style up sections of text. While the latter are predefined, the first are integrated into the document content.xml itself. Each style has a **generic** (!) name to be applied to certain sections of text. In order to be consistent with each roundtrip, the styles used for data categories are **fixed** (!). Since exporting to XLIFF does not imply to retain the styles, importing an XLIFF document into LibreOffice needs to reapply the styles. The converting class for ITS to ODT document is located under:
`de.init.its.process.re.ItsProcessor.BackwardHandler`, that uses the **`de.init.its.process.ItsStyleProcessor`** to apply the predefined styles for a data category.

The ItsProcessor uses a hardcoded fragment of an ODT document to insert the target content. To add new styles for the reconversion process, extend the predefined styles as XML.

While editing a text document, two classes are of importance concerning the management of bookmarks and styling:

- de.init.its.api.process.OdtStyleConfiguration

  This class provides a static and a dynamic mapping for styles used in LibreOffice. It has a simple experimental CSS Parser implemented, that can also be used to externalize the mapping between a data category and the respective styling.

- de.init.uno.text.MarkupService

  The main class to manage all bookmarks and the respective styling. One of the more interesting methods is **`findBookmark(...)`**. LibreOffice has no direct way to tell, if there is a specific bookmark at a certain location of text, but offers a rather difficult mechanism. In order to find a specific bookmark, the current selection is wrapped in a **`org.openoffice.demo.Range`** object, that contains the absolute position inside the text, using a cursor instance the algorithm iterates over all bookmarks before the current selection and adds them to a list. The matching bookmark will be returned as an **`de.init.its.ds.ItsTagElement`** .

The default styling for each data category was configured to be used in parallel (three tags on same content will be visible in the user interface as they are not mutual exclusive) as follows:

- Translate: Background-Color #aaffaa
- Locale Filter: Overline Black, Solid
- Localization Note: Underline Black, Solid
- Terminology: Background-Color: #ffaaaa

References

OASIS XLIFF Standard:	https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff
XLIFF to ITS Mapping:	http://www.w3.org/International/its/wiki/XLIFF_1.2_Mapping
W3C ITS 2.0 last call working draft:	http://www.w3.org/TR/2013/WD-its20-20130820/
]init[LibreOffice Extension:	http://extensions.libreoffice.org/?set_language=de