



XLIFF 1.0 Specification

Committee Specification, 15 Apr 2002

This version:

<http://www.oasis-open.org/committees/xliff/documents/xliff-20020415.htm>

Latest version:

<http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm>

Previous version:

<http://www.oasis-open.org/committees/xliff/documents/xliff-20020320.htm>

Editors:

Yves Savourel <ysavourel@translate.com>

John Reid <jreid@novell.com> (Initial contribution editor)

Copyright © The Organization for the Advancement of Structured Information Standards [[OASIS](#)] 2002. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Abstract

This document defines the XML Localisation Interchange File Format (XLIFF). The purpose of this format is to store localisable data and carry it from one step of the localisation process to the other, while allowing interoperability between tools.

Status of this Document

This Committee Specification was approved for publication by the OASIS XLIFF Technical Committee. It is a stable document which represents the consensus of the committee. Comments may be sent to xliff-comment@lists.oasis-open.org.

Table of Contents

1. [Introduction](#)
 - 1.1. [Naming Convention](#)
 - 1.1.1. [Elements and Attributes](#)
 - 1.1.2. [Attribute Values](#)
 - 1.1.3. [Processing Instructions](#)
 - 1.1.4. [XLIFF File Extension](#)
2. [General Structure](#)
 - 2.1. [Header](#)
 - 2.2. [Body](#)
 - 2.3. [Named Groups](#)
3. [Detailed Specifications](#)
 - 3.1. [XML Declaration](#)
 - 3.2. [Elements](#)
 - 3.2.1. [Top-level and Header Elements](#)
 - 3.2.2. [Named Group Elements](#)
 - 3.2.3. [Structural Elements](#)
 - 3.2.4. [Inline Elements](#)
 - 3.2.5. [Delimiter Element](#)
 - 3.3. [Attributes](#)
 - 3.3.1. [XLIFF Attributes](#)
 - 3.3.2. [XML Namespace Attributes](#)

Appendices

- A. [XLIFF Tree Structure](#)
- B. [Document Type Definition for XLIFF](#)
- C. [Changes Since Previous Version \(Non-Normative\)](#)
- D. [XLIFF Technical Committee \(Non-Normative\)](#)
- E. [References](#)

1. Introduction

XLIFF is the XML Localisation Interchange File Format designed by a group of software providers, localisation service providers, and localisation tools providers. It is intended to give any software provider a single interchange file format that can be understood by any localisation provider. It is loosely based on the OpenTag version 1.2 specification and borrows from the TMX 1.2 specification. However, it is different enough from either one to be its own format.

1.1. Naming Convention

The following naming conventions were used in writing this specification.

1.1.1. Elements and Attributes

The following conventions were used for element and attribute naming.

1. Standard English letters.
2. Lower case only.
3. Hyphen, '-', may be used for concatenation.
4. Elements and attributes should not have the same name, even attributes of different elements.
5. Attribute names must be consistently defined throughout.
6. Industry standard terminology should be followed where possible.

1.1.2. Attribute Values

Attribute values are case sensitive. It is strongly recommended that lower-case values are used. The specification recommends a number of values for some attributes, these are all lower-case.

The specification also recommends the use of the semi-colon as a concatenation separator for values. For example, multiple contacts may be listed for a `<file>` with the attribute-value written thusly:
`contact-name="Peter;Mark".`

1.1.3. Processing Instructions

XLIFF reserves processing instructions that begin with "`xliff-`" for definition at some future time.

This specification has recommended [`xliff-show-context-group`](#) and [`xliff-show-context`](#) as a means of displaying context information to anyone processing a XLIFF file. Also, the processing instruction [`xliff-update-count-group`](#) has been given as an example.

1.1.4. XLIFF File Extension

XLIFF documents use the `.xlf` extension. This conforms to a 8.3 standard name. No other extension is recommended by the specification.

2. General Structure

XLIFF is XML, as such it begins with an XML declaration. After the XML declaration comes the XLIFF document itself, enclosed within the `<xliff>` element. A XLIFF document is composed of zero, one or more sections, each enclosed within a `<file>` element. The `<file>` element consists of a `<header>` element, which contains meta-data about the `<file>`, and a `<body>` element, which contains the extracted translatable data from the `<file>`. The translatable data is contained within `<trans-unit>` elements in `<source>` and `<target>` paired elements. These `<trans-unit>` elements can be grouped recursively in `<group>` elements.

In addition, XLIFF provides the ability to maintain information about the processing of the file via the `<phase>` element. Possible translations for a specific `<source>` element can be generated from any number of MT (Machine Translation) and CAT (Computer Assisted Translation) systems and stored near the `<source>` in `<alt-trans>` elements. Context for a `<source>` that could be used by a translator or a TM (Translation Memory) system is provided by the `<context>` element. Binary data can be made available via the `<bin-unit>`, which may also be translated and contain an associated `<trans-unit>`.

The complete tree structure is available in [Appendix A](#).

2.1. Header

The XLIFF `<header>` contains meta-data about the file and the localisation process. It contains the `<skl>`, `<phase-group>`, `<glossary>`, `<reference>`, `<count-group>`, `<prop-group>`, and `<note>` elements. The `<skl>` element contains either a skeleton file of the file submitted for localisation or a hypertext link to that file.

The `<phase-group>` element contains information about each processing phase used in localising the file; references to these phases are stored along with the translations. The `<glossary>` and `<reference>` elements may contain hypertext links to a glossary and reference file, respectively, or the actual glossary and reference data that can be used in the localisation process.

The `<count-group>` element is a grouping element of count information of the entire file. The `<prop-group>` element contains tool-specific information used in combining the data with the skeleton file or storing the data in a repository. The `<note>` element contains instructions for the localisation process. The `<count-group>`, `<prop-group>`, and `<note>` elements can also appear in the body of the file.

2.2. Body

The XLIFF `<body>` contains the structure and the localisable content from the file. It contains the `<group>`, `<trans-unit>` and `<bin-unit>` elements. The structure is described using the `<group>`, `<trans-unit>`, `<bin-unit>` elements. The `<group>` element is a general purpose structural element that allows describing the hierarchy of the file; it can contain other `<group>` elements as children as well as `<trans-unit>` and `<bin-unit>` elements.

The `<trans-unit>` and `<bin-unit>` elements are the leaf nodes of the tree structure. The `<trans-unit>` element contains the text to be translated, the translations, and other related information. The `<bin-unit>` contains binary data that may or may not need to be translated; it also can contain

translated versions of the binary object as well as other related information.

In the `<trans-unit>` element the text to be translated is contained in a `<source>` element. This element may contain inline elements that either remove the codes from the source (`<g>`, `<x/>`, `<bx/>`, `<ex/>`) or that mask off codes left inline (`<bpt>`, `<ept>`, `<sub>`, `<it>`, `<ph>`). The translated text is contained in a `<target>` element that has the same inline codes available to it as does the `<source>` element. Translation matches generated by a TM or MT or entered by a translator may be provided in a `<alt-trans>` element, which also contains the `<source>` and `<target>` elements.

At every structural level contextual information for the localisation process can be provided by the `<context-group>` named group element, count information by the `<count-group>` named group element, and tool-specific information by the `<prop-group>` named group element.

2.3. Named Groups

XLIFF allows grouping of certain elements into named groups. A named group is simply a grouping element with a name attribute. These named groups can be interspersed throughout the file with information designed for specific purposes. Using XML processing instructions different actions can be performed with specific named groups. The named group elements are `<context-group>`, `<count-group>` and `<prop-group>`.

The `<context-group>` element contains context information for the source strings to be used in the translation process. There could be context information that is useful to the translator that should be shown and context information that is designed specifically for a translation tool. Each of these groups of context information can be grouped in a named `<context-group>` and processing instructions can tell the systems which `<context-group>` to use for each purpose.

The `<count-group>` element contains counts of words, translations, dialogs, or anything else that may need to be counted in the file. A different named group could be stored by the client, translator, reviewer, and localisation engineer. Processing instructions could inform a system which of these `<count-group>` to update during the localisation process.

The `<prop-group>` element contains tool specific data that can be used in creating the translated file, storing the translations, and any other specific task. Processing instructions can indicate to the tools which named `<prop-group>` to use when updating the repository or combining the localised data with the skeleton file to create a translated file.

3. Detailed Specifications

3.1. XML Declaration

The XML declaration is strongly recommended. It indicates the XML version and sets the defaults for the encoding of the file. For example, the following declaration specifies the document is in ISO 8859-1, the Latin-1 encoding.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

As in all XML files, the default encoding for an XLIFF file is assumed to be either UTF-8, which is a superset of the 7-bit ASCII character set, or UTF-16, which is UCS-2 with surrogate pairs for code points above U+FFFF. Thus, for these character sets, the encoding declaration is not necessary. Further, all XML parsers support these encodings. If the encoding is in UTF-16 the first character of the file must be the Unicode Byte-Order-Mark, U+FEFF, which indicates the endianness of the file. Other encodings may be desirable and may be generally supported by XML parsers. These must be declared using the encoding declaration. The values to use for the encoding declaration are defined in the [\[IANA Charsets\]](#) listing.

In addition to the XML declaration, it is recommended to use the following DOCTYPE declaration:

```
<!DOCTYPE xliiff PUBLIC "-//XLIFF//DTD XLIFF//EN"
"http://www.oasis-open.org/committees/xliiff/documents/xliiff.dtd" >
```

If necessary, you can also specify a namespace for XLIFF. The namespace identifier for this standard is "urn:oasis:names:tc:xliiff:1.0".

A minimum XLIFF document looks something like this:

```
<?xml version="1.0" ?>
<!DOCTYPE xliiff PUBLIC "-//XLIFF//DTD XLIFF//EN"
"http://www.oasis-open.org/committees/xliiff/documents/xliiff.dtd" >
<xliiff version="1.0">
  <file source-language="EN" datatype="plaintext" original="file.ext">
    <header></header>
    <body>
      <trans-unit id="1">
        <source>Hello World!</source>
      </trans-unit>
    </body>
  </file>
</xliiff>
```

3.2. Elements

XLIFF elements can be divided into five main categories: the top-level and header elements, the named group elements, the structural elements, the inline elements, and the delimiter elements. [Attributes](#) are shared among them.

Top Level and Header elements	<xliiff> , <file> , <header> , <skl> , <external-file> , <internal-file> , <glossary> , <reference> , <phase-group> , <phase> , <note> .
Named Group Elements	<context-group> , <context> , <count-group> , <count> , <prop-group> , <prop> .
Structural elements	<body> , <group> , <trans-unit> , <source> , <target> , <bin-unit> , <bin-source> , <bin-target> , <alt-trans> .

Inline elements	<g> , <x/> , <bx/> , <ex/> , <bpt> , <ept> , <sub> , <it> , <ph> .
Delimiter element	<mrk> .

3.2.1. Top-level and Header Elements

The top-level and header elements are the following:

<xliff>

XLIFF document - The [<xliff>](#) element encloses all the other elements of the document.

Required attributes:

[version](#).

Optional attributes:

[xml:lang](#).

Contents:

One or more [<file>](#) elements.

<file>

File - The [<file>](#) element corresponds to a single extracted original document.

Required attributes:

[original](#), [source-language](#), [datatype](#).

Optional attributes:

[tool](#), [date](#), [xml:space](#), [ts](#), [category](#), [target-language](#), [product-name](#), [product-version](#), [build-num](#).

Contents:

One [<header>](#) followed by one [<body>](#) element

<header>

File header - The [<header>](#) element contains data relating to the [<file>](#) element.

Required attributes:

None.

Optional attributes:

None.

Contents:

zero or one [<skl>](#) followed by
zero or one [<phase-group>](#) followed by
zero, one or more [<prop-group>](#), [<glossary>](#), [<reference>](#), [<note>](#), [<count-group>](#)
elements in any order.

<skl>

Sekeleton file - The [<skl>](#) element contains the skeleton file or the location of the skeleton file.

Required attributes:

None.

Optional attributes:

None

Contents:

Either exactly one [<internal-file>](#) or one [<external-file>](#) element.

<internal-file>

Internal file - The [<internal-file>](#) element will contain the data for the skeleton file.

Required attributes:

None.

Optional attributes:

[form](#), [crc](#).

Contents:

An embedded file.

<external-file>

External file - An empty element that specifies the file location.

Required attributes:

[href](#).

Optional attributes:

[uid](#), [crc](#).

Contents:

The `<external-file>` is an empty element, including attributes only.

<glossary>

Glossary - Element points to the glossary source.

Required attributes:

None.

Optional attributes:

None.

Contents:

The glossary description and either exactly one [<internal-file>](#) or one [<external-file>](#) element.

<reference>

Reference - Contains information about the reference material.

Required attributes:

None.

Optional attributes:

None.

Contents:

A description of the reference material and either exactly one [<internal-file>](#) or one [<external-file>](#) element.

<note>

Note - The `<note>` element is used to add localisation-related comments to the XLIFF document. The content of `<note>` may be instructions from developers about how to handle the [<source>](#), comments from the translator about the translation, or any comment from anyone involved in processing the XLIFF file.

Required attributes:

None.

Optional attributes:

[xml:lang](#), [from](#), [priority](#).

Contents:

Text, no standard elements.

<phase-group>

Phase group - The <phase-group> element contains phase information. This phase information is specific to the users.

Required attributes:

None.

Optional attributes:

None.

Contents:

One or more [<phase>](#) elements.

<phase>

Phase information - The <phase> contains metadata about a particular version of XLIFF data.

Required attributes:

[phase-name](#), [process-name](#).

Optional attributes:

[company-name](#), [tool](#), [date](#), [job-id](#), [contact-name](#), [contact-email](#), [contact-phone](#).

Contents:

Zero, one or more [<note>](#) elements.

3.2.2. Named Group Elements

The named group elements are the following:

<count-group>

Count group - The `<count-group>` element holds count elements relating to the level in the tree in which it occurs. Each group for `<count>` elements must be named, allowing different uses for each group. These uses can be controlled through the use of XML processing instructions. For example, the following instruction could indicate that any `<count-group>` element with a name set to 'value' should be updated during translation:

```
<?xliff-update-count-group name='value' ?>
```

Required attributes:

[name](#).

Optional attributes:

None.

Contents:

One or more `<count>` elements.

<count>

Count - For each `<count>` element the the required `count-type` attribute would indicate what kind of count the element represents, and the optional `unit` attribute would indicate the unit of the count (by default: `word`). A list of recommended values for `count-type` and `unit` is provided by the specification.

Required attributes:

[count-type](#).

Optional attributes:

[unit](#).

Contents:

Number (the count value).

<context-group>

Context group - The `<context-group>` element holds context elements relating to the level in the tree in which it occurs. Thus context can be set at a `<group>` level, a `<trans-unit>` level, or a `<alt-trans>` level.

Each `<context-group>` element must be named, allowing different uses for each group. These uses can be controlled through the use of XML processing instructions. For example, the following instruction could indicate that any `<context-group>` element with a name set to 'value' should be displayed to the end-user:

```
<?xliff-show-context-group name='value' ?>
```

Because the `<context-group>` element may occur at a very high level, a default context can be established for all `<trans-unit>` elements within a file. This default can be overridden at many subsequent levels.

Required attributes:

`name`.

Optional attributes:

`crc`.

Contents:

One or more `<context>` elements.

<context>

Context - The `<context>` element describes the context of a `<source>` within a `<trans-unit>` or a `<alt-trans>`. The purpose of this context information is to allow certain pieces of text to have different translations depending on where they came from. The translation of a piece of text may differ if it is a web form or a dialog or an Oracle form or a Lotus form for example. This information is thus required by a translator when working on the file. Likewise, the information may be used by any tool proposing to automatically leverage the text successfully.

The visibility of `<context>` information can be controlled by XML processing instructions. For example, the following instruction may indicate that any `<context>` element with a `context-type` set to 'value' should be displayed to the end-user.

```
<?xliff-show-context context-type='value' ?>
```

Required attributes:

`context-type`

Optional attributes:

`match-mandatory`, `crc`

Contents:

Text, no standard elements.

<prop-group>

Property group - The `<prop-group>` element contains `<prop>` elements. Each `<prop-group>` element

may be named, allowing different uses for each group. These uses can be controlled through the use of XML processing instructions.

Required attributes:

None.

Optional attributes:

[name](#).

Contents:

One or more [<prop>](#) elements.

<prop>

Property - The [<prop>](#) element allows the tools to specify non-standard information in the XLIFF document. This information can be used by the tools that have produced the file or that translate the file or that do any other amount of processing specific to the producer.

Required attributes:

[prop-type](#).

Optional attributes:

[xml:lang](#).

Contents:

Tool-specific data or text, no standard elements.

3.2.3. Structural Elements

The structural elements specify the frame of a XLIFF document as well as contextual and processing information. The [<source>](#) element contains the extracted data and, possibly, [inline elements](#).

<body>

File body - The [<body>](#) element contains the structural elements.

Required attributes:

None.

Optional attributes:

None.

Contents:

Zero, one or more [<group>](#), [<trans-unit>](#), [<bin-unit>](#) elements in any order.

<group>

Group - The [<group>](#) element specifies a set of elements that should be processed together. For example: all the items of a menu, etc. Note that a [<group>](#) element can contain other [<group>](#) elements.

Required attributes:

None.

Optional attributes:

[id](#), [datatype](#), [xml:space](#), [ts](#), [restype](#), [resname](#), [extradata](#), [help-id](#), [menu](#), [menu-option](#), [menu-name](#), [coord](#), [font](#), [css-style](#), [style](#), [exstyle](#), [extype](#).

Contents:

Zero, one or more [<context-group>](#) elements followed by

Zero, one or more [<count-group>](#) elements followed by

Zero, one or more [<prop-group>](#) elements followed by

Zero, one or more [<note>](#) elements followed by

Zero, one or more [<group>](#), [<trans-unit>](#), [<bin-unit>](#) elements in any order.

All [<context-group>](#), [<count-group>](#), [<prop-group>](#), and [<note>](#) elements pertain to the subsequent elements in the tree but can be overridden within a child element.

<trans-unit>

Translation unit - The [<trans-unit>](#) elements contains a [<source>](#), [<target>](#) and associated elements. Lists of recommended values for the [datatype](#), [restype](#), and [size-unit](#) attributes are available.

Required attributes:

[id](#).

Optional attributes:

[approved](#), [translate](#), [reformat](#), [xml:space](#), [datatype](#), [ts](#), [phase-name](#), [restype](#), [resname](#), [extradata](#), [help-id](#), [menu](#), [menu-option](#), [menu-name](#), [coord](#), [font](#), [css-style](#), [style](#), [exstyle](#), [extype](#), [maxbytes](#), [minbytes](#), [size-unit](#), [maxheight](#), [minheight](#), [maxwidth](#), [minwidth](#), [charclass](#).

Contents:

One [<source>](#) element followed by

Zero or one [<target>](#) elements followed by

Zero, one or more [<note>](#), [<context-group>](#), [<prop-group>](#), [<alt-trans>](#), [<count-group>](#)

elements in any order.

All child elements of <trans-unit> pertain to their sibling [<source>](#) element.

<source>

Source text - The <source> element is used to delimit a unit of text that could be a paragraph, a title, a menu item, a caption, etc.

Required attributes:

None.

Optional attributes:

[xml:lang](#), [ts](#).

Contents:

Text,

Zero, one or more of the following elements: [<q>](#), [<x/>](#), [<bx/>](#), [<ex/>](#), [<bpt>](#), [<ept>](#), [<ph>](#), [<it>](#), [<mrk>](#), in any order.

<target>

Target - The <target> element is used to delimit a unit of text. A paragraph in XLIFF does not necessarily correspond to a "paragraph" in a word-processor. It's simply a unit of text that could be a paragraph, a title, a menu item, a caption, etc. A list of preferred values for the [restype](#) attribute is available.

Required attributes:

None.

Optional attributes:

[state](#), [phase-name](#), [xml:lang](#), [ts](#), [restype](#), [rename](#), [coord](#), [font](#), [css-style](#), [style](#), [exstyle](#).

Contents:

Text,

Zero, one or more of the following elements: [<q>](#), [<x/>](#), [<bx/>](#), [<ex/>](#), [<bpt>](#), [<ept>](#), [<ph>](#), [<it>](#), [<mrk>](#), in any order.

<alt-trans>

Translation match - The <alt-trans> element contains a possible translation in a <target> along with optional context, notes, etc.

Required attributes:

None.

Optional attributes:

[match-quality](#), [tool](#), [crc](#), [xml:lang](#), [datatype](#), [xml:space](#), [ts](#), [restype](#), [resname](#), [extradata](#), [help-id](#), [menu](#), [menu-option](#), [menu-name](#), [coord](#), [font](#), [css-style](#), [style](#), [exstyle](#), [extype](#), [origin](#).

Contents:

Zero or One [<source>](#), followed by
One or more [<target>](#), followed by
Zero, one or more [<note>](#), [<context-group>](#), [<prop-group>](#) elements in any order.
All child elements of [<alt-trans>](#) pertain to their sibling [<target>](#) element.

<bin-unit>

Binary unit - The [<bin-unit>](#) element contains a binary object that may or may not be translatable.

Required attributes:

[id](#), [mime-type](#).

Optional attributes:

[approved](#), [translate](#), [reformat](#), [ts](#), [phase-name](#), [restype](#), [resname](#).

Contents:

One [<bin-source>](#) element followed by
Zero or one [<bin-target>](#) elements followed by
Zero, one or more [<note>](#), [<context-group>](#), [<prop-group>](#), [<trans-unit>](#), [<count-group>](#)
elements in any order.
All child elements of [<bin-unit>](#) pertain to their sibling [<source>](#) element.

<bin-source>

Binary source - The [<bin-source>](#) element is the container for the binary source data.

Required attributes:

None.

Optional attributes:

[ts](#).

Contents:

One of [<internal-file>](#) or [<external-file>](#).

<bin-target>

Binary target - The <bin-target> element is the container for the translated version of the binary data.

Required attributes:

None.

Optional attributes:

[mime-type](#), [ts](#), [state](#), [phase-name](#), [restype](#), [resname](#).

Contents:

One of [<internal-file>](#) or [<external-file>](#).

3.2.4. Inline Elements

The inline elements are the elements that can appear inside the [<source>](#) and [<target>](#) elements. They enclose or replace any formatting or control codes that is not text, but resides within the text unit.

<g>

Generic group placeholder - The <g> element is used to replace any inline code of the original document that has a beginning and an end and can be moved within its parent structural element. When possible, the [ctype](#) allows you to specify what kind of attribute the placeholder represents. A list of preferred values for the [ctype](#) attribute is available.

A <g> element can contain another <g> element. In this case, if the embedded group has an [id](#) attribute, it should never be moved outside of its parent group.

Required attributes:

[id](#).

Optional attributes:

[ctype](#), [ts](#), [clone](#).

Contents:

Text,

Zero, one or more of the following elements: [<g>](#), [<x/>](#), [<bx/>](#), [<ex/>](#), [<bpt>](#), [<ept>](#), [<ph>](#), [<it>](#), [<mrk>](#), in any order.

<x/>

Generic placeholder - The `<x/>` element is used to replace any code of the original document. When possible, the `ctype` allows you to specify what kind of attribute the placeholder represents. A list of preferred values for the `ctype` attribute is available.

Required attributes:

`id`.

Optional attributes:

`ctype`, `ts`, `clone`.

Contents:

Empty.

`<bx/>`

Begin paired placeholder - The `<bx/>` element is used to replace a beginning paired code of the original document. It should be used for paired codes that do not follow XML well-formedness rules (i.e. no overlapping elements). If the paired codes follow that rule, it is strongly recommended that the `<g>` element is used because it simplifies processing. The `<bx/>` element should be followed by a matching `<ex/>` element. These paired elements are related via their `rid` attributes. When possible, the `ctype` allows you to specify what kind of attribute the placeholder represents. A list of preferred values for the `ctype` attribute is available.

Required attributes:

`id`.

Optional attributes:

`rid`, `ctype`, `ts`, `clone`.

Contents:

Empty.

`<ex/>`

End paired placeholder - The `<ex/>` element is used to replace a beginning paired code of the original document. It should be used for paired codes that do not follow XML well-formedness rules (i.e. no overlapping elements). If the paired codes follow that rule, it is strongly recommended that the `<g>` element is used because it simplifies processing. The `<ex/>` element should be preceded by a matching `<bx/>` element. These paired elements are related via their `rid` attributes.

Required attributes:

`id`.

Optional attributes:

[rid](#), [ts](#).

Contents:

Empty.

<ph>

Placeholder - The <ph> element is used to delimit a sequence of native stand-alone codes in the segment. When possible, the [ctype](#) allows you to specify what kind of attribute the placeholder represents. A list of preferred values for the [ctype](#) attribute is available.

Required attributes:

[id](#).

Optional attributes:

[ctype](#), [ts](#), [crc](#), [assoc](#).

Contents:

Code data,
Zero, one or more [<sub>](#) elements.

<bpt>

Begin paired tag - The <bpt> element is used to delimit the beginning of a paired sequence of native codes. Each <bpt> has a corresponding [<ept>](#) element within the segment. When possible, the [ctype](#) allows you to specify what kind of attribute the placeholder represents. A list of preferred values for the [ctype](#) attribute is available.

Required attributes:

[id](#).

Optional attributes:

[rid](#), [ctype](#), [ts](#), [crc](#).

Contents:

Code data,
Zero, one or more [<sub>](#) elements.

<ept>

End paired tag - The <ept> element is used to delimit the end of a paired sequence of native codes. Each <ept> has a corresponding [<bpt>](#) element within the segment.

Required attributes:

[id](#).

Optional attributes:

[rid](#), [ts](#), [crc](#).

Contents:

Code data,
Zero, one or more [<sub>](#) elements.

<it>

Isolated tag - The `<it>` element is used to delimit a beginning/ending sequence of native codes that does not have its corresponding ending/beginning within the segment. When possible, the [ctype](#) allows you to specify what kind of attribute the placeholder represents. A list of preferred values for the [ctype](#) attribute is available.

Required attributes:

[id](#), [pos](#).

Optional attributes:

[rid](#), [ctype](#), [ts](#), [crc](#).

Contents:

Code data,
Zero, one or more [<sub>](#) elements.

<sub>

Sub-flow - The `<sub>` element is used to delimit sub-flow text inside a sequence of native code, for example: the definition of a footnote or the text of a `title` attribute in a HTML `<a>` element. When possible, the [ctype](#) allows you to specify what kind of attribute the placeholder represents. Lists of preferred values for the [ctype](#) and [datatype](#) attributes are available.

Required attributes:

None.

Optional attributes:

[datatype](#), [ctype](#).

Contents:

Text,
Zero, one or more of the following elements: [<g>](#), [<x/>](#), [<bx/>](#), [<ex/>](#), [<bpt>](#), [<ept>](#), [<ph>](#),

[<it>](#), [<mrk>](#), in any order.

3.2.5. Delimiter Element

XLIFF defines an additional element to support various types of text processing. This element is usually not generated by the extraction module and are ignored most of the time during merging, but it can be very powerful with tools such as Machine Translation, glossary handling, quality assurance, etc.

<mrk>

Marker - The [<mrk>](#) element delimits a section of text that has special meaning, such as a terminological unit, a proper name, an item that should not be modified, etc. It can be used for various processing tasks. For example, to indicate to a Machine Translation tool proper names that should not be translated; for terminology verification, to mark suspect expressions after a grammar checking. The [<mrk>](#) element is usually not generated by the extraction tool and it is not part of the tags used to merge the XLIFF file back into its original format. A list of preferred values for the [mtype](#) attribute is available.

Required attributes:

[mtype](#).

Optional attributes:

[mid](#), [ts](#), [comment](#).

Contents:

Text,

Zero, one or more of the following elements: [<g>](#), [<x/>](#), [<bx/>](#), [<ex/>](#), [<bpt>](#), [<ept>](#), [<ph>](#), [<it>](#), [<mrk>](#), in any order.

3.3. Attributes

This section lists the various attributes used in the XLIFF elements. An attribute is never specified more than once for each element. Along with some of the attributes are the "Recommended Attribute Values". Values for these attributes are case sensitive. These lists are purely informative, the goal is to specify a preferred syntax so tools can have some level of compatibility.

XLIFF attributes	
	approved , assoc , build-num , ctype , category , charclass , comment , company-name , contact-email , contact-name , contact-phone , coord , count-type , crc , css-style , datatype , date , exstyle , extradata , extype , font , form , from , help-id , href , id ,

	job-id , match-mandatory , match-quality , maxheight , maxbytes , maxwidth , menu , menu-name , menu-option , mid , mime-type , minheight , minbytes , minwidth , mtype , name , original , phase-name , pos , priority , process-name , product-name , product-version , prop-type , reformat , resname , restype , rid , source-language , state , style , tool , target-language , translate , ts , uid , unit , version .
XML namespace attributes	xml:lang , xml:space .

3.3.1. XLIFF Attributes

approved

Approved - Indicates whether a translation is final.

Value description:

Boolean: *yes* or *no*.

Default value:

no.

Used in:

[<trans-unit>](#), [<bin-unit>](#).

assoc

Association - Indicates the association of a [<ph>](#) with the text prior or after.

Value description:

preceding (the element is associated with the text preceding the element), *following* (the element is associated with the text following the element), and *both* (the element is associated with the text on both sides).

Default value:

Undefined.

Used in:

[<ph>](#).

build-num

Build number - The build number of version of the product or application the localizable material is for.

Value description:

Alpha-numeric.

Default value:

Undefined.

Used in:

[<file>](#).

category

Category - This would give information on the subject of what is being translated.

Value description:

Text.

Default value:

Undefined.

Used in:

[<file>](#).

charclass

Character class - This indicates that a translation is restricted to a subset of characters (e.g. ASCII only, Katakana only, uppercase only, etc.). A blank value indicates there is no limitation.

Value description:

Text.

Default value:

Undefined.

Used in:

[<trans-unit>](#).

clone

Clone - This indicates that a copy of the given inline element can be made and placed in the [<target>](#).

Value description:

Boolean: yes or no.

Default value:

yes.

Used in:

[<g>](#), [<x/>](#), [<bx/>](#).

comment

Comment - A comment in a tag.

Value description:

Alpha-numeric.

Default value:

Undefined.

Used in:

[<mrk>](#).

company-name

Company name - The client name.

Value description:

Text.

Default value:

Undefined.

Used in:

[<phase>](#).

contact-email

Contact email - The contact email at the client company (email of the contact-name person).

Value description:

Text.

Default value:
Undefined.

Used in:
[<phase>](#).

contact-name

Contact name - The contact at the client company.

Value description:
Text.

Default value:
Undefined.

Used in:
[<phase>](#).

contact-phone

Contact phone - Phone number of the contact-name person.

Value description:
Text.

Default value:
Undefined.

Used in:
[<phase>](#).

context-type

Context type - The `context-type` attribute specifies the context and the type of resource or style of the data of a given element. For example, to define if it is a label, or a menu item in the case of resource-type data, or the style in the case of document-related data.

Value description:
Text. The recommended values for the `context-type` attribute of the [<context>](#) element are as follow (this list is not exhaustive):
- Database

- Element
- ElementTitle
- Record
- RecordTitle

Default value:

Undefined.

Used in:

[<context>](#).

coord

Coordinates - The `coord` attribute specifies the x, y, cx and cy coordinates of the text for a given element. The cx and cy values must represent the width and the height (like in Windows resources). The extraction and merging tools must make the right conversion if the original format uses a top-left/bottom-right coordinate system.

Value description:

Four decimal (possibly negative) values, in the order: x,y,cx and cy, separated by semi-colons. Null values may be entered as "#"; (e.g. `coord="#;#;183;272"`).

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<target>](#), [<alt-trans>](#).

count-type

Count type - The `count-type` attribute specifies the purpose of the [<count>](#) element.

Value description:

Text. These values may correspond to phases. The recommended values for the `count-type` attribute are as follow (this list is not exhaustive):

- `new` = New items to translate.
- `exact-match` = Items that have at least one exact match (in a [<alt-trans>](#) element).
- `fuzzy-match` = Items that have at least one fuzzy match (in a [<alt-trans>](#) element).
- `total` = Total count for the given item.

Default value:

None.

Used in:

[<count>](#).

crc

Cyclic redundancy checking - A private value used to verify data as it is returned to the producer. The generation and verification of this number is tool-specific.

Value description:

Number (possibly not decimal).

Default value:

None.

Used in:

[<internal-file>](#), [<external-file>](#), [<context-group>](#), [<context>](#), [<alt-trans>](#), [<bpt>](#), [<ept>](#), [<it>](#), [<ph>](#).

css-style

Cascading style-sheet style - The `css-style` attribute allows any valid CSS style statement to be specified.

Value description:

Text, the value is subject to CSS syntax rules.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<target>](#), [<alt-trans>](#).

ctype

Content type - The type attribute specifies the content and the type of resource or style of the data of a given element. For example, to define if it is a label, or a menu item in the case of resource-type data, or the style in the case of document-related data.

Value description:

The value will depend on each element. The recommended values for the `ctype` attribute of the [<x/>](#) and [<ph>](#) elements are as follow (this list is not exhaustive):

- pb = Paragraph break.
- lb = Line-break.

- image = Image or other graphic.

Text. The recommended values for the `ctype` attribute of the other elements are as follow (this list is not exhaustive):

- bold = Bold or strong text.
- font = Font size, font face, color change, etc.).
- italic = Italicized text.
- underlined = Underlined text.
- link = hypertext link.

Default value:

Undefined.

Used in:

[<g>](#), [<x/>](#), [<bx/>](#), [<bpt>](#), [<sub>](#), [<it>](#), [<ph>](#).

datatype

Data type - The `datatype` attribute specifies the kind of text contained in the element. Depending on that type, you may apply different processes to the data.

Value description:

Text. The recommended values for the `datatype` attribute are as follow (this list is not exhaustive):

- cdf = Channel Definition Format.
- cpp = C and C++ style text.
- html = HTML, DHTML, etc.
- interleaf = Interleaf documents.
- java = Java, source and property files.
- javascript = JavaScript, ECMAScript scripts.
- lisp = Lisp.
- mif = Framemaker MIF, MML, etc.
- pascal = Pascal, Delphi style text.
- plaintext = Plain text.
- rtf = Rich Text Format.
- sgml = SGML.
- vbscript = Visual Basic scripts.
- winres = Windows resources from RC, DLL, EXE.
- xml = XML.

Default value:

Empty string.

Used in:

[<file>](#), [<group>](#), [<trans-unit>](#), [<alt-trans>](#), [<sub>](#).

date

Date - The `date` attribute indicates when a given element was created or modified.

Value description:

Date in [ISO 8601] Format. The recommended pattern to use is: `CCYY-MM-DDThh:mm:ssZ`

Where: `CCYY` is the year (4 digits), `MM` is the month (2 digits), `DD` is the day (2 digits), `hh` is the hours (2 digits), `mm` is the minutes (2 digits), `ss` is the second (2 digits), and `Z` indicates the time is UTC time. For example:

```
date="2002-01-25T21:06:00Z"  
is January 25, 2002 at 9:06pm GMT  
is January 25, 2002 at 2:06pm US Mountain Time  
is January 26, 2002 at 6:06am Japan time
```

Default value:

Undefined.

Used in:

[<file>](#), [<phase>](#).

exstyle

Extended style - The `exstyle` attribute stores the extended style of a control. For example, in Windows resources it corresponds to the EXSTYLE statement.

Value description:

Text.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<target>](#), [<alt-trans>](#).

extradata

Extra data - The `extradata` attribute stores the extra data properties of an item.

Value description:

Text.

Default value:
Undefined.

Used in:
[<group>](#), [<trans-unit>](#), [<alt-trans>](#).

extype

Extended type - The `extype` attribute stores the extra type properties of an item.

Value description:
Text.

Default value:
Undefined.

Used in:
[<group>](#), [<trans-unit>](#), [<alt-trans>](#).

font

Font - The `font` attribute specifies the font name, size, and weight of the text for a given element. The `font` attribute would generally be used for resource-type data: change of font in document-type data can be marked with the [<q>](#) element.

Value description:
Name of the font and its size, weight separated by a semi-colon.

Default value:
Undefined.

Used in:
[<group>](#), [<trans-unit>](#), [<target>](#), [<alt-trans>](#).

form

Format - Describes the type of format used in an [<internal-file>](#) element.

Value description:
The value can be either `text` (for plain text data), `base64` (for data coded in base64 format), or one of values available from the [\[RFC 1341\]](#) document: the MIME specification.

Default value:

text.

Used in:

[<internal-file>](#).

from

From - Indicates the author of a [<note>](#) element.

Value description:

Text.

Default value:

Undefined.

Used in:

[<note>](#).

help-id

Help ID - The `help-id` attribute stores the help identifier of an item. For example, in Windows resources it corresponds to the Help ID parameter of a control.

Value description:

Number.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<alt-trans>](#).

href

Hypertext reference - The location of the file or the URL for an [<external-file>](#) element.

Value description:

Text.

Default value:

Undefined.

Used in:

[<external-file>](#).

id

Identifier - The `id` attribute is used in many elements, as a unique reference to the original corresponding code data or format for the given element. the `id` element is not the identifier of a resource: use [resname](#) for that purpose.

Value description:

Alpha-numeric without spaces.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<bin-unit>](#), [<g>](#), [<x/>](#), [<bx/>](#), [<ex/>](#), [<bpt>](#), [<ept>](#), [<it>](#), [<ph>](#).

job-id

Job ID - The identifier given to the localisation job.

Value description:

Text.

Default value:

Undefined.

Used in:

[<phase>](#).

match-mandatory

Match mandatory - Indicates that any [<alt-trans>](#) element of the parent [<trans-unit>](#) must have the same [<context>](#) as the [<trans-unit>](#).

Value description:

Boolean: yes or no.

Default value:

no.

Used in:

[<context>](#).

match-quality

Match quality - The match quality of the [<alt-trans>](#) element. This value is tool specific and can be a score expressed in percentage or an arbitrary value (e.g. `match-quality="high"`).

Value description:

Text.

Default value:

Undefined.

Used in:

[<alt-trans>](#).

maxheight

Maximum height - The maximum height for the [<target>](#) of a [<trans-unit>](#). This could be interpreted as lines, pixels, or any other relevant unit. The unit is determined by the [size-unit](#) attribute, which defaults to `pixel`.

Value description:

Number.

Default value:

Undefined.

Used in:

[<trans-unit>](#).

maxbytes

Maximum bytes - The maximum number of bytes for the [<target>](#) of a [<trans-unit>](#). The verification of whether the relevant text respects this requirement must be done using the encoding and line-break type of the final target environment.

Value description:

Number.

Default value:
Undefined.

Used in:
[<trans-unit>](#).

maxwidth

Maximum width - The maximum width for the [<target>](#) of a [<trans-unit>](#). This could be interpreted as lines, pixels, or any other relevant unit. The unit is determined by the [size-unit](#) attribute, which defaults to `pixel`.

Value description:
Number.

Default value:
Undefined.

Used in:
[<trans-unit>](#).

menu

Menu - The `menu` attribute stores the menu property of an item.

Text.

Default value:
Undefined.

Used in:
[<group>](#), [<trans-unit>](#), [<alt-trans>](#).

menu-name

Menu name - The `menu-name` attribute stores the menu name of a control.

Value description:
Text.

Default value:
Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<alt-trans>](#).

menu-option

Menu option - The `menu-option` attribute stores the option data of a control.

Value description:

Text.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<alt-trans>](#).

mid

Marker ID - Identifier for an [<mrk>](#) element.

Value description:

Text.

Default value:

Undefined.

Used in:

[<mrk>](#).

mime-type

Mime type - Indicates the type of a binary object. This is important in determining how to edit the binary object.

Value description:

Text. A list of preferred values is available from the [\[RFC 1341\]](#) document: the MIME specification.

Default value:

Undefined.

Used in:

[<bin-unit>](#), [<bin-target>](#).

minheight

Minimum height - The minimum height for the [<target>](#) of a [<trans-unit>](#). This could be interpreted as lines, pixels, or any other relevant unit. The unit is determined by the [size-unit](#) attribute, which defaults to `pixel`.

Value description:

Number.

Default value:

Undefined.

Used in:

[<trans-unit>](#).

minbytes

Minimum bytes - The minimum number of bytes for the [<target>](#) of a [<trans-unit>](#). The verification of whether the relevant text respects this requirement must be done using the encoding and line-break type of the final target environment.

Value description:

Number.

Default value:

Undefined.

Used in:

[<trans-unit>](#).

minwidth

Minimum width - The minimum width for the [<target>](#) of a [<trans-unit>](#). This could be interpreted as lines, pixels, or any other relevant unit. The unit is determined by the [size-unit](#) attribute, which defaults to `pixel`.

Value description:

Number.

Default value:

Undefined.

Used in:

[<trans-unit>](#).

mtype

Marker type - The `mtype` attribute specifies what an [<mrk>](#) element is defining within the content of a [<source>](#) or [<target>](#) element.

Value description:

Text. The recommended values for the `mtype` attribute are as follow (this list is not exhaustive):

- `abbrev` = abbreviation, acronym, etc.
- `datetime` = date or time information.
- `name` = proper or common name.
- `phrase` = sub-sentence level.
- `protected` = text that should remain untouched during the process.
- `term` = one or more words of a terminology entry.

Default value:

Undefined.

Used in:

[<mrk>](#).

name

Name - The `name` attribute specifies the user-defined name of a named group element.

Value description:

Text.

Default value:

Undefined.

Used in:

[<prop-group>](#), [<context-group>](#), [<count-group>](#).

origin

Translation Match Origin - The `origin` attribute specifies where a translation match came from; for example, from a previous version of the same product, a different product, a shared translation

memory, etc.

Value description:

Text.

Default value:

Undefined.

Used in:

[<alt-trans>](#).

original

Original file - The original attribute specifies the name of the original file from which the contents of a [<file>](#) element has been extracted.

Value description:

Text.

Default value:

Undefined.

Used in:

[<file>](#).

phase-name

Phase Name - The `phase-name` attribute allows to name uniquely a [<phase>](#) element. It is also used in other elements in the file to refer to the given [<phase>](#) element.

Value description:

Text.

Default value:

Undefined.

Used in:

[<phase>](#), [<trans-unit>](#), [<target>](#), [<bin-unit>](#), [<bin-target>](#).

pos

Position - Indicates whether an isolated tag [<it>](#) is a beginning or and ending tag.

Value description:

open or close.

Default value:

Undefined.

Used in:

[<it>](#).

priority

Priority - The priority of a [<note>](#) element.

Value description:

A number between 1 and 10, 1 being the highest priority.

Default value:

1

Used in:

[<note>](#).

process-name

Process name - The name specifying the type of process a given [<phase>](#) corresponds (e.g. Translation, Proofreading, Sizing, etc.).

Value description:

Text.

Default value:

Undefined.

Used in:

[<phase>](#).

product-name

Product name - The name of the product which uses this file.

Value description:

Text.

Default value:

Undefined.

Used in:

[<file>](#).

product-version

Product version - The version of the product which uses this file.

Value description:

Alpha-numeric.

Default value:

Undefined.

Used in:

[<file>](#).

prop-type

Property type - The `prop-type` attribute specifies the type of a [<prop>](#) element.

Value description:

Text. No value defined by the standard.

Default value:

Undefined.

Used in:

[<prop>](#).

reformat

Reformat - Indicates whether the target can be formatted (size, font, etc.) different than the source.

Value description:

Boolean: `yes` or `no`.

Default value:

yes.

Used in:

[<trans-unit>](#), [<bin-unit>](#).

resname

Resource name - Resource name or identifier of a item. For example: the key in the key/value pair in a Java properties file, the ID of a string in a Windows string table, the index value of an entry in a database table, etc.

Value description:

Text.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<alt-trans>](#), [<target>](#), [<bin-unit>](#), [<bin-target>](#).

restype

Resource type - Indicates the type of the container element.

Value description:

Text. The recommended values for the `restype` attribute are as follow (this list is not exhaustive):

- `button` = Button in UI.
- `caption` = Title in UI, caption in documentation, alternate text, etc.
- `checkbox` = Check box in UI.
- `cell` = Text in a table cell.
- `dialog` = Dialog box in UI.
- `file` = Filename, path.
- `fn` = Footnote.
- `footer` = Footer text.
- `font` = Font name.
- `frame` = Frame or window, or any generic group of components.
- `header` = Header text.
- `heading` = Title or header-type segment.
- `keywords` = List of keywords, enumeration within a paragraph, etc.
- `label` = Static text, label in UI, etc.
- `listitem` = Paragraph in a list, entry in a list box, etc.
- `menu` = Menu.
- `menuitem` = Entry in a UI menu.

- `message` = Prompt, error or warning message.
- `radio` = Radio button in UI.
- `shortcut` = Windows accelerators, shortcuts in resource or property files.
- `string` = Generic text from source code, string table, etc.
- `var` = Variable.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<target>](#), [<alt-trans>](#), [<bin-unit>](#), [<bin-target>](#).

rid

Reference identifier - The `rid` attribute is used to link different elements that are related. For example, a reference to its definition, or paragraphs belonging to the same group, etc.

Value description:

Alpha-numeric without spaces.

Default value:

Undefined.

Used in:

[<bpt>](#), [<ept>](#), [<it>](#), [<bx/>](#), [<ex/>](#).

size-unit

Unit of size attributes - The `size-unit` attribute specifies the units of measure used in the [maxheight](#), [minheight](#), [maxwidth](#), and [minwidth](#) attributes.

Value description:

Text. The recommended values for the `size-unit` attribute are as follow (this list is not exhaustive):

- `pixel` = Pixel.
- `byte` = 8-bit byte.
- `char` = Unicode character.

Default value:

`pixel`.

Used in:

[<trans-unit>](#).

source-language

Source language - The language for the [<source>](#) elements in the given [<file>](#) element.

Value description:

A language code as described in the [\[RFC 3066\]](#). The values for this attribute follow the same rules as the values for [xml:lang](#). Unlike the other XLIFF attributes, the values for [xml:lang](#) are not case-sensitive. For more information see [the section on xml:lang in the XML specification](#), and the [erratum E11](#) (which replaces RFC 1766 by RFC 3066).

Default value:

Undefined.

Used in:

[<file>](#).

state

State - The status of a particular translation in a [<target>](#) or [<bin-target>](#) element.

Value description:

Text. The recommended values for the `state` attribute are as follow (this list is not exhaustive):

- `needs-translation` = The item needs to be translated.
- `needs-review` = The item needs to be reviewed.
- `needs-resizing` = The item needs to be resized or reformatted.

Default value:

Undefined.

Used in:

[<target>](#), [<bin-target>](#).

style

Style - The resource style of a control. For example, in Windows resources it corresponds to the STYLE statement.

Value description:

Text.

Default value:

Undefined.

Used in:

[<group>](#), [<trans-unit>](#), [<target>](#), [<alt-trans>](#).

target-language

Target language - The language for the [<target>](#) elements in the given [<file>](#) element.

Value description:

A language code as described in the [\[RFC 3066\]](#). The values for this attribute follow the same rules as the values for [xml:lang](#). Unlike the other XLIFF attributes, the values for `xml:lang` are not case-sensitive. For more information see [the section on xml:lang in the XML specification](#), and the [erratum E11](#) (which replaces RFC 1766 by RFC 3066).

Default value:

Undefined.

Used in:

[<file>](#).

tool

Creation tool - The `tool` attribute is used to specify the signature and version of the tool that created or modified the document.

Value description:

Text

Default value:

manual.

Used in:

[<file>](#), [<phase>](#), [<alt-trans>](#).

translate

Translate - Indicates whether or not the text referred to should be translated.

Value description:

Boolean: `yes` or `no`.

Default value:

`yes`.

Used in:

[<trans-unit>](#), [<bin-unit>](#).

ts

Tool-specific data - The `ts` attribute allows you to include short data understood by a specific toolset. You can also use the [<prop>](#) element to define large properties at the element level.

Value description:

Text. No value defined by the standard.

Default value:

Undefined.

Used in:

[<file>](#), [<group>](#), [<trans-unit>](#), [<source>](#), [<target>](#), [<bin-unit>](#), [<bin-source>](#), [<bin-target>](#), [<alt-trans>](#), [<mrk>](#), [<q>](#), [<x/>](#), [<bx/>](#), [<ex/>](#), [<bpt>](#), [<ept>](#), [<ph>](#), [<it>](#).

uid

Unique ID - The unique id used to identify the skeleton file.

Value description:

Text.

Default value:

Undefined.

Used in:

[<external-file>](#).

unit

Unit - The units counted in a [<count>](#) element.

Value description:

Text. The recommended values for the `unit` attribute are as follow (this list is not exhaustive):

- word = standard words.
- page = pages.
- trans-unit = number of [<trans-unit>](#).
- bin-unit = number of [<bin-unit>](#).
- item = number of [<trans-unit>](#) and [<bin-unit>](#).

Default value:
Undefined.

Used in:
[<count>](#).

version

XLIFF version - The `version` attribute is used to specify the format version of the XLIFF document.

Value description:
Fixed text.

Default value:
1.0

Used in:
[<xliff>](#).

3.3.2. XML Namespace Attributes

xml:lang

Language - The `xml:lang` attribute specifies the locale of the text of a given element.

Value description:

A language code as described in the [\[RFC 3066\]](#). This declared value is considered to apply to all elements within the content of the element where it is specified, unless overridden with another instance of the `xml:lang` attribute. Unlike the other XLIFF attributes, the values for `xml:lang` are not case-sensitive. For more information see [the section on `xml:lang` in the XML specification](#), and the [erratum E11](#) (which replaces RFC 1766 by RFC 3066).

Default value:
Undefined.

Used in:
[<xliff>](#), [<note>](#), [<prop>](#), [<source>](#), [<target>](#), [<alt-trans>](#).

xml:space

White spaces - The `xml:space` attribute specifies how white spaces (ASCII spaces, tabs and line-breaks) should be treated.

Value description:

`default` or `preserve`. The value `default` signals that applications' default white-space processing modes are acceptable for this element; the value `preserve` indicates the intent that applications preserve all the white space. This declared intent is considered to apply to all elements within the content of the element where it is specified, unless overridden with another instance of the `xml:space` attribute.

For more information [see the section on `xml:space` in the XML specification](#).

Default value:

`default`.

Used in:

[<file>](#), [<group>](#), [<trans-unit>](#), [<alt-trans>](#).

A. XLIFF Tree structure

The following figure shows the possible structure as a tree. Each element is followed by notation indicating its possible occurrence according to the corresponding legend.

```
(legend: 1 = one
         + = one or more
         ? = zero or one
         * = zero, one or more)

<xliff>1
|
+---- <file>+
|
+---- <header>1
|   |
|   +---- <skl>?
|   |   |
|   |   +---- (<internal-file> | <external-file>)1
|   |
|   +---- <phase-group>?
|   |   |
|   |   +---- <phase>+
|   |   |   |
|   |   |   +---- <note>*
|   |
|   +---- <glossary>*
|   |   |
|   |   +---- (<internal-file> | <external-file>)1
|   |
|   +---- <reference>*
```

```

|   |   |
|   |   +---- (<internal-file> | <external-file>)1
|   |
|   | +---- <note>*
|   |
|   | +---- <count-group>*
|   |   |
|   |   +---- <count>*
|   |
|   | +---- <prop-group>*
|   |   |
|   |   +---- <prop>*
|   |
+---- <body>1
|
+---- <group>*
|
|   +---- <context-group>*
|   |
|   |   +---- <context>+
|   |
|   | +---- <count-group>*
|   |
|   | +---- <prop-group>*
|   |
|   | +---- <note>*
|   |
|   | +---- <group>*
|   |
|   | +---- <trans-unit>*
|   |
|   | +---- <bin-unit>*
|   |
+---- <trans-unit>*
|
+---- <source>1 & <target>?
|
|   +---- <ph>* or <it>*
|   |   or <x/>* or <bx/>*
|   |   or <ex/>* or <mrk>* or
|   |   or <bpt>* or <ept>*
|   |
|   |   +---- <sub>*
|   |   |
|   |   |   +---- <bpt>* or <ept>* or <ph>* or <it>*
|   |   |   |   or <g>* or <mrk>* or <x/>* or <bx/>*
|   |   |   |   or <ex/>*
|   |   |
|   |   |   or <g>*
|   |   |   |
|   |   |   |   +---- <bpt>* or <ept>* or <ph>* or <it>* or <g>*
|   |   |   |   |   or <mrk>* or <x/>* or <bx/>* or <ex/>*
|   |   |
|   |   +---- <context-group>*
|   |
|   | +---- <count-group>*
|   |
|   | +---- <prop-group>*
|   |
|   | +---- <note>*

```



```

|
| +---- <alt-trans>*
|     |
|     +---- <context-group>*
|     |
|     +---- <source>? & <target>+
|     |
|     +---- <note>*
|     |
|     +---- <prop-group>*
|
+---- <bin-unit>*
|
+---- <bin-source>1 & <bin-target>?
|     |
|     +---- (<internal-file> | <external-file>)1
|
+---- <context-group>*
|
+---- <count-group>*
|
+---- <prop-group>*
|
+---- <note>*
|
+---- <trans-unit>*

```

B. Document Type Definition for XLIFF

The document type definition file for XLIFF is available at: <http://www.oasis-open.org/committees/xliff/documents/xliff.dtd>.

C. Changes Since Previous Version (Non-Normative)

The changes in this version relative to the previous version are as follows:

- Updated the link to ISO 3166 standard.
- Fixed the Status of this Document paragraph.

D. XLIFF Technical Committee (Non-Normative)

The XLIFF Technical Committee at OASIS is composed of the following members:

- Jonathan Clark, Lionbridge - (Vice-Chair)
- John Corrigan, Sun Microsystems
- Ian Dunlop, Novell
- Fiona Ebbs, Novell
- Tony Jewtushenko, Oracle - (Chair)
- Milan Karásek, Moravia-IT

- Caroline Koff, HP
- David Leland
- Mark Levins, IBM
- Christian Lieske, SAP
- Mat Lovatt, Oracle
- Ultan Ó Broin, Oracle
- Enda McDonnell
- Mike McKenna, CommerceOne
- David Ramsey, Xerox
- John Reid, Novell
- Peter Reynolds - (Secretary)
- François Richard, Xerox
- Yves Savourel, RWS Group - (Editor)
- Reinhard Schäler

E. References

Normative

[IANA Charsets]

[*IANA Names for Character Sets*](#). IANA (Internet Assigned Numbers Authority), Aug 2001

[ISO 639]

[*Codes for the Representation of Names of Languages*](#). ISO (International Standards Organization), Nov 2001.

[ISO 3166]

[*Codes for the representation of names of countries and their subdivisions*](#). ISO (International Organization for Standardization), Jun 2000.

[ISO 8601]

[*Representation of dates and times*](#). ISO (International Organization for Standardization), Dec 2000.

[OpenTag 1.2]

[*OpenTag Format Specifications*](#). ILE (International Language Engineering), Nov 1998.

[RFC 1341]

[*Multipurpose Internet Mail Extensions*](#). IETF (Internet Engineering Task Force), Jun 1992.

[RFC 3066]

[*RFC 3066 Tags for the Identification of Languages*](#). IETF (Internet Engineering Task Force), Jan 2001.

[TMX 1.3]

[*TMX Format Specifications*](#). LISA (Localisation Industry Standards association), Aug 2001.

[XML 1.0]

[*Extensible Markup Language \(XML\) 1.0 Second Edition*](#). W3C (World Wide Web Consortium), Oct 2000.

Non-Normative

[ISO]

[International Organization for Standardization](#) Web site.

[LISA]

[Localisation Industry Standards Association](#) Web site.

[OASIS]

[Organization for the Advancement of Structured Information Standards](#) Web site.

[Unicode]

[Unicode Consortium](#) Web site.

[W3C]

[World Wide Web Consortium](#) Web site.