

XLIFF 2 slides for AMTA Standards Workshop

Dr. David Filip

ADAPT Centre

OASIS XLIFF TC, OASIS XLIFF OMOS TC, GALA TAPICC SC, NSAI expert to
ISO TC 37 & ISO/IEC JTC 1 /WG 9, /SC 38, /SC 42

What's XLIFF?

[Do we need this slide? Demographics of the room..]

- XML Localiz(s)ation Interchange File Format
- The only ***open standard* *bitext format***
- XLIFF lives in OASIS since 2001
- First fully standardized as XLIFF 1.2 in February 2008
- XLIFF 2.0 August 2014
- XLIFF 2.1 February 2018

XLIFF 2

XLIFF Version 2.1
OASIS Standard
13 February 2018

<http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/xliff-core-v2.1-os.html>

<http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/xliff-core-v2.1-os.pdf>

<http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/xliff-core-v2.1-os.xml>

XML schemas: <http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/schemas/>

<http://docs.oasis-open.org/xliff/xliff-core/v2.1/os/xliff-core-v2.1-os.html#mediaType>

What's new compared to 1.2

Overall: Focus on the “naïve” implementer, don't assume tribal knowledge

- **Fixed inline data model [NOT BACKWARDS COMPATIBLE]**
- **Explicit** fragment identification mechanism
-> **IANA registration of media type [*xlf]**
provisional for 2.0 -> permanent for 2.1
- Re-segmentation [**segmentation modification**] capability
[NOT BACKWARDS COMPATIBLE]
- Better managed extensibility **[NOT BACKWARDS COMPATIBLE]**
- Advanced Constraints and Processing Requirements
-> **Enforceable Conformance Clause**

What's new compared to 2.0

Overall: All XLIFF 2 is and will be backward & forwards compatible

- IANA registration of media type [***.xlf**]
provisional for 2.0 -> **permanent as of XLIFF Version 2.1 for 2.0 and higher**
- Better managed extensibility [since 2.0]
 - > Demoted 2.0 Change Tracking Module to an extension
 - > Promoted **ITS 2.0** extension to a **huge and powerful module in 2.1**
- Advanced Constraints and Processing Requirements
 - > Enforceable Conformance Clause [since 2.0]
 - > Translated the human readable conformance requirements into **an exhaustive set of fully machine readable declarative validation artifacts**
 - > **enables complex automated roundtrip workflows relying on strict validity**

Why should MT implementers care for XLIFF?

- XLIFF has a **simple and efficient inline data model** capable of **representing any content formats**, well formed or not
 - Native code representation – masking
 - **Powerful annotation mechanism with standoff capability**
 - Localization Note | Terminology | Text Analytics
 - Subsegment or supersegment Matching | Custom annotations | State and substate
 - MT Confidence | Localization Quality Issue [MQM] & Rating | Provenance
- XLIFF is **bitext**, therefore **alignment** of source and target segments is **100% guaranteed**
- XLIFF is extremely **metadata rich**, hence suitable for on the fly **creation of custom corpora**
 - Domain | Terminology | Language tags | Quality | Provenance

XLIFF 2 data model reuse

- Industry and public sector fast to adopt the superior XLIFF 2 data model. Desire to unleash the XLIFF 2 data model goodne, free it from XMLisms and SGMisms ;-)
 - > **Formation of OASIS XLIFF OMOS TC**
 - <https://www.oasis-open.org/apps/org/workgroup/xliff-omos/>
 - <https://github.com/oasis-tcs/xliff-omos-om>
 - <https://github.com/oasis-tcs/xliff-omos-jliff>
- **JLIFF feeds into TAPICC Tracks 2 & 3**
 - Real time exchange of data at unit level
 - While XLIFF mandates the whole document structure, JLIFF is specifically defined as a ***Fragment*** Interchange format [JSON Localization Interchange Fragment Format]

Q&A