

SegmentStatistics

Category	Import
Purpose	Creates several segment metrics directly after import
Bootstrap Class	editor_Plugins_SegmentStatistics_Bootstrap
Type	Core plug-in (delivered with translate5 core)

Description

Counts several Segment Statistics (word, char and segment counts), currently directly after Import.

On each import and export information about terminology is gathered. These are: a terminology counter and the information if a term is found in source and target column.

This data is stored as XML files in the task data directory, this would be per default:

```
/data/editorImportedTasks/[TASKGUID]
```

In addition to the XML files the statistics are summarized in Microsoft Excel spreadsheet files (.XLSX). These files are also generated on import and export.

Optionally the statistics data can be filtered. Currently exists only a filter for the segments meta flag "transitLockedForRefMat". This flag is set by the [Transit Plugin](#).

If a filter is configured, one statistic file set is created unfiltered and one filtered.

Configuration

Config Name	Default Value	Description
runtimeOptions.plugins.SegmentStatistics.xlsTemplateExport	modules/editor/Plugins/SegmentStatistics/templates/export-template.xlsx	Path to the XLSX export template. Path can be absolute or relative to the application directory.
runtimeOptions.plugins.SegmentStatistics.xlsTemplateImport	modules/editor/Plugins/SegmentStatistics/templates/import-template.xlsx	Path to the XLSX import template. Path can be absolute or relative to the application directory.
runtimeOptions.plugins.SegmentStatistics.metaToIgnore.transitLockedForRefMat	0	0 or 1; decides, if segments with metadata "transitLockedForRefMat" will be ignored by this plugin.
runtimeOptions.plugins.SegmentStatistics.disableFileWorksheetCount	15	If there are more files in the task as configured here, the worksheets per file are disabled, and only the summary worksheet is shown.
runtimeOptions.worker.editor_Plugins_SegmentStatistics_Worker.maxParallelWorkers	3	Maximum of parallel running statistic workers.
runtimeOptions.plugins.SegmentStatistics.createFilteredOnly	0	Boolean, if true creates only the filtered statistics instead both.

Statistic files created by this plugin

All files are created in the task data directory under "data/editorImportedTasks/TASKGUID/":

files created after import	
segmentstatistics-import.xml	statistics in XML format, unfiltered
segmentstatistics-import-filtered.xml	statistics in XML format, filtered, only created if filters defined
segmentstatistics-import.xlsx	statistics in XLSX format, unfiltered
segmentstatistics-import-filtered.xlsx	statistics in XLSX format, filtered, only created if filters defined

files created after each export

segmentstatistics-export-2015-08-13-10-22.xml	statistics in XML format, unfiltered, timestamp of creation
segmentstatistics-export-2015-08-13-10-22-filtered.xml	statistics in XML format, filtered, only created if filters defined, timestamp of creation
segmentstatistics-export-2015-08-13-10-22.xlsx	statistics in XLSX format, unfiltered, timestamp of creation
segmentstatistics-export-2015-08-13-10-22-filtered.xlsx	statistics in XLSX format, filtered, only created if filters defined, timestamp of creation

XLSX Templates

In the plugin directory exists a "templates" directory, which contains templates for the XLSX files to be created. Two files are provided by translate5: "import-template.xlsx" and "export-template.xlsx".

Changes in these files are overwritten on translate5 updates. If you want to edit the templates, they have to be copied and the new filenames have to be configured in the above described configuration.

The Templates must contain 3 sheets as shown in the delivered files. The first and second sheet contain a simple template mechanism in row 3. The values of this row will be replaced for each file in the task.

There are several template variables starting with "STAT.". These fields are getting replaced by the data from translate5. For all available variables see the existing .XLSX template files.

Specialties

- For existing projects with statistics before the wordCount field was added, the wordCount is initialized with "-1". Therefore negative sums are possible.
- The state specific counter can contain "0" or they can be completely empty. Completely empty means, there are no segments with this state at all. 0 means, that there are segments with the given state, but they contain no terms with the desired term state (trans[Not]Found).

Description of XLSX

See last Worksheet in the XLSX file.

Description of XML

The following commented example file, describes the content of an XML statistic file.

Although the target specific counts are listed in the XML, they are currently always 0 since term[Not]Found information is not provided anymore in target columns.

Example statistic XML file

```
<?xml version="1.0"?>
<statistics>
  <taskGuid>{792b3c27-0f48-4eaf-aaa3-dbdffd4da62b}</taskGuid>          <!-- The unique guid of the task -->
  <taskName>testpaketneu  TRANSLATE-485</taskName>                    <!-- The textual task name -->
>

<filtered>
<!-- A list of set filters, if no filter is set the filtered tag is empty, the filters influences all counters
-->
  <filter>transitLockedForRefMat</filter>
</filtered>
  <segmentCount>234</segmentCount>                                     <!-- The
overall segment count of the task -->
  <segmentCountEditable>22</segmentCountEditable>                     <!-- How many segments are
editable in the task -->

<import>
<!-- The import section contains the stats directly after import -->

<files>
<!-- In the files section for each file in task one section is created -->
  <file>
    <fileName>MyNiceFile.ENG.transit</fileName>
```

```

<fileId>3539</fileId>
<fields>

<field>                                                                <!--
foreach field one stat block is created, these are mainly source and target -->
    <fieldName>source</fieldName>
    <charFoundCount>1316</charFoundCount>                                <!-- counts all chars in this
file segments with at least one blue (found) term -->
    <charNotFoundCount>801</charNotFoundCount>                            <!-- counts all chars in this
file segments with at least one red (not found) term -->
    <wordFoundCount>195</wordFoundCount>                                    <!-- counts all words in this
file segments with at least one blue (found) term -->
    <wordNotFoundCount>117</wordNotFoundCount>                            <!-- counts all words in this
file segments with at least one red (not found) term -->
    <termFoundCount>31</termFoundCount>                                    <!-- counts all blue
(found) terms in this file -->
    <termNotFoundCount>11</termNotFoundCount>                            <!-- counts all red (not found)
terms in this file -->

<!-- segments with red and blue terms are counted twice! -->
    <segmentsPerFile>117</segmentsPerFile>                                <!-- counts all segments in
this file -->
    <segmentsPerFileFound>24</segmentsPerFileFound>                        <!-- counts all segments in this
file with blue (found) terms -->
    <segmentsPerFileNotFound>11</segmentsPerFileNotFound>                <!-- counts all segments in this file
with red (not found) terms -->
    <targetCharFoundCount>1301</targetCharFoundCount>                    <!-- counts all chars in the target field where
the source contains blue (found) terms -->
    <targetCharNotFoundCount>839</targetCharNotFoundCount>                <!-- counts all chars in the target
field where the source contains red (not found) terms -->
    <targetSegmentsPerFileFound>24</targetSegmentsPerFileFound>          <!-- same value as
segmentsPerFileFound -->
    <targetSegmentsPerFileNotFound>11</targetSegmentsPerFileNotFound>    <!-- same value as
segmentsPerFileNotFound -->
</field>

<field>                                                                <!--
same statistics as described above, only for the target field -->
    <fieldName>target</fieldName>
    <charFoundCount>0</charFoundCount>                                    <!-- since target fields
does not contain trans[Not]Found info anymore, the values are always 0 -->
    <charNotFoundCount>0</charNotFoundCount>
    <wordFoundCount>0</wordFoundCount>
    <wordNotFoundCount>0</wordNotFoundCount>
    <termFoundCount>0</termFoundCount>
    <termNotFoundCount>0</termNotFoundCount>
    <segmentsPerFile>117</segmentsPerFile>
    <segmentsPerFileFound>0</segmentsPerFileFound>
    <segmentsPerFileNotFound>0</segmentsPerFileNotFound>
</field>
</fields>
</file>
<file>
    <!-- [...] here would be the next file -->
</file>
</files>

<fields>
<!-- The fields section contains the sum of the statistic values over all files -->
<field>
    <fieldName>source</fieldName>
    <taskCharFoundCount>2632</taskCharFoundCount>
    <taskCharNotFoundCount>1602</taskCharNotFoundCount>
    <taskWordFoundCount>390</taskWordFoundCount>
    <taskWordNotFoundCount>234</taskWordNotFoundCount>
    <taskTermFoundCount>62</taskTermFoundCount>
    <taskTermNotFoundCount>22</taskTermNotFoundCount>
    <taskTargetCharFoundCount>2602</taskTargetCharFoundCount>
    <taskTargetCharNotFoundCount>1678</taskTargetCharNotFoundCount>
    <taskTargetWordFoundCount>430</taskTargetWordFoundCount>
    <taskTargetWordNotFoundCount>294</taskTargetWordNotFoundCount>

```

```

        <taskTargetSegmentsPerFileFound>48</taskTargetSegmentsPerFileFound>
        <taskTargetSegmentsPerFileNotFound>22</taskTargetSegmentsPerFileNotFound>
    </field>
    <field>
        <fieldName>target</fieldName>
        <taskCharFoundCount>0</taskCharFoundCount>
        <taskCharNotFoundCount>0</taskCharNotFoundCount>
        <taskWordFoundCount>0</taskWordFoundCount>
        <taskWordNotFoundCount>0</taskWordNotFoundCount>
        <taskTermFoundCount>0</taskTermFoundCount>
        <taskTermNotFoundCount>0</taskTermNotFoundCount>
    </field>
</fields>
</import>

<export>
<!-- The export section contains the stats after triggered export -->
    <!-- [...] here would be the statistics at export -->
</export>
</statistics>

```

Debugging

Adding the following line to your installation.ini enables debugging output for SegmentStatistics Plugin.

```
runtimeOptions.debug.plugin.SegmentStatistics = 1
```

Enabled debugging for segment statistics does:

- create segmentstatistics-export files without a timestamp in filename, this makes checking file content easier.
- returns XML files formatted (intendation)
- writes in addition a CSV file with the XLS content
- writes to the error log when writing XLS is finished