Regular Expressions Patterns for TokXMLFile

by Bill Rich

Regular Expressions Patterns for TokXMLFile.

Table of contents

1. Overview

To control what is extracted from XML type files the `TokXMLFile` class uses a set of regular expressions. The regular expressions are presented in a properties file specified when the `Tok` class is called. This in turn is specified in the `L10NProcess.xml` file or through the use of a command line parameter. The easiest way to provide the appropriate properties file name is to use the `project.properties` file. If no `XMLPROPFILE` property is set then the default `xml.properties` file will be used.

The `project.properties` file and the properties file that contains the regular expressions (default is `xml.properties`) must be placed in the project directory.

Some design considerations used during development of the `TokXMLFile` class are:

- the patterns can vary from one project to another
- the project patterns properties file must be specified by the user at run time
- the project patterns properties file will be controlled by the Ant process files
- if no project patterns properties file is supplied the default pattern properties file will be used
- there is no affect on the `Detok` process

Some assumptions used during design and implementation of `TokXMLFile` are:

- all text is properly coded XML
- the extracted item is a standard Regular Expression pattern contained between (and)
- up to nine items can be extracted in a single pattern
- patterns can be combined if needed
- patterns must be specified in order from most specific to most general
- all patterns will be matched
- extracted strings that contain only whitespace or a token will be ignored and all other extracted strings will be replaced by a

token
- the regular expression syntax is AWK syntax

Some example regular expressions are:

Select the value of attribute a2 when the value of attribute a1 is x and the element is e.
```
<e[^>]*[[:space:]]+a1="x".*[[:space:]]+a2="([^"]+)"[^>]*>
```
Select the value of attribute a when the element is e.
```
<e[^>]*[[:space:]]+a="([^"]+)"[^>]*>
```
Select the text for the element when the element is e.
```
<e[^>]*>([^<]+)</e>
```
Select the value of attribute a.
```
<[^>]+[[:space:]]+a="([^"]+)"[^>]*>
```
Select the text for any element. (This will be the default pattern supplied with the tool kit.)
```
<[^>]+>([^<]+)</[^>]+>
```

Return to: Top of page