Connecting client infrastructure with Yamagata Europe machine translation using JSON-based data exchange

Jourik Ciesielski
Yamagata Europe
Zwijnaardsesteenweg 316 B
9000 Ghent, Belgium
jourik.ciesielski@yamagata-europe.com

Heidi Van Hiel
Yamagata Europe
Zwijnaardsesteenweg 316 B
9000 Ghent, Belgium
heidi.van.hiel@yamagata-europe.com

Abstract

This document describes how Yamagata Europe enables organizations to connect seamlessly to its machine translation and translation management system infrastructure using a JSON-based (JavaScript Object Notation) data exchange mechanism.

1 JSON protocol

Yamagata Europe's data exchange service is based on the JSON interchange format to transfer data between translation buyers on the one hand and Yamagata Europe on the other. The purpose of the service is to provide an easy-to-implement and extensible alternative for other translation data exchange standards such as COTI (Common Translation Interface) and TIPP (Translation Interoperability Protocol Package). Translatable objects are contained in a ZIP package together with a package description file (hereinafter referred to as *manifest*). The ZIP packages are compressed programmatically at the client's end, possibly with support of Yamagata Europe, and subsequently transferred to Yamagata Europe via for example file transfer protocol (FTP) or cloud storage utilities like Amazon S3. Package transfers are monitored and confirmed through a proprietary API (application programming interface). Once at the Yamagata Europe premises, packages are automatically decompressed, analyzed and, in accordance with the prescriptions in the manifest, the required workflow steps (machine translation, machine translation and post-editing, etc.) are organized and executed. When the workflow is completed, the service creates a response package and sends it back to the native repository at the client's end.

2 Data flow

The manifest in a data package is a JSON file that contains information about:

- The required service (e.g., machine translation only, machine translation and post-editing with or without desktop publishing work, etc.).
- The source and target language(s).
- The source file(s) for translation (file name including a unique identifier).
- Extra information regarding the source file format.

Example of a JSON manifest:

```
{
  "service": "Machine Translation",
  "id": "61d2d6260d3da511cf63c328",
  "source": "en",
  "target": [
    "fr"
  ],
  "files": [
    {
      "id": "61d2d6790d3da511cf63c329",
      "fileName": "NewEmployee.docx",
      "storyLine": false
    }
  ]
}
```

Figure 1: JSON manifest file

The decompressing of a package entails several checks to verify whether the data meets the conditions to be processed correctly. If zero or more than one JSON files are found, the process will stop and the package will be moved to an

"Error" folder, while an error message will be sent to the client through the provided API. The same thing will happen if an unknown service type or an unsupported target language is specified. Furthermore, the source files specified in the manifest must be identical to the files in the payload. The entire payload travels in a single folder without subfolder structures for the translatable objects.

If a package is valid, a translation project will be created in Yamagata Europe's translation management system using a dedicated project template that corresponds to the prescriptions in the manifest. If machine translation is specified as the desired service, the source files will be machine-translated using Yamagata Europe's internal machine translation service, which includes an automated pre-editing component (optimization of source content) and an automated post-editing component (automatic correction of recurring mistakes such as formatting or terminology issues). The last step of the flow consists of creating a response package and notifying the client about the project delivery through the provided API. The response package is a ZIP file containing the original JSON manifest and one or more language folders containing the translated objects.

3 Pitfalls

An important pitfall is the insertion of inline XML-style tag mistakes during the machine translation process. Inline tag issues might prevent the translation management system from generating translated objects, which will break the automation. For that reason, Yamagata Europe has developed a smart tag handling algorithm that remembers the content of opening as well as (self-)closing XML tags, converts the tags into numbered placeholders, protects them during the machine translation process and restores them in their original format and position after the process.

A second pitfall is related to the supported file types. The current setup foresees support for the following file formats:

- Office Open XML (OOXML) document, `docx`
- Articulate Storyline OOXML document, `docx`
- OOXML presentation, `.pptx`
- OOXML spreadsheet, `.xlsx`
- XML (flavor to be determined with customer)
- PDF
- Support for other file formats, including industry standards such as XLIFF (XML Localisation Interchange File Format), can be added upon request.

Articulate Storyline is an e-learning authoring tool that includes a translation export module to `.docx`. Certain metadata fragments in Storyline exports are not supposed to be modified during translation. To distinguish regular `.docx` files from Storyline `.docx` files, an additional parameter is added to the JSON manifest. This parameter triggers an additional script in the translation management system to protect metadata in the case of Storyline exports.

Portable Document Format (PDF) files are generally challenging for translation and might prevent the automation from executing successfully. Password-protected and scanned PDF files in particular will result in an empty translated object. Only PDF files that can be saved as `.docx` will go correctly through the process.

4 Assets

The JSON-based data exchange mechanism allows organizations to integrate their content repositories and self-service portals with Yamagata Europe's internally developed machine translation infrastructure. The flow automates repetitive and time-consuming tasks at every stage of the translation process — from data transfer to project creation, machine translation and delivery — and therefore enables companies to process more content at a faster pace and a high-quality standard.