XLIFF 2 Extraction and Merging Best Practice, Version 1.0

Edited by David Filip and Ján Husarčík
Rodolfo M. Raya
Andreas Galambos

TAPICC T1/WG3

Additional artifacts

This prose specification is one component of a Work Product that also includes:

- Extraction and merging examples from https://github.com/GALAglobal/TAPICC/tree/master/extraction_examples

Related work

This note provides informative best practice for XLIFF 2 Specifications:

- XLIFF Version 2.1 [[XLIFF-2.1]]
- XLIFF Version 2.0 [[XLIFF-2.0]]
- ISO 21720:2017 [[ISO XLIFF]]

Status

This Informational Best Practice was last revised by TAPICC T1/WG3 or the TAPICC Steering Committee on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Contributions to this deliverable or subsequent versions of this deliverable can be made via the GALA TAPICC GitHub Repository [https://github.com/GALAglobal/TAPICC] subject to signing the TAPICC Legal Agreement [https://www.gala-global.org/tapicc-legal-agreement].

Citation format

When referencing this specification the following citation format should be used:

[XLIFF-EM-BP]

XLIFF 2 Extraction and Merging Best Practice, Version 1.0 Edited by David Filip and Ján Husarčík. 24 January 2018. Working Draft 01. https://galaglobal.github.io/TAPICC/T1/WG3/wd01/XLIFF-EM-BP-V1.0-wd01.html. Latest version: N/A.html.

Notices

The Translation API Class and Cases (TAPICC) initiative is a collaborative, community-driven, open-source project to advance API standards in the localization industry. The overall

24 January 2018

Abstract

This Informational Best Practice specification targets designers of XLIFF Extracting and Merging Tools for content owners. It gathers common problems that are prone to appear when *Extracting XLIFF Documents* from HTML, generic XML, or MarkDown. This specification shows why some *Extraction* approaches will cause issues during an *XLIFF Roundtrip*. This best practice guidance provides better thought through alternatives and shows how to use many of advanced XLIFF features for lossless Localization roundtrip of HTML and XML based content.

Table of Contents

Terminology and Concepts

Context hints
: XLIFF attributes on structural or inline elements providing additional contexts, such as `disp [http://docs.oasis-open.org/xliff/xliff-core/v2.1/xliff-core-v2.1.html#disp]` or `equiv [http://docs.oasis-open.org/xliff/xliff-core/v2.1/xliff-core-v2.1.html#equiv]`.

Inline codes

marker

Introduction

This specification targets designers of XLIFF Extracting and Merging Tools for content owners. XLIFF Roundtrip designers of all kinds will benefit, no matter if they design their XLIFF Extractor/Merger for corporate or blog use.

Extraction and merging behavior is out of the normative scope of OASIS XLIFF Specifications. Although those specifications do provide some guidance for Extractor and Merger Agents, XLIFF TC did not attempt to prescribe how exactly to use XLIFF to represent native content. This is mostly because XLIFF is a native format agnostic Localization interchange Format.

This Informational Best Practice targets designers of XLIFF *Extracting* and *Merging* Tools for content owners. *XLIFF Roundtrip* designers of all kinds will benefit, no matter if they design their *XLIFF Extractor/Merger* for corporate or blog use.

Extraction and *Merging* behavior is out of the normative scope of OASIS XLIFF Specifications. Although those specifications do provide some guidance for *Extractor* and *Merger Agents*, XLIFF TC did not attempt to prescribe how exactly to use XLIFF to represent native content. This is mostly because XLIFF is a native format agnostic Localization Interchange Format.

This specification gathers common problems that are prone to appear when Extracting XLIFF Documents from HTML, generic XML, or MarkDown. This specification shows why some *Extraction* approaches will cause issues during an *XLIFF Roundtrip*, issues often so severe that *Merging* back of target content will not be possible without costly postprocessing or could fail utterly. This best practice guidance provides better thought through alternatives and shows how to use many of advanced XLIFF features for lossless Localization roundtrip of HTML and XML based content. Most of the times there are no ultimate prescribed solutions, rather possible design goals are described and best methods how to achieve them proposed.

Specification

Inline Codes

Representing Spanning Codes

Spanning codes in the original format are created by opening code, content and closing code. In HTML that can be `<bold>text</bold>`, in RTF `\b text \b0`.

In XLIFF2 such code can be represented using `<sc />`, `<ec/>` pair universally, or by `<pc></pc>` in case of well formed spanning code.

Ideally the original format is documented enough to instruct Extractor about role of each inline code. For example XML schema allows to declare elements using keyword EMPTY. This way all elements, which are not declared EMPTY, can be represented as described above. To further help the extraction process the following recommendation could be implemented in original XML format: [???For interoperability, the empty-element tag SHOULD be used, and SHOULD only be used, for elements which are declared EMPTY.]

•[spanning_as_ph] https://github.com/GALAglobal/TAPICC/tree/master/extraction_examples/spanning_as_ph •Extractor could use knowledge of schema and only use does not use <ph> for codes that are declared as EMPTY. To further help the extraction process, following W3C recommendation could be followed: „The empty-element tag *SHOULD* be used, and *SHOULD* only be used, for elements which are declared EMPTY.“ (https://www.w3.org/TR/REC-xml/#sec-starttags), e.g. even without content would use as compared to
. •https://issues.oasis-open.org/browse/XLIFF-14 http://docs.oasis-open.org/xliff/xliff-core/v2.1/xliff-core-v2.1.html#ph

Outermost Tag Pairs

•[outermost_inline_excluded] https://github.com/GALAglobal/TAPICC/tree/master/extraction_examples/outermost_inline_excluded •Both functional and formatting inline codes provide

additional context for translator and could be linguistically significant. •If they are important enough to be in native format, they should be present in extracted content.

Incomplete Extraction of Inline Codes

•[CDATA] https://github.com/GALAglobal/TAPICC/tree/master/extraction_examples/cdata •[inline_codes_plain_text] https://github.com/GALAglobal/TAPICC/tree/master/extraction_examples/inline_codes_plain_text •http://docs.oasis-open.org/xliff/xliff-core/v2.1/xliff-core-v2.1.html#d0e8112 •https://www.w3.org/TR/xml-i18n-bp/#AuthCDATA •Not using native XLIFF representation leaves inline codes unprotected and increases risk of roundtrip corrupting them.

Representing Multiple Subsequent Codes

•[multiple_codes_represented_as_single] https://github.com/GALAglobal/TAPICC/tree/master/extraction_examples/multiple_codes_represented_as_single •Grouping several independent inline codes into single representation could prove challenging with negative impact on •Translation quality •Fluency •Functionality •Automated actions •Validation •Some codes needs to be removed, copied, added or reordered. •If any of the above actions is to be prevented, it can be controlled using editing hints with finer granularity.

Target Content in Extracted XLIFF

Inserting unmodified source content into `<target>`

Inserting possible translation into `<target>`

State Machine

Editing and Context Hints

Non-deletable Inline Codes

Preserving Order of Codes

Controlling Segmentation

Providing Context

Context hints

Considerations for Using Spanning Codes

XLIFF Structure

File Structure

Role of <unit>

Miscellaneous

Value of attribute `id`

Whitespace Handling

Protecting Non-localizable Content

Merging Translated Content

Selecting Language Tags

Validation of Extracted Content

XLIFF Validations

Summary

References

Normative references

[] . Copyright © . .

Non-Normative References

[] . Copyright © . .