RelANNISModules

User's Guide

Florian Zipser <saltnpepper@lists.hu-berlin.de>
INRIA
SFB 632 Information Structure / D1 Linguistic Database
Humboldt-Universität zu Berlin
Universität Potsdam

RelANNISModules: User's Guide

by Florian Zipser, , , , and

Version ${project.version}

Table of Contents

List of Tables

Foreword

The intention of this document is first to give a guide to the user of how to use the here mentioned Pepper modules and how to utilize a mapping performed by them. Second this document shall give a closer view in the details of such a mapping in a declarative way, to give the user a chance to understand how specific data will be mapped by the presented Pepper modules.

Chapter 1. Overview

This project contains the Pepper modules listed in Table 1.1, “Pepper modules contained in this project”. A single module can be identified via its coordinates (module-name, format-name, format-version) also given in Table 1.1, “Pepper modules contained in this project”. You can use these coordinates in a Pepper workflow description file to identify the modules in a Pepper conversion process. A description of how to model a workflow description file can be found under https://korpling.german.hu-berlin.de/saltnpepper/.

relANNIS is the tab separated file format of the linguistic search and visualization tool ANNIS (see: http://www.sfb632.uni-potsdam.de/annis/). The relANNIS format is similar to the Salt meta model a graph based meta model. It also consists of nodes and edges, which are subtyped by a more specific linguistic meaning. Nevertheless, like Salt also the relANNIS format is a theory independent format.

Table 1.1. Pepper modules contained in this project

Name of Pepper module	Type of Pepper module	Format (if module is im- or exporter)
RelANNISImporter	importer	3.1
RelANNISExporter	exporter	3.1, 3.2

Chapter 2. RelANNISImporter

The RelANNISImporter make use of the relANNISModel project to map data from the relANNIS model to the Salt model. That means, the conversion process is divided into two steps:

- Importing the relANNIS data from file into relANNIS model (in main memory)
- Mapping of the relANNIS model to a Salt model

Chapter 3. RelANNISExporter

The RelANNISExporter make use of the relANNISModel project to map data from the relANNIS model to the Salt model. That means, the conversion process is divided into two steps:

- Mapping of the Salt model to a relANNIS model
- Exporting the relANNIS model to a file

Since relANNIS is available in two different versions, the relANNIS 3.1 format and the relANNIS 3.2 format, you can customize the relANNIS creation only when using the format description. That means the second case of identifying an im- or exporter. The relANNIS 3.1 format fits for all ANNIS 2.X versions, whereas the relANNIS 3.2 is importable since ANNIS 3.0.

The relANNIS format is a csv based format and uses for separating columns the tab character and for separating lines a line break character. Therefore it is necessary to not have tab characters or line breaks inside the data like annotation names, annotation values, primary data and so on. Of course, the data could be checked when exporting them. But since checking each String for occurances of these characxters is a time consuming job and a lot of data does not contain such characters, it is up to the user to switch one the that cleaning process. Therefore the two properties relANNIS.exporter.tab_escape and relANNIS.exporter.linebreak_escape exist, with which even the replacement character or character sequence can be determined.

Properties

The table Table 3.1, “properties to customize exporter behaviour” contains an overview of all usable properties to customize the behaviour of this Pepper module. The following section contains a close description to each single property and describes the resulting differences in the mapping to the Salt model.

Table 3.1. properties to customize exporter behaviour

Name of property	Type of property	optional/ mandatory	default value
relANNIS.exporter.tab_escape	String	optional	--
relANNIS.exporter.linebreak_escape	String	optional	--
relANNIS.exporter.backslash_escape	String	optional	--

relANNIS.exporter.tab_escape

With this property tab characters ('\t') will be replaces by the given character or character sequence. For instance imagine the following text:

```
This\tis\ta\ttext
```

Normally, in most editors, a tab is shown as a longer whitespace and not as '\t'. The use of the property:

```
relANNIS.exporter.tab_escape=BLANK
```

will result in the following text:

```
ThisBLANKisBLANKaBLANKtext
```

Of course, this replacement does not make much sense and normally you would replace a tab with one or a sequence of blanks. This is possible when using the blank key instead of the 'BLANK' sequence, but hard to read in this example.

relANNIS.exporter.linebreak_escape

With this property linebreak characters (like '\n' and '\r') will be replaces by the given character or character sequence. For instance imagine the following text:

```
This\nis\na\ntext
```

Normally, in most editors, a linebreak is not shown and not as '\t'. The use of the property:

```
relANNIS.exporter.linebreak_escape=BLANK
```

will result in the following text:

```
ThisBLANKisBLANKaBLANKtext
```

Of course, this replacement does not make much sense and normally you would replace a linebreak with one or a sequence of blanks. This is possible when using the blank key instead of the 'BLANK' sequence, but hard to read in this example.

relANNIS.exporter.backslash_escape

With this property bakcslash character ('\') will be replaces by the given character or character sequence. For instance imagine the following text:

```
This\is\a\text
```

. The use of the property:

```
relANNIS.exporter.backslash_escape=BLANK
```

will result in the following text:

```
ThisBLANKisBLANKaBLANKtext
```

.