odt2braille Developer Guide

for version 0.0.2, 30 August 2010

by Bert Frees

Table of Contents

Introduction

1 Hacking

System Requirements

In order to hack odt2braille, you need the following software:

OpenOffice.org

The latest version of the OpenOffice.org office suite can be downloaded here.

OpenOffice.org Software Development Kit

The OpenOffice.org SDK is an add-on for OpenOffice.org. It provides the necessary tools and documentation for programming the OpenOffice.org APIs and creating own extensions. The latest version of OpenOffice.org SDK can be downloaded here. Make sure you install OpenOffice.org first. The version of OpenOffice.org SDK should be equal or lower than the version of OpenOffice.org.

NetBeans

odt2braille was developed in NetBeans, an integrated development environment. The latest version of NetBeans can be downloaded here. Make sure the Java SE pack is included.

OpenOffice.org API Plugin for NetBeans

To install the OOo API Plugin for NetBeans, launch NetBeans, go to [*Tools* > *Plugins*], then select *OpenOffice.org API Plugin* in *Available Plugins* and click on *Install*. Now relaunch NetBeans and configure the plugin in [*Tools* > *Options* > *Miscellaneous* > *OOo API Plugin*] by selecting the appropriate OpenOffice.org installation and OpenOffice.org SDK folder.

Other Downloads

- Apache Subversion: `http://subversion.apache.org/`
- Apache Ant (1.7.1 or higher): `http://ant.apache.org/`
- Javadoc: `http://java.sun.com/j2se/javadoc/downloads/index.html#findjavadoc`
- Texinfo: `http://www.gnu.org/software/texinfo/`

Source Code

The source code is available on Sourceforge and can be downloaded directly into NetBeans as follows:

- [*Team - Subversion - Checkout*]
- *Repository URL*: "https://odt2braille.svn.sourceforge.net/svnroot/odt2braille". Leave "*User*" and "*Password*" fields blank.
- *Next* >
- *Repository Folders*: "Odt2BrailleAddOn"
- Check "*Scan for NetBeans Projects after checkout*".
- *Finish*

The code can also be downloaded without NetBeans:

– `svn checkout https://odt2braille.svn.sourceforge.net/svnroot/odt2braille path/to/odt2braille/folder/`

Building

Build with NetBeans:

– Open "*Projects*" window
– Right click the Odt2BrailleAddOn project and choose menu item "*Build*".

Build with Apache ANT:

– `cd path/to/odt2braille/folder/Odt2BrailleAddOn`
– `ant`

Create OpenOffice.org extension file (.oxt) with NetBeans:

– Open "*Projects*" window, than right click "*Odt2BrailleAddOn*" and choose menu item "*Create OXT*" or "*Deploy and Run Extension in OpenOffice.org*".
– The .oxt file will be created in `path/to/odt2braille/folder/Odt2BrailleAddOn/dist`.

Create OpenOffice.org extension file (.oxt) with Apache ANT:

– `cd path/to/odt2braille/folder/Odt2BrailleAddOn`
– `ant uno-package`
– The .oxt file will be created in `path/to/odt2braille/folder/Odt2BrailleAddOn/dist`.

2 Project Overview

be.docarch.odt2braille.ooo

be.docarch.odt2braille.ooo is the OpenOffice.org extension. It makes extensive use of the OpenOffice.org UNO API. It takes care of the graphical user interface (menu's, dialogs, progress bars, etc.) and allows for loading settings from and saving settings to OpenOffice.org or the OpenOffice.org Writer document.

Documentation

be.docarch.odt2braille.ooo

View code

- http://odt2braille.svn.sourceforge.net/svnroot/odt2braille/Odt2BrailleAddOn/src/be/docar
- http://odt2braille.svn.sourceforge.net/viewvc/odt2braille/Odt2BrailleAddOn/src/be/docarc

be.docarch.odt2braille

The be.docarch.odt2braille subpackage takes care of the actual document processing. It enables the conversion of a flat xml odt file to a pef (Portable Embosser Format) file. Furthermore, this pef file can be converted to a variety of other generic braille formats, or it can be converted to an embosser-specific braille file (and optionally sent to an embosser device). The braille transcription is powered by liblouisxml, and for the pef processing, odt2braille uses the BrailleUtils library. The Java OpenDocument Library (JODL) is used for creating and cleaning the flat .odt file (see http://odt2daisy.sourceforge.net/downloads/).

Documentation

be.docarch.odt2braille

View code

- https://odt2braille.svn.sourceforge.net/svnroot/odt2braille/Odt2BrailleAddOn/src/be/doca
- http://odt2braille.svn.sourceforge.net/viewvc/odt2braille/Odt2BrailleAddOn/src/be/docarc

3 Libraries

liblouisxml

liblouisxml is the heart of the braille transcription. It is an open-source library intended to provide complete Braille transcription services for XML documents. Liblouisxml is built on top of liblouis, its translation engine. The translation is driven through text based translation tables which define the translation rules. The formatting of braille is defined in semantic mappings that define how a specific XML input tag is to be rendered in the Braille output. liblouisxml is embedded in odt2braille as an executable. It can be found in the Odt2BrailleAddOn project under `liblouis/bin`. The translation tables and configuration files are kept under `liblouis/share`.

Source Code

- `svn checkout http://liblouis.googlecode.com/svn/trunk/ path/to/liblouis/folder/`
- `svn checkout http://liblouisxml.googlecode.com/svn/trunk/ path/to/liblouisxml/folder/`

Documentation

For more information read the `liblouisxml` and `libouis` manuals.

brailleutils

See the brailleutils website.

Source Code

- `svn checkout http://brailleutils.googlecode.com/svn/trunk/braille/ path/to/brailleutils/folder/`

Building

- `cd path/to/brailleutils/folder`
- `ant`
- The files `brailleUtils.jar` and `catalog.jar` will be created in `path/to/brailleutils/folder/ant-build/output/dist`.

Documentation

- `org.daisy.braille`
  - `org.daisy.braille.embosser`
  - `org.daisy.braille.facade`
  - `org.daisy.braille.pef`
  - `org.daisy.braille.table`
  - `org.daisy.braille.tools`
  - `org.daisy.braille.ui`
- `org.daisy.factory`
- `org.daisy.paper`

- org.daisy.printing
- org.daisy.validator
- com_braillo
- com_indexbraille
- com_yourdolphin
- de_brailletec
- org_daisy
- se_tpb

4 Tutorials

Adding an embosser

To add support for an embosser, the protocol for giving print instructions to that embosser has to be known. This includes e.g.

- the configuration of the header which initiates the print job and gives general printing information,
- the way Braille pages and Braille lines are represented,
- the way each Braille character is represented (the character set),
- the footer which ends the print job, etc.

In addition, you should know the dimensions of cell spacing and line spacing, whether the embosser can print interpoint (duplex), which paper sizes are supported, etc.

Once the protocol is known, it can be implemented. What follows is a more or less general way of adapting the code in order to add an embosser. But because each embosser is different, extra adjustments may have to be made.

- In `org_pef_text.pef2text.EmbosserFactory`, expand the `EmbosserType` enumeration with a new embosser type. Add a `case` for this new embosser type to the `switch` statement in the `newEmbosser` function.
- Possibly, a new character set may have to be defined in `org_pef_text.TableFactory` as well. Edit the `TableType` enumeration and the `newTable` function.
- Finally, in `be.docarch.odt2braille.Settings`, the functions `embosserIsSupported`, `changeEmbosser`, `tableIsSupported`, `paperSizeIsSupported`, `duplexIsSupported`, `eightDotsIsSupported` and `changeEightDots` need adjustments.

Building xml2brl on Windows

Texinfo & Javadoc

Texinfo:

- `cd path/to/odt2braille/folder/Odt2BrailleAddOn/doc`
- ```
  texi2dvi --pdf
  odt2braille-user-doc.texi
  ```
- ```
  makeinfo --html
  [--no-split]
  [--no-headers]
  --output=odt2braille-user-doc.html
  odt2braille-user-doc.texi
  ```

Javadoc:

- ```
  javadoc [-private]
  [-author]
  [-version]
  [-breakiterator]
  ```
  ```

```
 -d path\to\odt2braille\folder\Odt2BrailleAddOn\doc\javadoc
 -subpackages be.docarch:org.daisy:com_braillo:com_indexbraille:com_
yourdolphin:de_brailletec:org_daisy:se_tpb
 -sourcepath path\to\odt2braille\folder\Odt2BrailleAddOn\src;path\to\brailleutils\folder
```

Adding translation tables

(volgorde, benaming, waarop letten, tabellen opslaan in ANSI, ...)

Adding hyphenation tables

5 Task list

- More flexibility - provide more settings!
  - Footnotes, endnotes, transcriber's notes...
  - Volume info and transcription info => tags gebruiken die worden vervangen: <transcriber> = instelbaar <volumenaam> = instelbaar per volume <datum> <aantal volumes> <aantal supplementen> <aantal preliminaire volumes> <beginbraillepagina> <eindbraillepagina> <beginprintpagina> <eindprintpagina>
  - 'Continued' suffix.
  - Tables
  - Textboxes
  - Images
  - Table Of Contents: - headings tot level x weergeven ?
  - Bibliography
  - Special symbols: knop om default waarde van special symbol in te stellen => getDefaultSpecialSymbol(SpecialSymbolType type, String language)
- Mac OS, Linux, ... !!! (wachten op Debian, Ubuntu, Fink)
- Support more embossers.
- Add more "Braille formatting standards" (UK, Zwitserland, België, Nederland ?)
- odt2braille_Formatting_Standards.odt: roodgekleurde regels zijn degene die nog niet ondersteund worden.
- Voor Ubuntu: OOo 3.2.1 nodig ?
- Volledige conversie (DOM + XSLT) vervangen door een reeks van kleinere XSLT's (eenvoudiger, sneller) * OK: Get languages * OK: Get styles * Delete empty paragraphs, ... (don't delete paragraphs with a <pagenum> tag) * Page numbering (=> Vincent Spiewak is rewriting JODL in XSLT 2.0 ?) * List numbering & bullets * Heading numbering * Linking of captions (=> accessibility checker ?) * OK: Main * Continued headings * OK: Special typeface + languages * OK: Split volumes
- DOM <-> STAX?
- Voor elke nieuwe release: kijken of liblouis zijn tabellen heeft geupdate.
- Voor elke nieuwe release: localisatie bijwerken
- AccessibilityChecker integration
- Settings in Tools > Options...
- Javadoc bijwerken.
- Test embossers.
- Tooltips in dialogs.
- Hidden paragraphs (or paragraphs in hidden sections) => transcriber's notes (braille-only material). TN => settings!
- Tactile graphics
- Music Braille

- If the liblouisxml process takes to long (e.g. with Chinese), OpenOffice.org might think the program is not responding.
- OpenOffice.org accessibility for screen readers (on Windows)?
- http://www.thessalonica.org.ru/en/index.html ?
- Check if the number of cells per line (and the number of lines per page) is sufficient (if too small, this might cause liblouisxml to fail).
- Java bindings?
- UTD & liblouisutdml?
- Presentations & spreadsheets.
- Commandline tool (Odt2Braille.jar).
- Uitleg van WinBraille over g0,g1,g2,... * g0 = one to one * g1 = literary one to one with capital and figure prefix * g2,g3,g4,... = contracted braille
- Grieks in Engelstalige tekst => +/- zoals in Spaans
- Automatic splitting of volumes
- Varia
  - text:list-item[@text:style-override] ?
  - Images are omitted in Braille if they are anchored to a page => alert?
  - Pagenum inside notesection ?
  - Footnotes onder tabel: ook footnotes in captions meegerekend (terwijl die al onder de caption zelf gezet zijn)
  - als Format – section – options – collect at end of section niet expliciet wordt ingesteld => notes-configuration mogelijk niet aanwezig => notesection in Braille soms op verkeerde plaats (=> uitzoeken hoe OOo wel weet waar notes moeten staan ?)
  - OOo: linking sections?
  - Volume information block: preliminary braille page numbers (p...) => p moet zonder letter indicator ?
  - Underline: dots 7 & 8 ?
  - Veranderen van typeface in midden van woord => documentatie
  - Supplementary volumes: * pagenumber prefixed by s ? * 1st pagenum should be 1.
  - Endnotes in preliminary pages: fix bug.