

RSTModules

- the manual -

Foreword

The OSGi-bundle RSTModules contains two Pepper-modules, the RSTImporter to map a RST-model to a Salt-model and the RSTExporter to map a Salt-model to a RST-model.

RSTImporter

The `RSTImporter` can be used in two main flavors, first as a simple mapper, which maps the in RST existing model elements to model elements in Salt and second as an enhanced mapper, which uses a tokenizer (equal result as the `TreeTagger` tokenizer) to enhance the rst data with an additional token layer.

In general for all texts text inside the segments one `STextualDS` will be instantiated and its text will be set to the concatenated text of all segments in order of the segments. A segment as well as a group will be mapped to a `SStructure` object. Each relation of type (type or name attribute) span will be mapped to a `SDominanceRelation` in Salt. For each non-span relation in the path to a leaf of a node having a span relation, an additional `SDominanceRelation` will be created. For example imagine the following:

$$\begin{array}{l} a \text{ --> } [span] \ b \\ b \text{ --> } [non-span] \ c \end{array}$$

In this case an additional dominance relation between *a* and *c* will be created. This is done, to create a tree-like structure in the sense of a full syntactic tree.

The behavior of the `RSTImporter` can be adjusted by setting some parameters inside a property file. This property file can be included in the pepper workflow description file as described in the `QuickUserGuide` of the Pepper documentation. The using of a parameter file is optional for this module.

Here we show a property file, containing all possible parameters which can be used to adjust the `RSTImporters` behavior. A short description of each parameter is given with the comments. A closer explanation is given beyond.

```
# if parameter is set to yes the text being included in a segment will be
# tokenized
rstImporter.tokenize=(yes|no)
# FOLDER of a not default abbreviation file
rstImporter.abbreviationFolder= FOLDER
# determines the language of primary text, if no one is set, the language will
be detected automatically per primary text
rstImporter.language= (IT|FR|DE|EN)
```

Example 1: property file including all possible parameters to adjust the behavior of the RSTImporter

rstImporter.tokenize

This parameter is an optional parameter and can be set to “yes” or “no”. If it is set to “yes”, the text being included in a segment will be tokenized. The tokens will be mapped to `SToken`-objects in `Salt` and attached to the `SDocumentGraph`-object. Further, an `STextualRelation` between a token and the text will be created and a dominance relation between the token and the segment. The default configuration of this parameter is true, if non tokenization is required, this parameter must explicitly set to false.

rstImporter.abbreviationFolder

This parameter is an optional parameter and can be set to a folder containing user-defined abbreviations for the tokenizer mechanism. If the parameter “`rstImporter.tokenize`” is not set to “yes”, this parameter will be ignored.

rstImporter.language

This parameter is optional and determines the language to be used for tokenizing of primary text. If no language is given, it will be detected automatically.

RSTExporter

- more to come -