What is Gentry?

Gentry is the generic file parser technology that KantanMT uses to parse XML based file formats. Gentry is script driven and uses Rule files (*.rul) to instruct your KantanMT engine what to translate. These rule files are easy to create by using a simple text editor and it's not uncommon for a complete XML parser to be built in a matter of minutes.

While GENTRY is at its core a parsing technology, it is also used for:

- Data Tokenisation: A good word/morpheme text tokeniser is fundamental for precision word and phrase detection Data tokenisation is nontrivial for scarce resourced languages such as Urdu, as there is inconsistent use of space between words, whereas it's pretty simple to detect word boundaries in languages such as English.

- Data De-tokenisation: This is the opposite of data tokenisation and is focused on the re-assembly of words/phrases post-translation. GENTRY handles this process for all languages including single-byte, double-byte and bi-di language combinations.

- Pre-processor: GENTRY provides a library of service functions that can be used to pre-process training data to correct/adjust it prior its inclusion as training data.

  (Checkout http://www.kantanmt.com/help_preprocessor.php for more details on this process.)

- PEX: GENTRY provides the service functionality for PEX (Post-Editing Automation). PEX allows for post-translation adjustments to KantanMT outputs to correct/adjust any potential flaws of repetitive translation errors.

- Rule-Based Parser: This is the most powerful feature of GENTRY as it allows clients to enhance existing and known file parsers and to develop their own parsers to facilitate the translation of proprietary file formats on the KantanMT platform.

Rule-Based Parsing...

Rule-based parsing is a technique that GENTRY uses to help the KantanMT community rapidly develop custom parsers for their proprietary file formats. It is based on a simple yet powerful concept of **rules**. These rules define how GENTRY will parse files and identify which elements within the files are to be translated. These rule are stored in an XML file that is well structured, easily mastered and simple to create in any basic text editor.

For example, let's see how quickly we can write a rule to translate the following proprietary XML file on the KantanMT.com platform:-

```
view plain  copy to clipboard  print  ?
01. <? xml version="1.0" encoding="UTF-8" ?>
02. <chapters>
03. <chapter id="1001" source="printed">
04. <title>Introduction</title>
05.     <para>Acme Technologies provide intelligence, visual analysis, and for
06. </chapter>
07.
08. <chapter id="1002" source="online">
09.     <para>We protect servers by allocating them to a suitable virtual recc
    </para>
10. </chapter>
11. </chapters>
```

It's fairly quick to see that the text contained within the **<title>** and **<para>** elements are to be translated. These elements are referred to as **Roots** and are easily defined within a GENTRY rule file as follows:-

```
view plain  copy to clipboard  print  ?
01. <roots>
02.     <root>para</root>
03.     <root>title</root>
04. </roots>
```

That's it! We've created a GENTRY parser by simply defining two **Roots**.

What does a GENTRY .RUL file look like?

The GENTRY rule file is a well-structured, easy to read XML file. It has the following format:-

```
view plain  copy to clipboard  print  ?
01. <?xml version="1.0"?>
02. <rules>
03.      <!---Define Root element here ---->
04.     <roots>
05.         <root>para</root>
06.         <root>title</root>
07.     </roots>
08.
09.     <!---Define extraction/insertion operations here  ---->
10.     <regex>
11.         <gextractrule>(.*)</gextractrule>
12.         <gextractoutputrule>$1</gextractoutputrule>
13.         <ginsertrule>$1</ginsertrule>
14.     </regex>
15. </rules>
```

There are two major sections in a GENTRY rule file:-

- Root Definitions: This is the section where all the **Roots** are defined. A **Root** defines what elements of a file are to be translated.

- REGEX Definitions: These section defines a series of regular expressions that are used to parse each **Root** element.

Each of these sections are then wrapped into a **<rules>** element to ensure that the GENTRY rule file conforms to the expected XML well-formedness requirement.

To make sure that these files can contain valid DBCS and accented characters, the GENTRY rule file should also be saved as UTF8.

What are REGEX definitions for?

The REGEX definitions are applied to each **root** defined in a file. They are used to instruct GENRTY on how to parse each **root** element and use Regular Expressions to define this parsing behaviour.

A GENTRY rule can have an unlimited number of REGEX definitions which can be used to build complex parsers quickly and efficiently.

- <**gextractrule**>: A **root** element is selected for translation only if it matches this Regex expression.
- <**gextractoutputrule**>:This tells GENTRY how to format the content of a matched **root** before it is translated by a KantanMT engine.
- <**ginsertrule**>: This tells GENTRY how to re-insert the **root** element back into the original file post-translation.

Regex definitions are very powerful and can be used to develop smart parsers.

For example, suppose you wanted to translate **root** elements that are preceded by a four-digit identification code, while ignoring everything else in a file - use the expression:-

<gextractrule>(^[\d]{1,4})(.*)**</gextractrule>**

Only **root** elements that matched this expression would be selected for translation.

Get in touch today to set up a private demo...

T. +353-1-700-7874 | E. info@kantanmt.com | www.KantanMT.com