



Internationalization Tag Set (ITS) Version 2.0

W3C Working Draft 06 December 2012

This version:

<http://www.w3.org/TR/2012/WD-its20-20121206/>

Latest version:

<http://www.w3.org/TR/its20/>

Previous version:

<http://www.w3.org/TR/2012/WD-its20-20121023/>

Editors:

Shaun McCane, Invited Expert
Dave Lewis, TCD
Arle Lommel, DFKI
Jirka Kosek, UEP
Felix Sasaki, DFKI / W3C Fellow
Yves Savourel, ENLASO

This document is also available in these non-normative formats: [ODD/XML document](#), [self-contained zipped archive](#), and [XHTML Diff markup to previous publication 2012-10-23](#).

[Copyright](#) © 2012 [W3C](#)[®] ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

This document defines data categories and their implementation as a set of elements and attributes called the *Internationalization Tag Set (ITS)* 2.0. ITS 2.0 is the successor of [ITS 1.0](#); it is designed to foster the creation of multilingual Web content, focusing on HTML, XML based formats in general, and to leverage localization workflows based on the XML Localization Interchange File Format (XLIFF).

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <http://www.w3.org/TR/>.

This document defines data categories and their implementation as a set of elements and attributes called the *Internationalization Tag Set (ITS)* 2.0. ITS 2.0 is the successor of [ITS 1.0](#); it is designed to foster the creation of multilingual Web content, focusing on HTML, XML based formats in general, and to leverage localization workflows based on the XML Localization Interchange File Format (XLIFF).

This document was published by the [MultilingualWeb-LT Working Group](#) as a Last Call Working Draft. The Working Group expects to advance this Working Draft to Recommendation status (see [W3C document maturity levels](#)). The Last Call period ends 10 January 2013.

The normative sections of this document (from [Section 3: Notation and Terminology](#) to [Section 8: Description of Data Categories](#) and [Appendix A: References](#) to [Appendix D: Schemas for ITS](#)) are stable. The other, non-normative sections contain only explanatory material and will be updated in a later working draft. Hence, the Working Group especially encourages feedback on the normative sections. The goal is to move out of last call without any substantive changes to these sections.

To give feedback send your comments to public-multilingualweb-lt-comments@w3.org. Use "Comment on ITS 2.0 specification WD" in the subject line of your email. The [archives for this list](#) are publicly available. See also [issues discussed within the Working Group](#) and the [list of changes since the previous publication](#).

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). W3C maintains a [public list of any patent disclosures](#) made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains [Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

Table of Contents

- 1 [Introduction](#)
 - 1.1 [Relation to ITS 1.0 and New Principles](#)
 - 1.1.1 [Relation to ITS 1.0](#)
 - 1.1.2 [New Principles](#)
 - 1.2 [Motivation for ITS](#)
 - 1.2.1 [Typical Problems](#)
 - 1.3 [Users and Usages of ITS](#)
 - 1.3.1 [Potential Users of ITS](#)
 - 1.3.2 [Ways to Use ITS](#)
 - 1.4 [Usage in HTML](#)
 - 1.4.1 [Support for legacy HTML content](#)
 - 1.5 [Out of Scope](#)
 - 1.6 [Important Design Principles](#)
- 2 [Basic Concepts](#)
 - 2.1 [Selection](#)
 - 2.1.1 [Local Approach](#)
 - 2.1.2 [Global Approach](#)
 - 2.2 [Overriding and Inheritance](#)
 - 2.3 [Adding Information or Pointing to Existing Information](#)
- 3 [Notation and Terminology](#)
 - 3.1 [Notation](#)
 - 3.2 [Data category](#)
 - 3.3 [Selection](#)
 - 3.4 [ITS Local Attributes](#)
 - 3.5 [Rule Elements](#)
 - 3.6 [Usage of Internationalized Resource Identifiers in ITS](#)
 - 3.7 [The Term HTML](#)
- 4 [Conformance](#)
 - 4.1 [Conformance Type 1: ITS Markup Declarations](#)
 - 4.2 [Conformance Type 2: The Processing Expectations for ITS Markup](#)
 - 4.3 [Conformance Type 3: Processing Expectations for ITS Markup in HTML](#)

- 4.4 [Conformance Class for HTML5+ITS documents](#)
- 5 [Processing of ITS information](#)
 - 5.1 [Indicating the Version of ITS](#)
 - 5.2 [Locations of Data Categories](#)
 - 5.2.1 [Global, Rule-based Selection](#)
 - 5.2.2 [Local Selection in an XML Document](#)
 - 5.3 [Query Language of Selectors](#)
 - 5.3.1 [Choosing Query Language](#)
 - 5.3.2 [XPath 1.0](#)
 - 5.3.3 [CSS Selectors](#)
 - 5.3.4 [Additional query languages](#)
 - 5.3.5 [Variables in selectors](#)
 - 5.4 [Link to External Rules](#)
 - 5.5 [Precedence between Selections](#)
 - 5.6 [Associating ITS Data Categories with Existing Markup](#)
 - 5.7 [Conversion to NIF](#)
 - 5.8 [ITS Tools Annotation](#)
- 6 [Using ITS Markup in HTML](#)
 - 6.1 [Mapping of Local Data Categories to HTML](#)
 - 6.2 [Global rules](#)
 - 6.3 [Standoff Markup in HTML](#)
 - 6.4 [Precedence between Selections](#)
- 7 [Using ITS Markup in XHTML](#)
- 8 [Description of Data Categories](#)
 - 8.1 [Position, Defaults, Inheritance and Overriding of Data Categories](#)
 - 8.2 [Translate](#)
 - 8.2.1 [Definition](#)
 - 8.2.2 [Implementation](#)
 - 8.3 [Localization Note](#)
 - 8.3.1 [Definition](#)
 - 8.3.2 [Implementation](#)
 - 8.4 [Terminology](#)
 - 8.4.1 [Definition](#)
 - 8.4.2 [Implementation](#)
 - 8.5 [Directionality](#)
 - 8.5.1 [Definition](#)
 - 8.5.2 [Implementation](#)
 - 8.6 [Ruby](#)
 - 8.6.1 [Definition](#)
 - 8.6.2 [Implementation](#)
 - 8.7 [Language Information](#)
 - 8.7.1 [Definition](#)
 - 8.7.2 [Implementation](#)
 - 8.8 [Elements Within Text](#)
 - 8.8.1 [Definition](#)
 - 8.8.2 [Implementation](#)
 - 8.9 [Domain](#)
 - 8.9.1 [Definition](#)
 - 8.9.2 [Implementation](#)
 - 8.10 [Disambiguation](#)
 - 8.10.1 [Definition](#)
 - 8.10.2 [Implementation](#)
 - 8.11 [Locale Filter](#)
 - 8.11.1 [Definition](#)

- 8.11.2 [Implementation](#)
- 8.12 [Provenance](#)
 - 8.12.1 [Definition](#)
 - 8.12.2 [Implementation](#)
- 8.13 [External Resource](#)
 - 8.13.1 [Definition](#)
 - 8.13.2 [Implementation](#)
- 8.14 [Target Pointer](#)
 - 8.14.1 [Definition](#)
 - 8.14.2 [Implementation](#)
- 8.15 [Id Value](#)
 - 8.15.1 [Definition](#)
 - 8.15.2 [Implementation](#)
- 8.16 [Preserve Space](#)
 - 8.16.1 [Definition](#)
 - 8.16.2 [Implementation](#)
- 8.17 [Localization Quality Issue](#)
 - 8.17.1 [Definition](#)
 - 8.17.2 [Implementation](#)
- 8.18 [Localization Quality Rating](#)
 - 8.18.1 [Definition](#)
 - 8.18.2 [Implementation](#)
- 8.19 [MT Confidence](#)
 - 8.19.1 [Definition](#)
 - 8.19.2 [Implementation](#)
- 8.20 [Allowed Characters](#)
 - 8.20.1 [Definition](#)
 - 8.20.2 [Implementation](#)
- 8.21 [Storage Size](#)
 - 8.21.1 [Definition](#)
 - 8.21.2 [Implementation](#)

Appendices

- A [References](#)
 - B [Internationalization Tag Set \(ITS\) MIME Type](#)
 - C [Values for the Localization Quality Issue Type](#)
 - D [Schemas for ITS](#)
 - E [References](#) (Non-Normative)
 - F [Checking ITS Markup Constraints With Schematron](#) (Non-Normative)
 - G [Conversion NIF2ITS](#) (Non-Normative)
 - H [List of ITS 2.0 Global Elements and Local Attributes](#) (Non-Normative)
 - I [Revision Log](#) (Non-Normative)
 - J [Acknowledgements](#) (Non-Normative)
-

1 Introduction



This section is informative.

ITS 2.0 is a technology to add metadata to Web content, for the benefit of localization, language technologies, and internationalization. The ITS 2.0 specification both identifies concepts (such as

“Translate”) that are important for internationalization and localization, and defines implementations of these concepts (termed “ITS data categories”) as a set of elements and attributes called the *Internationalization Tag Set (ITS)*. The document provides implementations for HTML, serializations in [NIF](#), and provides definitions of ITS elements and attributes in the form of XML Schema [\[XML Schema\]](#) and RELAX NG [\[RELAX NG\]](#).

This document aims to realize many of the ideas formulated in the [ITS 2.0 Requirements document](#), in [\[ITS REQ\]](#) and [\[Localizable DTDs\]](#).

Not all requirements listed there are addressed in this document. Those which are not addressed here are either covered in [\[XML i18n BP\]](#) (potentially in an as yet unwritten best practice document on multilingual Web content), or may be addressed in a future version of this specification.

1.1 Relation to ITS 1.0 and New Principles

1.1.1 Relation to ITS 1.0

ITS 2.0 has the following relations to ITS 1.0:

- It adopts and maintains the following principles from ITS 1.0:
 - It adopts the use of data categories to define discrete units of functionality
 - It adopts the separation of data category definition from the mapping of the data category to a given content format
 - It adopts the conformance principle of ITS1.0 that an implementation only needs to implement one data category to claim conformance to ITS 2.0
- ITS 2.0 supports all ITS 1.0 data category definitions and adds new definitions, with the exceptions of [Directionality](#) and [Ruby](#).
- ITS 2.0 adds a number of new data categories not found in ITS 1.0.
- While ITS 1.0 addressed only XML, ITS 2.0 specifies implementations of data categories in *both XML and HTML*.

1.1.2 New Principles

ITS 2.0 also adds the following principles and features not found in ITS 1.0:

- ITS 2.0 data categories are intended to be format neutral, with support for XML, HTML, and NIF: a data category implementation only needs to support a single content format mapping in order to support a claim of ITS 2.0 conformance.
- ITS 2.0 provides algorithms to generate NIF out of HTML or XML with ITS 2.0 metadata.
- A global implementation of ITS 2.0 requires at least the XPath version 1.0. Other versions of XPath or other query languages (e.g., CSS selectors) can be expressed via a dedicated [queryLanguage](#) attribute.

The new data categories included in ITS 2.0 are:

- [Domain](#)
- [Disambiguation](#)
- [Locale Filter](#)
- [Provenance](#)
- [External Resource](#)
- [Target Pointer](#)
- [Id Value](#)
- [Preserve Space](#)

- [Localization Quality Issue](#)
- [Localization Quality Rating](#)
- [MT Confidence](#)
- [Allowed Characters](#)
- [Storage Size](#)

1.2 Motivation for ITS

Content or software that is authored in one language (the **source language**) is often made available in additional languages or adapted with regard to other cultural aspects. This is done through a process called **localization**, where the original material is translated and adapted to the target audience.

In addition, document formats expressed by schemas may be used by people in different parts of the world, and these people may need special markup to support the local language or script. For example, people authoring in languages such as Arabic, Hebrew, Persian, or Urdu need special markup to specify directionality in mixed direction text.

From the viewpoints of feasibility, cost, and efficiency, it is important that the original material should be suitable for localization. This is achieved by appropriate design and development, and the corresponding process is referred to as internationalization. For a detailed explanation of the terms “localization” and “internationalization”, see [\[I10n i18n\]](#).

[Ed. note: Note: This should refer to the best practice document as well, when ready.]

The increasing usage of XML as a medium for documentation-related content (e.g. DocBook and DITA as formats for writing structured documentation, well suited to computer hardware and software manuals) and software-related content (e.g. the eXtensible User Interface Language [\[XUL\]](#)) creates challenges and opportunities in the domain of XML internationalization and localization.

1.2.1 Typical Problems

The following examples sketch one of the issues that currently hinder efficient XML-related localization: the lack of a standard, declarative mechanism that identifies which parts of an XML document need to be translated. Tools often cannot automatically perform this identification.

Example 1: Document with partially translatable content

In this document it is difficult to distinguish between those `string` elements that are translatable and those that are not. Only the addition of an explicit flag could resolve the issue.

```
<resources>
  <section id="Homepage">
    <arguments>
      <string>page</string>
      <string>childlist</string>
    </arguments>
    <variables>
      <string>POLICY</string>
      <string>Corporate Policy</string>
    </variables>
    <keyvalue_pairs>
      <string>Page</string>
      <string>ABC Corporation - Policy Repository</string>
      <string>Footer_Last</string>
    </keyvalue_pairs>
  </section>
</resources>
```

```

    <string>Pages</string>
    <string>bgColor</string>
    <string>NavajoWhite</string>
    <string>title</string>
    <string>List of Available Policies</string>
  </keyvalue_pairs>
</section>
</resources>

```

[Source file: [examples/xml/EX-motivation-its-1.xml](#)]

Example 2: Document with partially translatable content

Even when metadata are available to identify non-translatable text, the conditions may be quite complex and not directly indicated with a simple flag. Here, for instance, only the text in the nodes matching the expression

`//component[@type!='image']/data[@type='text']` is translatable.

```

<dialogue xml:lang="en-gb">
  <rsrc id="123">
    <component id="456" type="image">
      <data type="text">images/cancel.gif</data>
      <data type="coordinates">12,20,50,14</data>
    </component>
    <component id="789" type="caption">
      <data type="text">Cancel</data>
      <data type="coordinates">12,34,50,14</data>
    </component>
    <component id="792" type="string">
      <data type="text">Number of files: </data>
    </component>
  </rsrc>
</dialogue>

```

[Source file: [examples/xml/EX-motivation-its-2.xml](#)]

1.3 Users and Usages of ITS

1.3.1 Potential Users of ITS

The ITS specification aims to provide different types of users with information about what markup should be supported to enable worldwide use and effective internationalization and localization of content. The following paragraphs sketch these different types of users, and their usage of ITS. In order to support all of these users, the information about what markup should be supported to enable worldwide use and effective localization of content is provided in this specification in two ways:

- abstractly in the data category descriptions: [Section 8: Description of Data Categories](#)
- concretely in the ITS schemas: [Appendix D: Schemas for ITS](#)

1.3.1.1 Schema developers starting a schema from the ground up

This type of user will find proposals for attribute and element names to be included in their new schema (also called "host vocabulary"). Using the attribute and element names proposed in the ITS specification may be helpful because it leads to easier recognition of the concepts represented by

both schema users and processors. It is perfectly possible, however, for a schema developer to develop his own set of attribute and element names. The specification sets out, first and foremost, to ensure that the required markup is available, and that the behavior of that markup meets established needs.

1.3.1.2 Schema developers working with an existing schema

This type of user will be working with schemas such as DocBook, DITA, or perhaps a proprietary schema. The ITS Working Group has sought input from experts developing widely used formats such as the ones mentioned.

Note:

The question "How to use ITS with existing popular markup schemes?" is covered in more details (including examples) in a separate document: [\[XML i18n BP\]](#).

Developers working on existing schemas should check whether their schemas support the markup proposed in this specification, and, where appropriate, add the markup proposed here to their schema.

In some cases, an existing schema may already contain markup equivalent to that recommended in ITS. In this case it is not necessary to add duplicate markup since ITS provides mechanisms for associating ITS markup with markup in the host vocabulary which serves a similar purpose (see [Section 5.6: Associating ITS Data Categories with Existing Markup](#)). The developer should, however, check that the behavior associated with the markup in their own schema is fully compatible with the expectations described in this specification.

1.3.1.3 Vendors of content-related tools

This type of user includes companies which provide tools for authoring, translation or other flavors of content-related software solutions. It is important to ensure that such tools enable worldwide use and effective localization of content. For example, translation tools should prevent content marked up as not for translation from being changed or translated. It is hoped that the ITS specification will make the job of vendors easier by standardizing the format and processing expectations of certain relevant markup items, and allowing them to more effectively identify how content should be handled.

1.3.1.4 Content producers

This type of user comprises authors, translators and other types of content author. The markup proposed in this specification may be used by them to mark up specific bits of content. Aside: The burden of inserting markup can be removed from content producers by relating the ITS information to relevant bits of content in a global manner (see [global, rule-based approach](#)). This global work, however, may fall to information architects, rather than the content producers themselves.

Content producers often work with content management systems (CMS). In various CMS, some of the CMS fields only allow to store plain text. For these fields, the current ITS 2.0 data categories can only be applied globally and not with local attributes. This issue should be addressed in another way, apart from the ITS 2.0 standard. One way would be to allow HTML in these fields if possible, or using an extra field which allows HTML input and save the plain text of this extra field in the plain text field.

1.3.1.5 Machine Translation Systems

This type of service is intended for a broad user community ranging from developers and integrators through translation companies and agencies, freelance translators and post-editors to ordinary translation consumers and other types of MT employment. Data categories are envisaged for

supporting and guiding the different automated backend processes of this service type, thereby adding substantial value to the service results as well as possible subsequent services. These processes include basic tasks, like parsing constraints and markup, and compositional tasks, such as disambiguation. These tasks consume and generate valuable metadata from and for third party users, for example, provenance information and quality scoring, and add relevant information for follow-on tasks, processes and services, such as MT post-editing, MT training and MT terminological enhancement.

1.3.1.6 Text Analytics

These types of users fulfil the role of providing services for automatic generation of metadata for improving localization, data integration or knowledge management workflows. This class of users comprises of developers and integrators of services that automate language technology tasks such as domain classification, named entity recognition and disambiguation, term extraction, language identification and others. Text analytics services generate data that contextualizes the raw content with more explicit information. This can be used to improve the output quality in machine translation systems, search result relevance in information retrieval systems, as well as management and integration of unstructured data in knowledge management systems.

1.3.1.7 Localization Workflow Managers

This type of users is concerned with localization workflows in which content goes through certain steps: preparation for localization, start of the localization process by e.g. a conversion into a bitext format like [XLIFF](#), the actual localization by human translators or machine translation and other adaptations of content, and finally the integration of the localized content into the original format. That format is often based on XML or HTML; (Web) content management systems are widely used for content creation, and their integration with localization workflows is an important task for the workflow manager. For the integration of content creation and localization, metadata plays a crucial role. E.g. an ITS data category like [translate](#) can trigger the extraction of localizable text. "*Metadata roundtripping*", that is the availability of metadata both before and after the localization process is crucial for many tasks of the localization workflow manager. An example is metadata based quality control, with checks like "*Have all pieces of content set to `translate='no'` been left unchanged?*". Other pieces of metadata are relevant for proper internationalization during the localization workflow, e.g. the availability of [Directionality](#) markup for adequate visualization of bidirectional text.

1.3.2 Ways to Use ITS



The ITS specification proposes several mechanisms for supporting worldwide use and effective internationalization and localization of content. We will sketch them below by looking at them from the perspectives of certain user types. For the purpose of illustration, we will demonstrate how ITS can indicate that certain parts of content should or should not be translated.

- A content author uses an attribute on a particular element to say that the text in the element should not be translated.

Example 3: Use of ITS by content author

The `its:translate="no"` attributes indicate that the `path` and the `cmd` elements should not be translated.

```
<?xml version="1.0" encoding="UTF-8" ?>
<help xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <head>
    <title>Building the Zebulon Toolkit</title>
  </head>
  <body>
    <p>To re-compile all the modules of the Zebulon toolkit you need to go in the

```

```

        its:translate="no">\Zebulon\Current Source\binary</path> directory. Then
        batch file <cmd its:translate="no">Build.bat</cmd>.</p>
    </body>
</help>

```

[Source file: <examples/xml/EX-ways-to-use-its-1.xml>]

- A content author or information architect uses markup at the top of the document to identify a particular type of element or context in which the content should not be translated.

Example 4: Use of ITS by information architect

The `translateRule` element is used in the header of the document to indicate that none of the `path` or `cmd` elements should be translated.

```

<help xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <head>
    <title>Building the Zebulon Toolkit</title>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:translateRule selector="//path | //cmd" translate="no"/>
    </its:rules>
  </head>
  <body>
    <p>To re-compile all the modules of the Zebulon toolkit you need to go in the
      <path>\Zebulon\Current Source\binary</path> directory. Then from there, r
      <cmd>Build.bat</cmd>.</p>
  </body>
</help>

```

[Source file: <examples/xml/EX-ways-to-use-its-2.xml>]

- A processor may insert markup at the top of the document which links to ITS information outside of the document.

Example 5: Use of ITS by processor

A `rules` element is inserted in the header of the document. It has a XLink `href` attribute used to link to an [ITS external rule](#) document.

```

<help xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <head>
    <title>Building the Zebulon Toolkit</title>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
      xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="EX-ways-to-use-its-4
    </head>
  <body>
    <p>To re-compile all the modules of the Zebulon toolkit you need to go in the
      <path>\Zebulon\Current Source\binary</path> directory. Then from there, r
      <cmd>Build.bat</cmd>.</p>
  </body>
</help>

```

[Source file: <examples/xml/EX-ways-to-use-its-3.xml>]

Example 6: ITS rule file shared by different documents

The `rules` element contains several ITS rules that are common to different documents. One of them is a `translateRule` element that indicates that no `path` or `cmd` element should be translated.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
  <its:translateRule selector="//path | //cmd" translate="no"/>
</its:rules>
```

[Source file: <examples/xml/EX-ways-to-use-its-4.xml>]

- A schema developer integrates ITS markup declarations in his schema to allow users to indicate that specific parts of the content should not be translated.

[Ed. note: Following schema example has to be updated once we have final XSD schema for ITS 2.0]

Example 7: An XSD schema with ITS declaration

The declarations for the `translate` attribute is added to a group of common attributes `commonAtts`. This allows to use the `translate` attribute within the documents like in [Example 3](#).

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:its="http://www.w3.org/2005/11/its"
  elementFormDefault="qualified">
  <xs:import namespace="http://www.w3.org/2005/11/its" schemaLocation="its.xsd"/>
  <xs:attributeGroup name="commonAtts">
    <xs:attributeGroup ref="its:att.local.with-ns.attribute.translate"/>
    <xs:attribute name="id" type="xs:ID" use="optional"/>
  </xs:attributeGroup>
  <xs:element name="help">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="head">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title" type="xs:string"/>
            </xs:sequence>
            <xs:attributeGroup ref="commonAtts"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="body">
          <xs:complexType>
            <xs:choice minOccurs="1" maxOccurs="unbounded">
              <xs:element name="p">
                <xs:complexType mixed="true">
                  <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:element ref="path"/>
                    <xs:element ref="cmd"/>
                  </xs:choice>
                  <xs:attributeGroup ref="commonAtts"/>
                </xs:complexType>
              </xs:element>
            </xs:choice>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="its:att.version.attribute.version"/>
    </xs:complexType>
  </xs:element>
```

```

<xs:element name="path">
  <xs:complexType mixed="true">
    <xs:attributeGroup ref="commonAtts"/>
  </xs:complexType>
</xs:element>
<xs:element name="cmd">
  <xs:complexType mixed="true">
    <xs:attributeGroup ref="commonAtts"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```

[Source file: [examples/xml/EX-ways-to-use-its-5.xsd](#)]

The first two approaches above can be likened to the use of CSS in [\[XHTML 1.0\]](#). Using a `style` attribute, an XHTML content author may assign a color to a particular paragraph. That author could also have used the `style` element at the top of the page to say that all paragraphs of a particular class or in a particular context would be colored red.

1.4 Usage in HTML



ITS 2.0 adds support for usage in HTML. In HTML, ITS local selection is realized via dedicated, [data category specific attributes](#).

[Ed. note: Add example of HTML with local attributes for illustration purposes]

For the so-called “[global approach](#)” in HTML, this specification defines a link type for referring to files with global rules in [Section 6.2: Global rules](#).

Example 8: Using ITS global rules in HTML

The `link` element points to the rules file `EX-translateRule-html5-1.xml`. The `rel` attribute identifies the ITS specific link relation `its-rules`.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Translate flag global rules example</title>
    <link href=EX-translateRule-html5-1.xml rel=its-rules>
  </head>
  <body>
    <p>This sentence should be translated, but code names like the <code>span</code>
  </body>
</html>

```

[Source file: [examples/html5/EX-translate-html5-global-1.html](#)]

Example 9: ITS rules file linked from HTML

The rules file linked in [Example 8](#).

```

<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:h="http://www.w3.org/1999/xhtml">

```

```
<its:translateRule translate="no" selector="//h:code"/>
</its:rules>
```

[Source file: <examples/html5/EX-translateRule-html5-1.xml>]

1.4.1 Support for legacy HTML content



ITS 2.0 does not define how to use ITS in HTML versions prior version 5. Users are encouraged to migrate their content to HTML or XHTML. While it is possible to use `its-*` attributes introduced for HTML5 in older versions of HTML (such as 3.2 or 4.01) and pages using these attributes will work without any problems, `its-*` attributes will be marked as invalid in validators.

1.5 Out of Scope



The definition of what a localization process or localization parameters must address is outside the scope of this standard and it does not address all of the mechanisms or data formats (sometimes called localization project parameters) that may be needed to configure localization workflows or process specific formats. However, it does define standard data categories that may be used in defining localization workflows or processing specific formats.

Note:

“**XML localization project parameters**” is a generic term to name the mechanisms and data formats that allow localization tools to be configured in order to process a specific XML format. Examples of XML localization project parameters are the Trados “DTD Settings” file, and the SDLX “Analysis” file.

1.6 Important Design Principles



Abstraction via *data categories*: ITS defines **data categories** as an abstract notion for information needed for the internationalization and localization of XML documents and HTML documents. This abstraction is helpful in realizing independence from any one particular implementation (e.g., as an element or attribute). (See [Section 3.2: Data category](#) for a definition of the term data categories, [Section 8: Description of Data Categories](#) for the definition of the various ITS data categories, and subsections in [Section 8: Description of Data Categories](#) for the data category implementations.)

Powerful *selection mechanism*: For ITS markup that appears in an XML instance, the XML nodes to which the ITS-related information pertains must be clearly defined. Thus, ITS defines [selection mechanisms](#) to specify to what parts of an XML document an ITS data category and its values should be applied. Selection relies on the information which is given in the XML Information Set [\[XML Infoset\]](#). ITS applications may implement inclusion mechanisms such as XInclude or DITA's [\[DITA 1.0\]](#) conref.

Content authors, for example, need a simple way to work with the [Translate](#) data category in order to express whether the content of an element or attribute should be translated or not. Localization managers, on the other hand, need an efficient way to manage translations of large document sets based on the same schema. These needs could be realized by a specification of defaults for the [Translate](#) data category along with exceptions to those defaults (e.g. all `p` elements should be translated, but not `p` elements inside of an `index` element).

To meet these requirements this specification introduces mechanisms that add ITS information to XML documents, see [Section 5: Processing of ITS information](#). These mechanisms also provide a means for specifying ITS information for attributes (a task for which no standard means previously existed).

The ITS selection mechanisms allows you to provide information about content [locally](#) (specified at the XML or HTML element to which it pertains) or [globally](#) (specified in another part of the document). Global selection mechanisms can be in the same document, or in a separate file.

No dedicated extensibility: It may be useful or necessary to extend the set of information available for internationalization or localization purposes beyond what is provided by ITS. This specification does not define a dedicated extension mechanism, since ordinary XML mechanisms (e.g. XML Namespaces [\[XML Names\]](#)) may be used.

Ease of integration:

- ITS follows the example from [section 4](#) of [\[XLink 1.1\]](#), by providing mostly global attributes for the implementation of ITS data categories. Avoiding elements for ITS purposes as much as possible ensures ease of integration into existing markup schemes, see [section 3.14](#) in [\[ITS REQ\]](#). Only for some requirements do additional child elements have to be used, see for example [Section 8.6: Ruby](#).
- ITS has no dependency on technologies which are still under development.
- ITS fits with existing work in the W3C architecture (e.g. use of [\[XPath 1.0\]](#) for the selection mechanism and use of IRI's [\[RFC 3987\]](#) as references to relevant external resources).

2 Basic Concepts



This section is informative.

2.1 Selection



Information (e.g. "translate this") captured by ITS markup (e.g. `its:translate='yes'`) always pertains to one or more XML or HTML nodes (primarily element and attribute nodes). In a sense, ITS markup "selects" the relevant node(s). Selection may be explicit or implicit. ITS distinguishes two approaches to selection: (1) local, and (2) using global rules.

The mechanisms defined for ITS selection resemble those defined in [\[CSS 2.1\]](#). The local approach can be compared to the `style` attribute in HTML/XHTML, and the approach with global rules is similar to the `style` element in HTML/XHTML. ITS usually uses XPath for identifying nodes although CSS and other query languages can be used if supported by application. Thus,

- the local approach puts ITS markup in the relevant element of the host vocabulary (e.g. the `author` element in DocBook)
- the [rule-based, global approach](#) puts the ITS markup in elements defined by ITS itself (namely the `rules` element)

ITS markup can be used with XML documents (e.g. a DocBook article), or schemas (e.g. an XML Schema document for a proprietary document format).

The following two examples sketch the distinction between the local and global approaches, using the `translate` as one example of ITS markup.

2.1.1 Local Approach



The document in [Example 10](#) shows how a content author may use the ITS `translate` attribute to indicate that all content inside the `author` element should be protected from translation. Translation tools that are aware of the meaning of this attribute can then screen the relevant content from the translation process.

Example 10: ITS markup on elements in an XML document (local approach)

```

<article xmlns="http://docbook.org/ns /docbook"
         xmlns:its="http://www.w3.org/2005/11/its"
         its:version="2.0" version="5.0" xml:lang="en">
  <info>
    <title>An example article</title>
    <author its:translate="no">
      <personname>
        <firstname>John</firstname>
        <surname>Doe</surname>
      </personname>
      <affiliation>
        <address><email>foo@example.com</email></address>
      </affiliation>
    </author>
  </info>
  <para>This is a short article.</para>
</article>

```

[Source file: <examples/xml/EX-basic-concepts-1.xml>]

For this example to work, the schema developer will need to add the `translate` attribute to the schema as a common attribute or on all the relevant element definitions. Note how there is an expectation in this case that inheritance plays a part in identifying which content does have to be translated and which does not. Tools that process this content for translation will need to implement the expected inheritance.

2.1.2 Global Approach



The document in [Example 11](#) shows a different approach to identifying non-translatable content, similar to that used with a `style` element in [XHTML 1.0](#), but using an ITS-defined element called `rules`. It works as follows: A document can contain a `rules` element (placed where it does not impact the structure of the document, e.g., in a “head” section). It contains one or more ITS rule elements (for example `translateRule`). Each of these specific elements contains a `selector` attribute. As its name suggests, this attribute selects the node or nodes to which a corresponding ITS information pertains. The values of ITS selector attributes are XPath absolute location paths (or CSS selectors if `queryLanguage` is set to “css”). Information for the handling of namespaces in these path expressions is taken from namespace declarations [\[XML Names\]](#) at the current rule element.

Note:

Caveat Related to XSLT-based Processing of ITS Selector Attributes

The values of ITS `selector` attributes are XPath absolute location paths. Accordingly, the following is a legitimate value:

```
myElement/descendant-or-self::*/@*
```

Unfortunately, values like this cause trouble when they are used in XSLT-based processing of ITS where the values of the ITS `selector` attributes are used as values of `match` attributes of XSLT templates. The reason for this is the following: `match` attributes may only contain a restriction/subset of XPath expressions, so-called [patterns](#).

Basically the following restrictions hold for patterns:

- only axes “child” or “attribute” allowed

- "/" or "/" possible
- id() or key() function possible
- predicates possible

Using only XSLT patterns in ITS `selector` attributes helps to avoid this issue. In many cases, this is possible by using patterns with predicates. The value above may for example be rewritten as follows:

`*[self::myElement]/@* | myElement//*/@*`

Example 11: ITS global markup in an XML document (rule-based approach)

```
<myTopic xmlns="http://mysuri.example.com" id="topic01" xml:lang="en-us">
  <prolog>
    <title>Using ITS</title>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:translateRule selector="//n:term" translate="no" xmlns:n="http://mysu
    </its:rules>
  </prolog>
  <body>
    <p>ITS defines <term>data category</term> as an abstract concept for a partic
      information for internationalization and localization of XML schemas and do
    </body>
  </myTopic>
```

[Source file: <examples/xml/EX-basic-concepts-2.xml>]

For this approach to work, the schema developer needs to add the `rules` element and associated markup to the schema. In some cases global rules may be sufficient to allow the schema developer to avoid adding other ITS markup (such as an `translate` attribute) to the elements and attributes in the schema. However, it is likely that authors will want to use attributes on markup from time to time to override the general rule.

For specification of the [Translate](#) data category information, the contents of the `rules` element would normally be designed by an information architect familiar with the document format and familiar with, or working with someone familiar with, the needs of the localization group.

The global, rule-based approach has the following benefits:

- Content authors do not have to concern themselves with creating additional markup or verifying that the markup was applied correctly. ITS data categories are associated with sets of nodes (for example all `p` elements in an XML instance)
- Changes can be made in a single location, rather than by searching and modifying local markup throughout a document (or documents, if the `rules` element is stored as an external entity)
- ITS data categories can designate attribute values as well as elements.
- It is possible to associate ITS markup with existing markup (for example the `term` element in DITA)

The commonality in both examples above is the markup `translate='no'`. This piece of ITS markup can be interpreted as follows:

- it pertains to the [Translate](#) data category
- the attribute `translate` holds a value of "no"

The ITS `selector` attribute allows:

- ITS data category attributes to appear in global rules (even outside of an XML document or schema)
- ITS data categories attributes to pertain to sets of XML nodes (for example all `p` elements in an XML document)
- ITS markup to pertain to attributes
- ITS markup to [associate with existing markup](#) (for example the `term` element in DITA)

2.2 Overriding and Inheritance



The power of the ITS selection mechanisms comes at a price: rules related to [overriding/precedence](#), and [inheritance](#), have to be established.

The document in [Example 12](#) shows how inheritance and overriding work for the [Translate](#) data category. By default elements are translatable. Here, the `translateRule` element declared in the header overrides the default for the `head` element inside `text` and for all its children. Because the `title` element is actually translatable, the global rule needs to be overridden by a local `its:translate="yes"`. Note that the global rule is processed first, regardless of its position inside the document. In the main body of the document, the default applies, and here it is `its:translate="no"` that is used to set “faux pas” as non-translatable.

Example 12: Overriding and Inheritance

```
<text xmlns:its="http://www.w3.org/2005/11/its">
  <head>
    <revision>Sep-10-2006 v5</revision>
    <author>Ealasaidh McIan</author>
    <contact>ealasaidh@hogw.ac.uk</contact>
    <title its:translate="yes">The Origins of Modern Novel</title>
    <its:rules version="2.0">
      <its:translateRule translate="no" selector="/text/head"/>
    </its:rules>
  </head>
  <body>
    <div xml:id="intro">
      <head>Introduction</head>
      <p>It would certainly be quite a <span its:translate="no">faux pas</span> t
        dissertation on the origin of modern novel without mentioning the <tl>Epi
        Gilgamesh</tl>...</p>
    </div>
  </body>
</text>
```

[Source file: [examples/xml/EX-basic-concepts-3.xml](#)]

2.3 Adding Information or Pointing to Existing Information



For some data categories, special attributes add or point to information about the selected nodes. For example, the [Localization Note](#) data category can add information to selected nodes (using a `locNote` element), or point to existing information elsewhere in the document (using a `locNotePointer` attribute).

The functionality of adding information to the selected nodes is available for each data category except [Language Information](#). Pointing to existing information is not possible for data categories that express a *closed set of values*; that is: [Translate](#), [Directionality](#), [Locale Filter](#) and [Elements Within Text](#).

The functionalities of adding information and pointing to existing information are *mutually exclusive*. That is to say, attributes for pointing and adding must not appear at the same rule element.

3 Notation and Terminology

This section is normative.

3.1 Notation

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC 2119](#).

The namespace URI that [MUST](#) be used by implementations of this specification is:

```
http://www.w3.org/2005/11/its
```

The namespace prefix used in this specification for this URI is “its”. It is recommended that implementations of this specification use this prefix.

In addition, the following namespaces are used in this document:

- <http://www.w3.org/2001/XMLSchema> for the XML Schema namespace, here used with the prefix “xs”
- <http://www.w3.org/1999/xlink> for the XLink namespace, here used with the prefix “xlink”
- <http://www.w3.org/1999/xhtml> for the HTML namespace, here used with the prefix “h”

3.2 Data category

[Definition: ITS defines **data category** as an abstract concept for a particular type of information for internationalization and localization of XML schemas and documents.] The concept of a data category is independent of its implementation in an XML and HTML environment (e.g. using an element or attribute).

For each data category, ITS distinguishes between the following:

- the prose description, see [Section 8: Description of Data Categories](#)
- schema language independent formalization, see the “implementation” subsections in [Section 8: Description of Data Categories](#)
- schema language specific implementations, see [Appendix D: Schemas for ITS](#)

Example 13: A data category and its implementation

The [Translate](#) data category conveys information as to whether a piece of content should be translated or not.

The simplest formalization of this prose description on a schema language independent level is a `translate` attribute with two possible values: "yes" and "no". An implementation on a schema language specific level would be the declaration of the `translate` attribute in, for example, an XML Schema document or an RELAX NG document. A different implementation would be a `translateRule` element that allows for specifying [global rules](#) about the [Translate](#) data category.

3.3 Selection



[Definition: **selection** encompasses mechanisms to specify to what parts of an XML or HTML document an ITS data category and its values should be applied to.] Selection is discussed in detail in [Section 5: Processing of ITS information](#). Selection can be applied globally, see [Section 5.2.1: Global, Rule-based Selection](#), and locally, see [Section 5.2.2: Local Selection in an XML Document](#). As for global selection, ITS information can be [added](#) to the selected nodes, or it can [point to existing information](#) which is related to selected nodes.

Note:

The selection of the ITS data categories applies to textual values contained within element or attribute nodes. In some cases these nodes form pointers to other resources; a well-known example is the `src` attribute on the `img` element in HTML. The ITS [Translate](#) data category applies to the text of the pointer itself, not the object to which it points. Thus in the following example, the translation information specified via the `translateRule` element applies to the filename "instructions.jpg", and is not an instruction to open the graphic and change the words therein.

Example 14: Selecting the text of a pointer to an external object

```
<text>
  <its:rules version="2.0"
    xmlns:its="http://www.w3.org/2005/11/its">
    <its:translateRule translate="yes" selector="//p/img/@src"/>
  </its:rules>
  ...
  <p xmlns:its="http://www.w3.org/2005/11/its">As you can see in
    , the truth is not always out there.</p>
</text>
```

[Source file: [examples/xml/EX-notation-terminology-1.xml](#)]

3.4 ITS Local Attributes



[Definition: **ITS Local Attributes** are all attributes defined in [Section 8: Description of Data Categories](#) as a local markup.]

3.5 Rule Elements



[Definition: **Rule Elements** are all elements defined in [Section 8: Description of Data Categories](#) as elements for global rules.]

3.6 Usage of Internationalized Resource Identifiers in ITS



All attributes that have the type `anyURI` in the normative RELAX NG schema in [Appendix D: Schemas for ITS MUST](#) allow the usage of Internationalized Resource Identifiers (IRIs, [RFC 3987](#)) or its successor) to ease the adoption of ITS in international application scenarios.

3.7 The Term HTML



This specification uses the term `HTML` to refer to HTML5 or its successor [\[HTML5\]](#).

4 Conformance



This section is normative.

The usage of the term *conformance clause* in this section is in compliance with [\[QAFRAMEWORK\]](#).

This specification defines three types of conformance: conformance of [1\) ITS markup declarations](#), conformance of [2\) processing expectations for ITS Markup](#) and conformance of [3\) processing expectations for ITS Markup in HTML](#). Also special [conformance class](#) is defined for using ITS markup in HTML5 document which serves as an applicable specification for HTML5+ITS. These conformance types and classes complement each other. An implementation of this specification [MAY](#) use them separately or together.

4.1 Conformance Type 1: ITS Markup Declarations



Description: ITS markup declarations encompass all declarations that are part of the Internationalization Tag Set. They do not concern the *usage* of the markup in XML documents. Such markup is subject to the conformance clauses in [Section 4.2: Conformance Type 2: The Processing Expectations for ITS Markup](#).

Definitions related to this conformance type: ITS markup declarations are defined in various subsections in in a schema language independent manner.

Who uses this conformance type: Schema designers integrating ITS markup declarations into a schema. All conformance clauses for this conformance type concern the position of ITS markup declarations in that schema, and their status as mandatory or optional.

Conformance clauses:

- 1-1: At least one of the following [MUST](#) be in the schema:
 - `rules` element
 - one of the local ITS attributes
 - `span` element
 - `ruby` element
- 1-2: If the `rules` element is used, it [MUST](#) be part of the content model of at least one element declared in the schema. It [SHOULD](#) be in a content model for meta information, if this is available in that schema (e.g. the `head` element in [\[XHTML 1.0\]](#)).
- 1-3: If the `ruby` element is used, it [SHOULD](#) be declared as an inline element.
- 1-4: If the `span` element is used, it [SHOULD](#) be declared as an inline element.

Full implementations of this conformance type will implement all markup declarations for ITS. Statements related to this conformance type [MUST](#) list all markup declarations they implement.

Examples: Examples of the usage of ITS markup declarations in various existing schemas are given in a separate document [\[XML i18n BP\]](#).

4.2 Conformance Type 2: The Processing Expectations for ITS Markup

Description: Processors need to compute the ITS information that pertains to a node in an XML document. The ITS processing expectations define how the computation has to be carried out. Correct computation involves support for [selection mechanism](#), [defaults / inheritance / overriding characteristics](#), and [precedence](#). The markup [MAY](#) be valid against a schema which conforms to the clauses in [Section 4.1: Conformance Type 1: ITS Markup Declarations](#).

Definitions related to this conformance type: The processing expectations for ITS markup make use of selection mechanisms defined in [Section 5: Processing of ITS information](#). The individual data categories defined in [Section 8: Description of Data Categories](#) have [defaults / inheritance / overriding characteristics](#), and allow for using ITS markup in various positions ([global](#) and [local](#)).

Who uses this conformance type: Applications that need to process the nodes captured by a data category for internationalization or localization. Examples of this type of application are: ITS markup-aware editors, or translation tools that make use of ITS markup to filter translatable text as an input to the localization process.

Note:

Application-specific processing (that is processing that goes beyond the computation of ITS information for a node) such as automated filtering of translatable content based on the [Translate](#) data category is not covered by the conformance clauses below.

Conformance clauses:

- 2-1: A processor [MUST](#) implement at least *one* [data category](#). For each implemented [data category](#), the following [MUST](#) be taken into account:
 - 2-1-1: processing of at least one selection mechanism ([global](#) or [local](#)).
 - 2-1-2: the [default selections for the data category](#).
 - 2-1-3: the precedence definitions for selections defined in [Section 5.5: Precedence between Selections](#), for the type of selections it processes.
- 2-2: If an application claims to process ITS markup for the global selection mechanism, it [MUST](#) process an XLink `href` attribute found on a `rules` elements.
- 2-3: If an application claims to process ITS markup implementing the conformance clauses 2-1, 2-2 and 2-3, it [MUST](#) process that markup with XML documents.
- 2-4: After processing ITS information on the basis of conformance clauses [2-1](#) and [2-2](#), an application [MAY](#) convert an XML document to NIF, using the algorithm described in [Section 5.7: Conversion to NIF](#).

Note:

The conformance clause [2-4](#) essentially means that the conversion to NIF is an optional feature of ITS 2.0, and that the conversion is independent of whether ITS information has been made available via the global or local selection mechanisms, see conformance clause [2-1-1](#).

Statements related to this conformance type [MUST](#) list all [data categories](#) they implement, and for each [data category](#) which type of selection they support, whether they support processing of XML. If the implementation provides the conversion to NIF (see conformance clause [2-4](#)), this [MUST](#) be stated.

Note:

The above conformance clauses are directly reflected in the [ITS 2.0 test suite](#). All tests specify which data category is processed (clause 2-1); they are relevant for (clause 2-1-1) global or local selection, or both; they require the processing of defaults and precedence of selections (clauses 2-1-2 and 2-1-3); for each data category there are tests with linked rules (2-2); and all types of tests are given for XML (clause 2-3). In addition, there are test cases for conversion to NIF (clause 2-4). Implementors are encouraged to organize their documentation in a similar way, so that users of ITS 2.0 easily can understand the processing capabilities available.

4.3 Conformance Type 3: Processing Expectations for ITS Markup in HTML



Description: Processors need to compute the ITS information that pertains to a node in a HTML document. The ITS processing expectations define how the computation has to be carried out. Correct computation involves support for [selection mechanism](#), [defaults / inheritance / overriding characteristics](#), and [precedence](#).

Definitions related to this conformance type: The processing expectations for ITS markup make use of selection mechanisms defined in [Section 5: Processing of ITS information](#). The individual data categories defined in [Section 8: Description of Data Categories](#) have [defaults / inheritance / overriding characteristics](#), and allow for using ITS markup in various positions ([local](#), [external global](#) and [inline global](#)).

Who uses this conformance type: Applications that need to process the nodes captured by a data category for internationalization or localization. Examples of this type of application are: ITS markup-aware editors, or translation tools that make use of ITS markup to filter translatable text as an input to the localization process.

Note:

Application-specific processing (that is processing that goes beyond the computation of ITS information for a node) such as automated filtering of translatable content based on the [Translate](#) data category is not covered by the conformance clauses below.

Conformance clauses:

- 3-1: A processor **MUST** implement at least one [data category](#). For each implemented [data category](#), the following **MUST** be taken into account:
 - 3-1-1: processing of at least one selection mechanism ([global](#) or [local](#)).
 - 3-1-2: the [default selections for the data category](#).
 - 3-1-3: the precedence definitions for selections defined in [Section 6.4: Precedence between Selections](#), for the type of selections it processes.
- 3-2: If an application claims to process ITS markup for the global selection mechanism, it **MUST** process a `href` attribute found on a `link` elements which has a `rel` attribute with the value `its-rules`.
- 3-3: If an application claims to process ITS markup implementing the conformance clauses 3-1, 3-2 and 3-3, it **MUST** process that markup within HTML documents.

Statements related to this conformance type **MUST** list all [data categories](#) they implement, and for each [data category](#) which type of selection they support.

4.4 Conformance Class for HTML5+ITS documents



Conforming HTML5+ITS documents are those that comply with all the conformance criteria for documents as defined in [\[HTML5\]](#) with the following exception:

- [Global attributes](#) which can be used on all HTML elements are extended by attributes for local data categories as defined in [Section 6.1: Mapping of Local Data Categories to HTML](#).

5 Processing of ITS information

This section is normative.

Note:

Additional definitions about processing of HTML are given in [Section 6: Using ITS Markup in HTML](#).

5.1 Indicating the Version of ITS

The version of the ITS schema defined in this specification is "2.0". The version is indicated by the ITS `version` attribute. This attribute is mandatory for the `rules` element, where it **MUST** be in no namespace. If there is no `rules` element in an XML document, a prefixed ITS `version` attribute (e.g. `its:version`) **MUST** be provided at the root element of the document. If there is both a `version` attribute at the root element and a `rules` element in a document, they **MUST NOT** specify different versions.

External, linked rules can have different versions than internal rules.

5.2 Locations of Data Categories

ITS data categories can appear in two places:

- [Global rules](#): the selection is realized within a `rules` element. It contains [rule elements](#) for each data category. Each rule element has a `selector` attribute and possibly other attributes. The `selector` attribute contains an absolute selector as defined in [Section 5.3: Query Language of Selectors](#).
- [Locally in a document](#): the selection is realized using ITS local attributes, which are attached to an element node, or the `span` or `ruby` element. There is no additional `selector` attribute. The default selection for each data category defines whether the selection covers attributes and child elements. See [Section 8.1: Position, Defaults, Inheritance and Overriding of Data Categories](#).

The two locations are described in detail below.

5.2.1 Global, Rule-based Selection

Global, rule-based selection is implemented using the `rules` element. It contains zero or more [rule elements](#). Each [rule element](#) has a mandatory `selector` attribute. This attribute and all other possible attributes on [rule elements](#) are in the empty namespace and used without a prefix.

If there is more than one `rules` element in an XML document, the rules from each section are to be processed at the same precedence level. The `rules` sections are to be read in document order, and the ITS rules with them processed sequentially. The versions of these `rules` elements **MUST NOT** be different.

Depending on the data category and its usage, there are additional attributes for adding information to the selected nodes, or for pointing to existing information in the document. For example, the [Localization Note](#) data category can be used for adding notes to selected nodes, or for pointing to

existing notes in the document. For the former purpose, a `locNote` element can be used. For the latter purpose, a `locNotePointer` attribute can be used.

Each data category allows users to add information to the selected nodes except for [language information](#). Pointing to existing information is not possible for data categories that express a *closed set of values*, that is: [Translate](#), [Directionality](#), [Locale Filter](#), and [Elements Within Text](#).

The functionalities of adding information and pointing to existing information are *mutually exclusive*. That is: markup for pointing and adding **MUST NOT** appear in the same rule element.

Global rules can appear in the XML document they will be applied to, or in a separate XML document. The precedence of their processing depends on these variations. See also [Section 5.5: Precedence between Selections](#).

5.2.2 Local Selection in an XML Document



Local selection in XML documents is realized with [ITS local attributes](#) or the `span` element. `span` serves just as a carrier for the local ITS attributes.

The data category determines what is being selected. The necessary data category specific defaults are described in [Section 8.1: Position, Defaults, Inheritance and Overriding of Data Categories](#).

Example 15: Defaults for various data categories

By default the content of all elements in a document is translatable. The attribute `its:translate="no"` in the `head` element means that the content of this element, including child elements, should not be translated. The attribute `its:translate="yes"` in the `title` element means that the content of this element, should be translated (overriding the `its:translate="no"` in head). Attribute values of the selected elements or their children are not affected by local `translate` attributes. By default they are not translatable.

The default directionality of a document is left-to-right. The `its:dir="rtl"` in the `quote` element means that the directionality of the content of this element, including child elements and attributes, is right-to-left. Note that `xml:lang` indicates only the language, not the directionality.

```
<text xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0" xml:lang="en">
  <head its:translate="no">
    <author>Sven Corneliusson</author>
    <date>2006-09-26T17:34:04Z</date>
    <title its:translate="yes" role="header">Bidirectional Text</title>
  </head>
  <body>
    <par>In Arabic, the title <quote xml:lang="ar" its:dir="rtl">التدويل، نشاط W3</quote>
      <quote>Internationalization Activity, W3C</quote>.</par>
  </body>
</text>
```

[Source file: [examples/xml/EX-selection-local-1.xml](#)]

Note:

The `dir` and `translate` attributes are not listed in the ITS attributes to be used in HTML. The reason is that these two attributes are available in HTML natively, so there is no need to provide them as `its-` attributes. The definition of the two attributes in HTML is compatibly, that is it

provides the same values and interpretation, as the definition for the two data categories [Translate](#) and [Directionality](#).

5.3 Query Language of Selectors

5.3.1 Choosing Query Language

[Rule elements](#) have attributes which contain absolute and relative selectors. Interpretation of these selectors depends on the actual query language. The query language is set by `queryLanguage` attribute on `rules` element. If `queryLanguage` is not specified XPath 1.0 is used as a default query language.

5.3.2 XPath 1.0

XPath 1.0 is identified by `xpath` value in `queryLanguage` attribute.

5.3.2.1 Absolute selector

The absolute selector [MUST](#) be an XPath expression which starts with `/`. That is, it must be an [AbsoluteLocationPath](#) or union of [AbsoluteLocationPath](#)s as described in [XPath 1.0](#). This ensures that the selection is not relative to a specific location. The resulting nodes [MUST](#) be either element or attribute nodes.

Context for evaluation of the XPath expression is as follows:

- Context node is set to [Root Node](#).
- Both context position and context size are 1.
- All variables defined by `param` elements are bind.
- All functions defined in the [XPath Core Function Library](#) are available. It is an error for an expression to include a call to any other function.
- The set of namespace declarations are those in scope on the element which has the attribute in which the expression occurs. This includes the implicit declaration of the prefix `xml` required by the [XML Namespaces Recommendation](#); the default namespace (as declared by `xmlns`) is not part of this set.

Example 16: XPath expressions with namespaces

The `term` element from the TEI is in a namespace `http://www.tei-c.org/ns/1.0`.

```
<!-- Definitions for TEI -->
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:termRule selector="//tei:term" term="yes" xmlns:tei="http://www.tei-c.org/ns/1.0"/>
</its:rules>
```

[Source file: [examples/xml/EX-selection-global-1.xml](#)]

Example 17: XPath expressions without namespaces

The `term` element from DocBook V4.5 is in no namespace.

```
<!-- Definitions for DocBook -->
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:termRule selector="//term" term="yes"/>
</its:rules>
```

```
</its:rules>
```

[Source file: <examples/xml/EX-selection-global-2.xml>]

5.3.2.2 Relative selector

The relative selector **MUST** use a [RelativeLocationPath](#) or an [AbsoluteLocationPath](#) as described in [XPath 1.0](#). The XPath expression is evaluated relative to the nodes selected by the selector attribute.

The following attributes point to existing information: `allowedCharactersPointer`, `disambigClassPointer`, `disambigClassRefPointer`, `disambigIdentPointer`, `disambigIdentRefPointer`, `disambigSourcePointer`, `domainPointer`, `externalResourceRefPointer`, `langPointer`, `locNotePointer`, `locNoteRefPointer`, `locQualityIssuesRefPointer`, `provenanceRecordsRefPointer`, `storageEncodingPointer`, `storageSizePointer`, `targetPointer`, `termInfoPointer`, `termInfoRefPointer`.

Context for evaluation of the XPath expression is same as for absolute selector with the following changes:

- Nodes selected by the expression in the `selector` attribute form the current node list.
- Context node comes from the current node list.
- The context position comes from the position of the current node in the current node list; the first position is 1.
- The context size comes from the size of the current node list.

5.3.3 CSS Selectors



Note:

As of writing the working group has no implementation commitment for CSS selectors. If this doesn't change CSS selectors will be marked as feature at risk for the candidate recommendation draft.

CSS Selectors are identified by `css` value in `queryLanguage` attribute.

5.3.3.1 Absolute selector

Absolute selector **MUST** be interpreted as selector as defined in [Selectors Level 3](#). Both simple selectors and groups of selectors can be used.

5.3.3.2 Relative selector

Relative selector **MUST** be interpreted as selector as defined in [Selectors Level 3](#). Selector is not evaluated against the complete document tree but only against subtrees rooted at nodes selected by selector in the `selector` attribute.

5.3.4 Additional query languages



ITS processors [MAY](#) support additional query languages. For each additional query language processor [MUST](#) define:

- identifier of query language used in `queryLanguage`;
- rules for evaluating absolute selector to collection of nodes;
- rules for evaluating relative selector to collection of nodes.

Future versions of this specification [MAY](#) define additional query languages. The following query language identifiers are reserved: `xpath`, `css`, `xpath2`, `xpath3`, `xquery`, `xquery3`, `xslt2`, `xslt3`.

5.3.5 Variables in selectors



A `param` element (or several ones) can be placed as the first child element(s) of the `rules` element to define the default values of variables used in the various selectors used in the rules.

Implementation [MUST](#) support the `param` element for all query languages it supports and which at the same time define how variables are bind for evaluation of selector expression. Implementations [SHOULD](#) also provide means for changing the default values of the `param` elements. Such means are implementation-specific.

The `param` element has a required name attribute. The value of the name attribute is a [QName](#), see [\[XML Names\]](#). The content of the element is a string used as default value for the corresponding variable.

Example 18: Using the `param` element to define the default value of a variable in a `selector` attribute.

The `param` element defines the default value for the `$LCID` variable. In this case, only the `msg` element with the attribute `lcid` set to "0x0409" is seen as translatable.

```
<doc its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:rules version="2.0">
    <its:param name="LCID">0x0409</its:param>
    <its:translateRule selector="/doc" translate="no"/>
    <its:translateRule selector="//msg[@lcid=$LCID]" translate="yes"/>
  </its:rules>
  <msg lcid="0x0409" num="1">Create a folder</msg>
  <msg lcid="0x0411" num="1">フォルダーを作成する</msg>
  <msg lcid="0x0407" num="1">Erstellen Sie einen Ordner</msg>
  <msg lcid="0x040c" num="1">Cr  er un dossier</msg>
</doc>
```

[Source file: [examples/xml/EX-param-in-global-rules-1.xml](#)]

Note:

In XSLT-based applications, it may make sense to map ITS parameters directly to XSLT parameters. To avoid naming conflicts one can use a prefix with the parameter name's value to distinguish between the ITS parameters and the XSLT parameters.

5.4 Link to External Rules



One way to associate a document with a set of external ITS rules is to use the optional XLink [XLink 1.1](#) `href` attribute in the `rules` element. The referenced document must be a valid XML document containing at most one `rules` element. That `rules` element can be the root element or anywhere within the document tree (for example, the document could be an XML Schema).

The rules contained in the referenced document **MUST** be processed as if they were at the top of the `rules` element with the XLink `href` attribute.

Example 19: External file EX-link-external-rules-1.xml with global rules:

The example demonstrates how metadata can be added to ITS rules.

```
<myFormatInfo>
  <desc>ITS rules used by the Open University</desc>
  <hostVoc>http://www.tei-c.org/ns/1.0</hostVoc>
  <rulesId>98ECED99DF63D511B1250008C784EFB1</rulesId>
  <rulesVersion>v 1.81 2006/03/28 07:43:21</rulesVersion>
  ...
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <its:translateRule selector="//header" translate="no"/>
    <its:translateRule selector="//term" translate="no"/>
    <its:termRule selector="//term" term="yes"/>
    <its:withinTextRule withinText="yes" selector="//term | //b"/>
  </its:rules>
</myFormatInfo>
```

[Source file: [examples/xml/EX-link-external-rules-1.xml](#)]

Example 20: Document with a link to EX-link-external-rules-1.xml

```
<myDoc>
  <header>
    <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
      xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="EX-link-external-rul
    <its:translateRule selector="//term" translate="yes"/>
  </its:rules>
  <author>Theo Brumble</author>
  <lastUpdate>Apr-01-2006</lastUpdate>
  </header>
  <body>
    <p>A <term>Palouse horse</term> has a spotted coat.</p>
  </body>
</myDoc>
```

[Source file: [examples/xml/EX-link-external-rules-2.xml](#)]

The result of processing the two documents above is the same as processing the following document.

Example 21: Document with identical rules as in the case of included rules

```

<myDoc>
  <header>
    <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
      <its:translateRule selector="//header" translate="no"/>
      <its:translateRule selector="//term" translate="no"/>
      <its:termRule selector="//term" term="yes"/>
      <its:withinTextRule withinText="yes" selector="//term | //b"/>
      <its:translateRule selector="//term" translate="yes"/>
    </its:rules>
    <author>Theo Brumble</author>
    <lastUpdate>Apr-01-2006</lastUpdate>
  </header>
  <body>
    <p>A <term>Palouse horse</term> has a spotted coat.</p>
  </body>
</myDoc>

```

[Source file: <examples/xml/EX-link-external-rules-3.xml>]

Example 22: External rules file with the `rules` element as the root element

Like [Example 19](#), these rules can be applied e.g. to [Example 20](#). The only difference is that in [Example 22](#), the `rules` element is the root element of the external file.

```

<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:translateRule selector="//header" translate="no"/>
  <its:translateRule selector="//term" translate="no"/>
  <its:termRule selector="//term" term="yes"/>
  <its:withinTextRule withinText="yes" selector="//term | //b"/>
</its:rules>

```

[Source file: <examples/xml/EX-link-external-rules-4.xml>]

Applications processing global ITS markup **MUST** recognize the XLink `href` attribute in the `rules` element; they **MUST** load the corresponding referenced document and process its rules element before processing the content of the `rules` element where the original XLink `href` attribute is.

External rules may also have links to other external rules. The linking mechanism is recursive, the deepest rules being overridden by the top-most rules, if any.

5.5 Precedence between Selections



The following precedence order is defined for selections of ITS information in various positions (the first item in the list has the highest precedence):

1. Selection via explicit (that is, not inherited) local ITS markup in documents ([ITS local attributes](#) on a specific element)
2. Global selections in documents (using a `rules` element)

Inside each `rules` element the precedence order is:

1. Any rule inside the rules element
2. Any rule linked via the XLink `href` attribute

Note:

ITS does not define precedence related to rules defined or linked based on non-ITS mechanisms (such as processing instructions for linking rules).

3. Selections via defaults for data categories, see [Section 8.1: Position, Defaults, Inheritance and Overriding of Data Categories](#)

In case of conflicts between global selections via multiple [rules](#) elements, the last rule has higher precedence.

Note:

The precedence order fulfills the same purpose as the built-in template rules of [\[XSLT 1.0\]](#). Override semantics are always complete, that is all information provided via lower precedence is overridden by the higher precedence. E.g. defaults are overridden by inherited values, these are overridden by nodes selected via global rules, which are in turn overridden by local markup.

Example 23: Conflicts between selections of ITS information which are resolved using the precedence order

The two elements `title` and `author` of this document should be treated as separate content when inside a `prolog` element, but as part of the content of their parent element otherwise. In order to make this distinction two `withinTextRule` elements are used:

The first rule specifies that `title` and `author` in general should be treated as an element within text. This overrides the default.

The second rule indicates that when `title` or `author` are found in a `prolog` element their content should be treated separately. This is normally the default, but the rule is needed to override the first rule.

```
<text>
  <prolog>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:withinTextRule withinText="yes" selector="//title|//author"/>
      <its:withinTextRule withinText="no" selector="//prolog/title|//prolog/author"/>
    </its:rules>
    <title>Designing User Interfaces</title>
    <author>Janice Prakash</author>
    <keywords>user interface, ui, software interface</keywords>
  </prolog>
  <body>
    <p>The book <title>Of Mice and Screens</title> by <author>Aldus Brandywine</author>
      the best introductions to the vast topic of designing user interfaces.</p>
  </body>
</text>
```

[Source file: [examples/xml/EX-selection-precedence-1.xml](#)]

5.6 Associating ITS Data Categories with Existing Markup



Some markup schemes provide markup which can be used to express ITS data categories. ITS data categories can be associated with such existing markup, using the global selection mechanism described in [Section 5.2.1: Global, Rule-based Selection](#).

Associating existing markup with ITS data categories can be done only if the processing expectations of the host markup are the same as, or greater than, those of ITS. For example, the [\[DITA 1.0\]](#) format can use its `translate` attribute to apply to “transcluded” content, going beyond the ITS 2.0 local selection mechanism, but not contradicting it.

Example 24: Association of the ITS data categories [Translate](#) and [Terminology](#) with DITA 1.0 markup

In this example, there is an existing `translate` attribute in DITA, and it is associated with the ITS semantics using the `its:rules` section. Similarly, the DITA `dt` and `term` elements are associated with the ITS [Terminology](#) data category.

```
<topic id="myTopic">
  <title>The ITS Topic</title>
  <prolog>
    <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
      <its:translateRule selector="/*[@translate='no']" translate="no"/>
      <its:translateRule selector="/*[@translate='yes']" translate="yes"/>
      <its:termRule selector="//term | //dt" term="yes"/>
    </its:rules>
  </prolog>
  <body>
    <dl>
      <dlentry id="tDataCat">
        <dt>Data category</dt>
        <dd>ITS defines <term>data category</term> as an abstract concept for a p
          information related to internationalization and localization of XML sch
          documents.</dd>
      </dlentry>
    </dl>
    <p>For the implementation of ITS, apply the rules in the order:</p>
    <ul>
      <li>Defaults</li>
      <li>Rules in external files</li>
      <li>Rules in the document</li>
      <li>Local attributes</li>
    </ul>
    <p><ph translate="no" xml:lang="fr">Et voilà !</ph>.</p>
  </body>
</topic>
```

[Source file: [examples/xml/EX-associating-its-with-existing-markup-1.xml](#)]

Global rules can be associated with a given XML document using different means:

- By using an `rules` element in the document itself:
 - with the rules directly inside the document, as shown in [Example 24](#)
 - with a link to an external rules file using the XLink `href` attribute, as shown in [Example 19](#)
- By associating the rules and the document through a tool-specific mechanism. For example, for a command-line tool: providing the paths of both the XML document to process and its corresponding external rules file.

5.7 Conversion to NIF



This section defines an algorithm to convert XML or HTML documents (or their DOM representations) that contain ITS metadata to the RDF-based format [NIF](#). The conversion results in RDF triples.

Note:

The algorithm is intended to extract the text from the XML/HTML/DOM for an NLP tool and can produce a lot of "*phantom*" predicates from excessive whitespace, which 1) increases the size of

the intermediate mapping and 2) extracts this whitespace as text. This might decrease NLP performance. It is recommended to normalize whitespace in the input XML/HTML/DOM in order to minimize such phantom predicates. A normalized example is given below. The whitespace normalization algorithm itself is format dependent, e.g. it differs for HTML compared to general XML. Hence no normative algorithm for whitespace normalization is given as part of this specification.

Example 25: Example of an HTML document with whitespace normalized as preparation for conversion to NIF

```
<html><body><h2 translate="yes">Welcome to <span
  its-disambig-ident-ref="http://dbpedia.org/resource/Dublin"
  translate="no">Dublin</span> in <b translate="no">Ireland</b>!</h2></body></ht
```

The conversion algorithm to generate NIF consists of seven steps.

- STEP 1: Get an ordered list of all text nodes of the document.
- STEP 2: Generate an XPath expression for each non-empty text node of all leaf elements and remember them.
- STEP 3: Get the text for each node and make a tuple with the XPath expressions (X,T). Since the text nodes have a certain order we now have a list of ordered tuples ((x0,t0), (x1,t1), ..., (xn,tn)).
- STEP 4 (optional): Serialize as XML or as RDF. The list with the XPath-to-text mapping can also be kept in memory. Part of a serialization example is given below.

```
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/exampledoc.html#xpath(x0)>
  itsrdf:xpath2nif <http://example.com/exampledoc.html#offset_b0_e0>
<http://example.com/exampledoc.html#xpath(x1)>
  itsrdf:xpath2nif <http://example.com/exampledoc.html#offset_b1_e1>
# ...
<http://example.com/exampledoc.html#xpath(xn)>
  itsrdf:xpath2nif <http://example.com/exampledoc.html#offset_bn_en>
<mappings>
  <mapping x="xpath(x0)" b="b0" e="e0" />
  <mapping x="xpath(x1)" b="b1" e="e1" />
  <!-- ... -->
  <mapping x="xpath(xn)" b="bn" e="en" />
</mappings>
```

where

```
b0 = 0
e0 = b0 + (Number of characters of t0)
b1 = e0 + 1
e1 = b1 + (Number of characters of t1)
...
bn = e(n-1) + 1
en = bn + (Number of characters of tn)
```

Example (continued)

```

@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
# "Welcome to "
<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/text()[1])>
  itsrdf:nif <http://example.com/exampledoc.html#offset_0_11> .
# "Dublin"
<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/span[1]/text()[1])>
  itsrdf:nif <http://example.com/exampledoc.html#offset_11_17> .
# " in "
<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/text()[2])>
  itsrdf:nif <http://example.com/exampledoc.html#offset_17_21> .
# "Ireland"
<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/b[1]/text()[1])>
  itsrdf:nif <http://example.com/exampledoc.html#offset_21_28> .
# "!"
<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/text()[3])>
  itsrdf:nif <http://example.com/exampledoc.html#offset_28_29> .
# "Welcome to Dublin Ireland!"
<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/text())>
  itsrdf:nif <http://example.com/exampledoc.html#offset_0_29> .
<mappings>
  <mapping x="xpath(/html/body[1]/h2[1]/text()[1])" b="0" e="11" />
  <mapping x="xpath(/html/body[1]/h2[1]/span[1]/text()[1])" b="11" e="17" />
  <mapping x="xpath(/html/body[1]/h2[1]/text()[2])" b="17" e="21" />
  <mapping x="xpath(/html/body[1]/h2[1]/b[1]/text()[1])" b="21" e="28" />
  <mapping x="xpath(/html/body[1]/h2[1]/text()[3])" b="28" e="29" />
  <mapping x="xpath(/html/body[1]/h2[1])" b="0" e="29" />
</mappings>

```

- STEP 5: Create a context URI and attach the whole concatenated text of the document as reference.
- STEP 6: Now attach any ITS metadata items from the XML/HTML/DOM input to respective NIF URIs.
- STEP 7: Omit all irrelevant URIs (those that do not carry annotations, they will just bloat the data).

```

@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
<http://example.com/exampledoc.html#offset_0_29>
  rdf:type          str:Context ;
  rdf:type          str:OffsetBasedString ;
# concatenate the whole text
  str:isString      "$(t0+t1+t2+...+tn)" ;
  itsrdf:translate  "yes"^^<http://www.w3.org/TR/its-2.0/its.xsd#yesOrNo> ;
  str:occursIn      <http://example.com/exampledoc.html> .
<http://example.com/exampledoc.html#offset_11_17>
  rdf:type          str:String ;
  rdf:type          str:OffsetBasedString ;
  itsrdf:translate  "no"^^<http://www.w3.org/TR/its-2.0/its.xsd#yesOrNo> ;
  itsrdf:disambigIdentRef <http://dbpedia.org/resource/Dublin> ;
  str:referenceContext <http://example.com/exampledoc.html#offset_0_29> .
<http://example.com/exampledoc.html#offset_21_28>
  rdf:type          str:String ;
  rdf:type          str:OffsetBasedString ;
  itsrdf:translate  "no"^^<http://www.w3.org/TR/its-2.0/its.xsd#yesOrNo> ;
  str:referenceContext <http://example.com/exampledoc.html#offset_0_29> .

```

A complete sample output in RDF/XML format after step 7, given the input document [Example 25](#), is available at <examples/nif/EX-nif-conversion-output.xml>.

Note:

The conversion to NIF is the basis for natural language processing (NLP) applications, creating for example named entity annotations. A non-normative algorithm to integrate these annotations into the original input document is given in [Appendix G: Conversion NIF2ITS](#). The algorithm in that appendix is non-normative since many choices depend on the actual NLP application.

5.8 ITS Tools Annotation



In some cases, it may be important for instances of data categories to be associated with information about the processor that generated them. For example, the score of the [MT Confidence](#) data category (provided via the `mtConfidence` attribute) is meaningful only when the consumer of the information also knows what MT engine produced it, because the score provides the relative confidence of translations from the same MT engine but does not provide a score that can be reliably compared between MT engines. The same is true for confidence provided for the [Disambiguation](#) data category, providing confidence information via the `disambigConfidence` attribute, or the [Terminology](#) data category, providing confidence information via the `termConfidence` attribute.

ITS 2.0 provides a mechanism to associate such processor information with the use of individual data categories in a document, independently from data category annotations themselves.

The attribute `annotatorsRef` provides a way to associate all the annotations of a given data category within the element with information about the processor that generated those data category annotations.

Note:

- Three cases of providing tool information can be expected:
 1. information about tools used for creating or modifying the textual content;
 2. information about tools that do 1), but also create ITS annotations, see [Appendix H: List of ITS 2.0 Global Elements and Local Attributes](#);
 3. information about tools that don't modify or create content, but just create ITS annotations.

`annotatorsRef` is only meant to be used when actual ITS annotation is involved, that is for 2) and 3). To express tool information related only to the creation or modification of textual content and independent of ITS data categories, that is case 1), one should use the tool or `toolRef` attribute provided by the [Provenance](#) data category.
- An example of case 2) is an MT engine that modifies content and creates ITS [MT Confidence](#) annotations. Here the situation may occur that several tools are involved in creating MT Confidence annotations: the MT engine and the tool inserting the markup. The `annotatorsRef` attribute should identify the tool most useful in further processes, in this case the MT engine.

The value of `annotatorsRef` is a space-separated list of references where each reference is composed of two parts: a data category identifier and an IRI. These two parts are separated by a character | VERTICAL LINE (U+007C).

- The data category identifier **MUST** be one of the identifiers specified in the [data category overview table](#).
- The IRI indicates information about the processor used to generate the data category annotation. No single means is specified for how this IRI should be used to indicate processor information. Possible mechanisms are: to encode information directly in the IRI, e.g. as parameters; to reference an external resource that provides such information, e.g.

an XML file or an RDF declaration; or to reference another part of the document that provides such information.

In HTML documents, the mechanism is implemented with the `its-annotators-ref` attribute.

The attribute applies to the content of the element where it is declared (including its children elements) and to the attributes of that element.

On any given node, the information provided by this mechanism is a space-separated list of the accumulated references found in the `annotatorsRef` attributes declared in the enclosing elements and sorted by data category identifiers. For each data category, the IRI part is the one of the inner-most declaration.

Example 26: Accumulation and Overriding of the `annotatorsRef` Values

In this example, the text shows the computed tools reference information for the given node. Note that the references are ordered alphabetically and that the IRI values are always the ones of the inner-most declaration.

```
<doc its:version="1.5" xmlns:its="http://www.w3.org/2005/11/its"
    its:annotatorsRef="mt-confidence|MT1"
  >doc node: "mt-confidence|MT1"
  <group its:annotatorsRef="lq-issue|ABC"
    >group node: "lq-issues|ABC mt-confidence|MT1"
    <p its:annotatorsRef="disambiguation|Tool3"
      >This p node: "disambiguation|Tool3 lq-issue|ABC mt-confidence|MT1"</p>
    <p its:annotatorsRef="mt-confidence|MT123"
      >This p node: "disambiguation|Tool3 lq-issue|ABC mt-confidence|MT123"</p>
    </group>
    <p its:annotatorsRef="disambiguation|XYZ"
      >This p node: "disambiguation|XYZ mt-confidence|MT1"</p>
  </doc>
```

[Source file: <examples/xml/EX-its-tool-annotation-1.xml>]

Example 27: Example of ITS Tools Annotation

The `annotatorsRef` attribute is used in this XML document to indicate that information about the processor that generated the `mtConfidence` values for the first two `p` elements are found in element with `id="T1"` in the external document `tools.xml`, while that information for the third `p` element is found in the element with `id="T2"` in the same document. In addition, `annotatorsRef` is used to identify a Web resource with information about the QA tool used to generate the [Localization Quality Issue](#) annotation in the document.

```
<doc its:version="2.0"
    its:annotatorsRef="mt-confidence|file:///tools.xml#T1 lq-issue|http://www.qal
    xmlns:its="http://www.w3.org/2005/11/its">
  <p its:mtConfidence="0.78">Text translated with tool T1</p>
  <p its:mtConfidence="0.55" its:locQualityIssueType="typographical"
    its:locQualityIssueComment="Sentence without capitalization"
    its:locQualityIssueSeverity="50">text also translated with tool T1</p>
  <p its:mtConfidence="0.34" its:annotatorsRef="mt-confidence|file:///tools.xml
    with tool T2</p>
</doc>
```

[Source file: <examples/xml/EX-its-tool-annotation-2.xml>]

Example 28: Example of ITS Tool Annotation

The `its-annotators-ref` attributes are used in this HTML document to indicate that the [MT Confidence](#) annotation on the first two `span` elements come from one MT (French to English) engine, while the annotation on the third comes from another (Italian to English) engine. Both `its-annotators-ref` attributes refer to a Web resource for information about the engine generating the [MT Confidence](#) annotation.

```
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Sentences about capital cities
      machine translated into english with mtConfidence defined
      locally.</title>
  </head>
  <body its-annotators-ref="mt-confidence|http://www.exmt-prov.com/2012/11/9/fr
    <p>
      <span its-mt-confidence=0.8982>Dublin is the capital of Ireland.</span>
      <span its-mt-confidence=0.8536>The capital of the Czech Republic is P
      <span its-mt-confidence=0.7009
        its-annotators-ref="mt-confidence|http://www.exmt-prov.com/2012/1
      </p>
    </body>
  </html>
```

[Source file: [examples/html5/EX-its-tool-annotation-html5-1.html](#)]

6 Using ITS Markup in HTML



6.1 Mapping of Local Data Categories to HTML



All data categories defined in [Section 8: Description of Data Categories](#) and having local implementation might be used in HTML with the exception of [Translate](#), [Directionality](#), [Ruby](#), and [Language Information](#) data categories.

Note:

The above mentioned data categories are excluded because HTML has native markup for them.

In HTML data categories are implemented as attributes. Name of HTML attribute is derived from the name of attribute defined in the local implementation by using the following rules:

1. Attribute name is prefixed with `its-`
2. Each uppercase letter in the attribute name is replaced by – (U+002D) followed by a lowercase variant of the letter.

Values of attributes which corresponds to data categories with a predefined set of values **MUST** be matched case-insensitively.

Note:

Case of attribute names is also irrelevant given the nature of HTML syntax. So in HTML terminology data category can be stored as `its-term`, `ITS-TERM`, `its-Term` etc. All those attributes are treated as equivalent and will get normalized upon DOM construction.

6.2 Global rules

Various aspects for global rules in general, external global rules or inline global rules need to be taken into account.

Note:

By default XPath 1.0 will be used for selection in global rules. If users prefer easier selection mechanism, they can switch query language to CSS selectors by using the `queryLanguage` attribute, see [Section 5.3.1: Choosing Query Language](#).

Note:

HTML5 parsing algorithm automatically puts all HTML elements into XHTML namespace (<http://www.w3.org/1999/xhtml>). Selectors used in global rules must take this into account.

Link to external global rules is specified in `href` attribute of `link` element, with the link relation `its-rules`.

Note:

Using XPath in global rules linked from HTML documents does not create an additional burden to implementers. Parsing HTML content produces a DOM tree that can be directly queried using XPath, functionality supported by all major browsers.

Inline global rules **MUST** be specified inside `script` which has `type` attribute with the value `application/its+xml`. The `script` element itself **SHOULD** be child of `head` element. Comments **MUST NOT** be used inside global rules. Each `script` element **MUST NOT** contain more than one `rules` element.

Note:

It is preferred to use external global rules linked using `link` element.

6.3 Standoff Markup in HTML

The constraints for [Provenance standoff markup in HTML](#) and [Localization quality issues markup in HTML](#) **MUST** be followed.

6.4 Precedence between Selections

The following precedence order is defined for selections of ITS information in various positions of HTML document (the first item in the list has the highest precedence):

1. Implicit local selection in documents ([ITS local attributes](#) on a specific element)
2. Global selections in documents (using mechanism of [external global rules](#) or [inline global rules](#))

Note:

ITS does not define precedence related to rules defined or linked based on non-ITS mechanisms (such as processing instructions for linking rules).

3. Selections via defaults for data categories, see [Section 8.1: Position, Defaults, Inheritance and Overriding of Data Categories](#)

In case of conflicts between global selections via multiple [rules](#) elements, the last rule has higher precedence.

7 Using ITS Markup in XHTML



XHTML documents aimed at public consumption by Web browsers [SHOULD](#) use syntax for local attributes described in [Section 6.1: Mapping of Local Data Categories to HTML](#) and [SHOULD NOT](#) use [inline global rules](#) in order to adhere to [DOM Consistency HTML Design Principle](#).

Example 29: Using ITS 2.0 markup in XHTML

This examples illustrates the use of ITS 2.0 local markup and global rules in XHTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>XHTML and ITS2.0</title>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
      xmlns:h="http://www.w3.org/1999/xhtml">
      <its:domainRule selector="/h:html/h:body"
        domainPointer="/h:html/h:head/h:meta[@name='keywords']/@content">
      <its:locNoteRule locNoteType="description" selector="/h:html/h:body">
        <its:locNote>ITS Rules can directly used inside of XHTML.</its:locNote>
      </its:locNoteRule>
    </its:rules>
    <meta name="keywords" content="ITS, domain, 'localization note', example">
  </head>
  <body>
    <h1>XHTML and ITS2.0</h1>
    <p>Don't use <span its-loc-note="Internationalization Tag Set">ITS</span>
      attributes inside the content, like its:locNote.</p>
  </body>
</html>
```

[Source file: [examples/html5/EX-xhtml5-markup-1.html](#)]

8 Description of Data Categories



This section is normative.

8.1 Position, Defaults, Inheritance and Overriding of Data Categories



The following table summarizes for each data category which selection, default value, and inheritance and overriding behavior applies. It also provides data category identifiers used in [Section 5.8: ITS Tools Annotation](#).

- *Default values* apply if both local and global selection are absent. The default value for the [Translate](#) data category for example mandates that elements are translatable, and attributes

are not translatable if there is no `translateRule` element and no `translate` attribute available.

- *Inheritance* describes whether ITS information is applicable to child elements of nodes and attributes related to these nodes or their child nodes. The inheritance for the [Translate](#) data category for example mandates that all child elements of nodes are translatable whereas all attributes related to these the nodes or their child nodes are not translatable.
- For ITS data categories with inheritance, the information conveyed by the data category can be overridden. For example, a local `translate` attribute overrides the [Translate](#) information conveyed by a global `translateRule`.

Note:

An ITS application is free to decide what pieces of content it uses. For example:

- [Terminology](#) information is added to a `term` element. The information pertains only to the content of the element, since there is no inheritance for [Terminology](#). Nevertheless an ITS application can make use of the complete element, e.g. including attribute nodes etc.
- Using [Id value](#), a unique identifier is provided for a `p` element. An application can make use of the complete `p` element, including child nodes and attributes nodes. The application is also free to make use just of the string value of `p`. Nevertheless the id provided via [ID value](#) pertains only to the `p` element. It cannot be used to identify nested elements or attributes.
- Using [target pointer](#), selected `source` element have the ITS information that their translation is available in a `target` element; see [Example 70](#). This information does not inherit to child elements of `target pointer`. E.g., the translation of a `span` element nested in `source` is not available in a specific `target` element. Nevertheless, an application is free to use the complete content of `source`, including `span`, and e.g. present it to a translator.

Data category (identifier)	Local Usage	Global, rule-based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance for elements nodes	Examples
Translate (translate)	Yes	Yes	Yes	No	<code>translate="yes"</code> for elements, and <code>translate="no"</code> for attributes	Textual content of element, <i>including</i> content of child elements, but <i>excluding</i> attributes	local , global
Localization Note (localization-note)	Yes	Yes	Yes	Yes	None	Textual content of element, <i>including</i> content of child elements, but <i>excluding</i> attributes	local , global
Terminology (terminology)	Yes	Yes	Yes	Yes	<code>term="no"</code>	None	local , global

Data category (identifier)	Local Usage	Global, rule- based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance for elements nodes	Examples
Directionality (directionality)	Yes	Yes	Yes	No	dir="ltr"	Textual content of element, <i>including</i> attributes and child elements	local , global
Ruby (ruby)	Yes	Yes	Yes	Yes	None	None	local , global
Language Information (language- information)	No	Yes	No	Yes	None	Textual content of element, <i>including</i> attributes and child elements	global
Elements Within Text (elements- within-text)	Yes	Yes	Yes	No	withinText="no"	None	local , global
Domain (domain)	No	Yes	Yes	Yes	None	Textual content of element, <i>including</i> attributes and child elements	global
Disambiguation (disambiguation)	Yes	Yes	Yes	Yes	None	None	local , global
Locale Filter (locale-filter)	Yes	Yes	Yes	No	localeFilterList="*"	Textual content of element, <i>including</i> attributes and child elements	local , global
Provenance (provenance)	Yes	Yes	No	Yes	None	Textual content of element, <i>including</i> child elements and attributes	local , global
External Resource (external- resource)	No	Yes	No	Yes	None	None	global
Target Pointer (target-pointer)	No	Yes	No	Yes	None	None	global
Id Value (id- value)	No	Yes	No	Yes	None	None	global

Data category (identifier)	Local Usage	Global, rule- based selection	Global adding of information	Global pointing to existing information	Default Values	Inheritance for elements nodes	Examples
Preserve Space (preserve-space)	Yes	Yes	Yes	No	default	Textual content of element, <i>including</i> attributes and child elements	local , global
Localization Quality Issue (localization- quality-issue)	Yes	Yes	Yes	Yes	None	Textual content of element, <i>including</i> child elements, but excluding attributes	local , global
Localization Quality Rating (localization- quality-rating)	Yes	Yes	Yes	Yes	None	Textual content of element, <i>including</i> child elements, but excluding attributes	local
MT Confidence (mt-confidence)	Yes	Yes	Yes	No	None	Textual content of element, <i>including</i> child elements, but excluding attributes	local , global
Allowed Characters (allowed- characters)	Yes	Yes	Yes	Yes	None	Textual content of element, <i>including</i> child elements, but excluding attributes	local , global
Storage Size (storage-size)	Yes	Yes	Yes	Yes	None	None	local , global

Example 30: Defaults, inheritance and overriding behavior of data categories

In this example, the content of all the `data` elements is translatable because the default for the `Translate` data category in elements is "yes". The content of `revision` and `locNote` is not translatable because the default is overridden by the local `its:translate="no"` attribute in the `prolog` element, and that value is inherited by all the children of `prolog`.

The localization note for the two first `data` elements is the text defined globally with the `locNoteRule` element. And this note is overridden for the last `data` element by the local `locNote` attribute.

```
<Res xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <prolog its:translate="no">
    <revision>Sep-07-2006</revision>
    <its:rules version="2.0">
      <its:translateRule selector="//msg/type" translate="no"/>
      <its:locNoteRule locNoteType="description" selector="//msg/data">
        <its:locNote>The variable {0} is the name of the host.</its:locNote>
      </its:locNoteRule>
    </its:rules>
  </prolog>
  <body>
    <msg id="HostNotFound">
      <type>Error</type>
      <data>Host {0} cannot be found.</data>
    </msg>
    <msg id="HostDisconnected">
      <type>Error</type>
      <data>The connection with {0} has been lost.</data>
    </msg>
    <msg id="FileNotFound">
      <type>Error</type>
      <data its:locNote="{0} is a filename">{0} not found.</data>
    </msg>
  </body>
</Res>
```

[Source file: <examples/xml/EX-datacat-behavior-1.xml>]

Note:

The data categories differ with respect to defaults. This is due to existing standards and practices. It is common practice for example that information about translation refers only to textual content of an element. Thus, the default selection for the [Translate](#) data category is the textual content.

8.2 Translate

8.2.1 Definition

The [Translate](#) data category expresses information about whether the content of an element or attribute should be translated or not. The values of this data category are "yes" (translatable) or "no" (not translatable).

8.2.2 Implementation

The [Translate](#) data category can be expressed with global rules, or locally on an individual element. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements, but *excluding* attributes. The default is that elements are translatable and attributes are not.

GLOBAL: The `translateRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `translate` attribute with the value "yes" or "no".

Example 31: The [Translate](#) data category expressed globally

The `translateRule` element specifies that the elements `code` must not be translated.

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:translateRule translate="no" selector="//code"/>
</its:rules>
```

[Source file: [examples/xml/EX-translate-selector-1.xml](#)]

LOCAL: The following local markup is available for the [Translate](#) data category:

- A `translate` attribute with the value "yes" or "no".

Note:

It is not possible to override the [Translate](#) data category settings of attributes using local markup. This limitation is consistent with the advised practice of not using translatable attributes. If attributes need to be translatable (e.g., an HTML `alt` attribute), then this must be declared globally.

Example 32: The [Translate](#) data category expressed locally

The local `its:translate="no"` specifies that the content of `panelmsg` must not be translated.

```
<messages its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <msg num="123">Click Resume Button on Status Display or <panelmsg its:translate="no">
    >CONTINUE</panelmsg> Button on printer panel</msg>
</messages>
```

[Source file: [examples/xml/EX-translate-selector-2.xml](#)]

Example 33: The [Translate](#) data category expressed locally in HTML

The local `translate="no"` attribute specifies that the content of `span` must not be translated.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Translate flag test: Default</title>
  </head>
  <body>
    <p>The <span translate=no>World Wide Web Consortium</span> is
      making the World Wide Web worldwide!</p>
  </body>
</html>
```

[Source file: [examples/html5/EX-translate-html5-local-1.html](#)]

8.3 Localization Note



8.3.1 Definition



The [Localization Note](#) data category is used to communicate notes to localizers about a particular item of content.

This data category can be used for several purposes, including, but not limited to:

- Tell the translator how to translate parts of the content
- Expand on the meaning or contextual usage of a specific element, such as what a variable refers to or how a string will be used in the user interface
- Clarify ambiguity and show relationships between items sufficiently to allow correct translation (e.g., in many languages it is impossible to translate the word "enabled" in isolation without knowing the gender, number and case of the thing it refers to.)
- Indicate why a piece of text is emphasized (important, sarcastic, etc.)

Two types of informative notes are needed:

- An alert contains information that the translator must read before translating a piece of text. Example: an instruction to the translator to leave parts of the text in the source language.
- A description provides useful background information that the translator will refer to only if they wish. Example: a clarification of ambiguity in the source text.

Editing tools may offer an easy way to create this type of information. Translation tools can be made to recognize the difference between these two types of localization notes, and present the information to translators in different ways.

8.3.2 Implementation



The [Localization Note](#) data category can be expressed with global rules, or locally on an individual element. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements, but *excluding* attributes.

GLOBAL: The `locNoteRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `locNoteType` attribute with the value "description" or "alert".
- Exactly one of the following:
 - A `locNote` element that contains the note itself and allows for [local ITS markup](#).
 - A `locNotePointer` attribute that contains a [relative selector](#) pointing to a node that holds the localization note.
 - A `locNoteRef` attribute that contains an IRI referring to the location of the localization note.
 - A `locNoteRefPointer` attribute that contains a [relative selector](#) pointing to a node that holds the IRI referring to the location of the localization note.

Example 34: The `locNote` element

The `locNoteRule` element associates the content of the `locNote` element with the message with the identifier 'DisableInfo' and flags it as important. This would also work if the rule was in an external file, allowing to provide notes without modifying the source document.

```

<myRes>
  <head>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0" its:transl
      <its:locNoteRule locNoteType="alert" selector="//msg[@id='DisableInfo']">
        <its:locNote>The variable {0} has three possible values: 'printer', 'stac
          options'.</its:locNote>
      </its:locNoteRule>
    </its:rules>
  </head>
  <body>
    <msg id="DisableInfo">The {0} has been disabled.</msg>
  </body>
</myRes>

```

[Source file: [examples/xml/EX-locNote-element-1.xml](#)]

Example 35: The `locNotePointer` attribute

The `locNotePointer` attribute is a [relative selector](#) pointing to a node that holds the note.

```

<Res>
  <prolog>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:translateRule selector="//msg/notes" translate="no"/>
      <its:locNoteRule locNoteType="description" selector="//msg/data" locNotePoi
    </its:rules>
  </prolog>
  <body>
    <msg id="FileNotFound">
      <notes>Indicates that the resource file {0} could not be loaded.</notes>
      <data>Cannot find the file {0}.</data>
    </msg>
    <msg id="DivByZero">
      <notes>A division by 0 was going to be computed.</notes>
      <data>Invalid parameter.</data>
    </msg>
  </body>
</Res>

```

[Source file: [examples/xml/EX-locNotePointer-attribute-1.xml](#)]

Example 36: The `locNoteRef` attribute

The `locNoteRule` element specifies that the message with the identifier 'NotFound' has a corresponding explanation note in an external file. The IRI for the exact location of the note is stored in the `locNoteRef` attribute.

```

<myRes>
  <head>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:locNoteRule locNoteType="description" selector="//msg[@id='NotFound']"
        locNoteRef="ErrorsInfo.html#NotFound"/>
    </its:rules>
  </head>
  <body>
    <msg id="NotFound">Cannot find {0} on {1}.</msg>
  </body>

```

```

</body>
</myRes>

```

[Source file: <examples/xml/EX-locNoteRef-attribute-1.xml>]

Example 37: The `locNoteRefPointer` attribute

The `locNoteRefPointer` attribute contains a [relative selector](#) pointing to a node that holds the IRI referring to the location of the note.

```

<dataFile>
  <prolog>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:locNoteRule locNoteType="description" selector="//data"
        locNoteRefPointer="../@noteFile"/>
    </its:rules>
  </prolog>
  <body>
    <string id="FileNotFound" noteFile="Comments.html#FileNotFound">
      <data>Cannot find the file {0}.</data>
    </string>
    <string id="DivByZero" noteFile="Comments.html#DivByZero">
      <data>Invalid parameter.</data>
    </string>
  </body>
</dataFile>

```

[Source file: <examples/xml/EX-locNoteRefPointer-attribute-1.xml>]

LOCAL: The following local markup is available for the [Localization Note](#) data category:

- One of the following:
 - A `locNote` attribute that contains the note itself.
 - A `locNoteRef` attribute that contains an IRI referring to the location of the localization note.
- An optional `locNoteType` attribute with the value "description" or "alert". If the `locNoteType` attribute is not present, the type of localization note will be assumed to be "description".

Example 38: The [Localization Note](#) data category expressed locally

```

<msgList xmlns:its="http://www.w3.org/2005/11/its" xml:space="preserve" its:versi
  <data name="LISTFILTERS_VARIANT" its:locNote="Keep the leading space!" its:locNo
    <value> Variant {0} = {1} ({2})</value>
  </data>
  <data its:locNote="%1$s is the original text's date in the format YYYY-MM-DD HH
    <value>Translated from English content dated <span id="version-info">%1$s</spa
  </data>
</msgList>

```

[Source file: <examples/xml/EX-locNote-selector-2.xml>]

Example 39: The [Localization Note](#) data category expressed locally in HTML

```

<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>LocNote test: Default</title>
  </head>
  <body>
    <p>This is a <span its-loc-note="Check with terminology engineer" its-loc-not
  </body>
</html>

```

[Source file: <examples/html5/EX-locNote-html5-local-1.html>]

Note:

It is generally recommended to avoid using attributes to store text, however, in this specific case, the need to provide the notes without interfering with the structure of the host document is outweighing the drawbacks of using an attribute.

8.4 Terminology

8.4.1 Definition

The [Terminology](#) data category is used to mark terms and optionally associate them with information, such as definitions. This helps to increase consistency across different parts of the documentation. It is also helpful for translation.

Note:

Existing terminology standards such as [\[ISO 30042\]](#) and its derived formats are about coding terminology data, while the ITS [Terminology](#) data category simply allows to identify terms in XML documents and optionally to point to corresponding information.

8.4.2 Implementation

The [Terminology](#) data category can be expressed with global rules, or locally on an individual element. There is no inheritance. The default is that neither elements nor attributes are terms.

GLOBAL: The `termRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `term` attribute with the value "yes" or "no".
- None or exactly one of the following:
 - A `termInfoPointer` attribute that contains a [relative selector](#) pointing to a node that holds the terminology information.
 - A `termInfoRef` attribute that contains an IRI referring to the resource providing information about the term.
 - A `termInfoRefPointer` attribute that contains a [relative selector](#) pointing to a node that holds the IRI referring to the location of the terminology information.

Example 40: Usage of the `termInfoPointer` attribute

```

<text>
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <its:termRule selector="//term" term="yes" termInfoPointer="id(@def)"/>
  </its:rules>
  <p>We may define <term def="TDPV">discoursal point of view</term> as
    <gloss xml:id="TDPV">the relationship, expressed through discourse structure,
      implied author or some other addresser, and the fiction.</gloss></p>
</text>

```

[Source file: <examples/xml/EX-terms-selector-1.xml>]

Example 41: Usage of the `termInfoRef` attribute

```

<text>
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <its:termRule selector="//term[1]" term="yes"
      termInfoRef="#TDPV"/>
  </its:rules>
  <p>We may define <term>discoursal point of view</term>
as <gloss xml:id="TDPV">the relationship, expressed through discourse
structure, between the implied author or some other addresser,
and the fiction.</gloss></p>
</text>

```

[Source file: <examples/xml/EX-terms-selector-2.xml>]

Example 42: Usage of the `termInfoRefPointer` attribute

```

<text>
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <its:termRule selector="//term" term="yes"
      termInfoRefPointer="@target"/>
  </its:rules>
  <p>We may define <term target="#TDPV">discoursal point of view</term>
as <gloss xml:id="TDPV">the relationship, expressed through discourse
structure, between the implied author or some other addresser,
and the fiction.</gloss></p>
</text>

```

[Source file: <examples/xml/EX-terms-selector-3.xml>]

LOCAL: The following local markup is available for the [Terminology](#) data category:

- A `term` attribute with the value "yes" or "no".
- An optional `termInfoRef` attribute that contains an IRI referring to the resource providing information about the term.
- An optional `termConfidence` attribute with the value of a rational number in the interval 0 to 1 (inclusive). The value follows the [XML Schema decimal data type](#) with the constraining facets `minInclusive` set to 0 and `maxInclusive` set to 1. `termConfidence` represents the confidence of the agents producing the annotation that the values of the term and, where provided, `termInfoRef`, are accurate. 1 represents the highest level of confidence.

Any node selected by the terminology data category with the `termConfidence` attribute specified **MUST** be contained in an element with the `annotatorsRef` (or in HTML `its-annotators-ref`)

attribute specified for the [Terminology](#) data category. See [Section 5.8: ITS Tools Annotation](#) for more information.

Example 43: The [Terminology](#) data category expressed locally, including term information reference and confidence score.

```
<book its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its" its:annotatorsR
  <head>...</head>
  <body>
    ...
    <p>And he said: you need a new
      <quote its:term="yes"
        its:termInfoRef="http://www.directron.com/motherboards1.html" its:termCon
      ...
    </body>
  </book>
```

[Source file: [examples/xml/EX-terms-selector-4.xml](#)]

Example 44: The [Terminology](#) data category expressed locally in HTML

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Terminology test: default</title>
  </head>
  <body>
    <p>We need a new <span its-term=yes>motherboard</span>
    </p>
  </body>
</html>
```

[Source file: [examples/html5/EX-term-html5-local-1.html](#)]

8.5 Directionality



This section is *informative*.

Note:

As time of writing, directionality is not clearly defined in HTML, and no implementation commitment is seen for the [Directionality](#) data category in ITS 2.0. Hence this data category is defined as informative, creating a non-backward compatibly change to ITS 1.0. This note and this section may be updated with the proper guidance if the HTML definition is stabilized before ITS 2.0 moves to proposed recommendation status. Nevertheless, to be able to move to last call, the [Directionality](#) data category will not be defined as a normative feature of ITS 2.0.

8.5.1 Definition



The [Directionality](#) data category allows the user to specify the base writing direction of blocks, embeddings and overrides for the Unicode bidirectional algorithm. It has four values: "ltr", "rtl", "lro" and "rlo".

Note:

ITS defines only the values of the [Directionality](#) data category and their inheritance. The behavior of text labeled in this way may vary, according to the implementation. Implementers are encouraged, however, to model the behavior on that described in the CSS 2.1 specification or its successor. In such a case, the effect of the data category's values would correspond to the following CSS rules:

- Data category value: "ltr" (left-to-right text)
CSS rule: `*[dir="ltr"] { unicode-bidi: embed; direction: ltr }`
- Data category value: "rtl" (right-to-left text)
CSS rule: `*[dir="rtl"] { unicode-bidi: embed; direction: rtl }`
- Data category value: "lro" (left-to-right override)
CSS rule: `*[dir="lro"] { unicode-bidi: bidi-override; direction: ltr }`
- Data category value: "rlo" (right-to-left text)
CSS rule: `*[dir="rlo"] { unicode-bidi: bidi-override; direction: rtl }`

More information about how to use this data category is provided by [\[Bidi Article\]](#).

8.5.2 Implementation



The [Directionality](#) data category can be expressed with global rules, or locally on an individual element. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements and attributes. The default is that both elements and attributes have the directionality of left-to-right.

GLOBAL: The `dirRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `dir` attribute with the value "ltr", "rtl", "lro" or "rlo".

Example 45: Document which needs global rules for directionality

In this document the right-to-left directionality is marked using a `direction` attribute with a value "rtlText".

```
<text xml:lang="en">
  <body>
    <par>In Hebrew, the title <quote xml:lang="he" direction="rtlText">פתיחת
    הבינאום , W3C</quote> means <quote>Internationalization Activity,
    W3C</quote>.</par>
  </body>
</text>
```

[Source file: [examples/xml/EX-dir-selector-1.xml](#)]

Example 46: The [Directionality](#) data category expressed with global rules

The `dirRule` element indicates that all elements with an attribute `direction="rtlText"` have right-to-left content.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
  <its:dirRule dir="rtl" selector="//*[@direction='rtlText']"/>
</its:rules>
```

[Source file: <examples/xml/EX-dir-selector-2.xml>]

LOCAL: The following local markup is available for the [Directionality](#) data category:

- A `dir` attribute with the value "ltr", "rtl", "lro" or "rlo".

Example 47: The [Directionality](#) data category expressed locally

On the first `quote` element, the `its:dir="rtl"` attribute indicates a right-to-left content.

```
<text xml:lang="en" xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <body>
    <par>In Arabic, the title <quote xml:lang="ar" its:dir="rtl">نشاط التدويل،
      W3C</quote> means <quote>Internationalization Activity,
      W3C</quote>.</par>
  </body>
</text>
```

[Source file: <examples/xml/EX-dir-selector-3.xml>]

Example 48: The [Directionality](#) data category expressed locally in HTML

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Dir test: Default</title>
  </head>
  <body>
    <p>In Arabic, the title <quote dir=rtl lang=ar>نشاط التدويل، W3C</quote>
      means <quote>Internationalization Activity, W3C</quote>.</p>
  </body>
</html>
```

[Source file: <examples/html5/EX-dir-html5-local-1.html>]

8.6 Ruby



This section is *informative*.

Note:

As time of writing, ruby is not clearly defined in HTML, and no implementation commitment is seen for the [Ruby](#) data category in ITS 2.0. Hence this data category is defined as informative, creating a non-backward compatibly change to ITS 1.0. This note and this section may be updated with the proper guidance if the HTML definition is stabilized before ITS 2.0 moves to proposed recommendation status. Nevertheless, to be able to move to last call, the [Ruby](#) data category will not be defined as a normative feature of ITS 2.0.

8.6.1 Definition



The [Ruby](#) data category is used for a run of text that is associated with another run of text, referred to as the base text. Ruby text is used to provide a short annotation of the associated base text. It is most often used to provide a reading (pronunciation) guide.

8.6.2 Implementation



[Ed. note: Examples for HTML need to be added;]

The [Ruby](#) data category can be expressed with global rules, or locally. There is no inheritance.

GLOBAL: The `rubyRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies. This is the ruby base text.
- An optional `rubyPointer` attribute that contains a [relative selector](#) pointing to a node that corresponds to the ruby element.
- An optional `rpPointer` attribute that contains a [relative selector](#) pointing to a node that corresponds to the ruby parenthesis.
- An optional `rubyText` element that contains the ruby text.
- An optional `rtPointer` attribute that contains a [relative selector](#) pointing to a node that corresponds to the ruby text.

Note:

Where legacy formats do not contain ruby markup, it is still possible to associate ruby text with a specified range of document content using the `rubyRule` element.

Example 49: Adding ruby text with a `rubyRule` element

```
<text xmlns:its="http://www.w3.org/2005/11/its">
  <head>
    ...
    <its:rules version="2.0">
      <its:rubyRule selector="/text/body/img[1]/@alt">
        <its:rubyText>World Wide Web Consortium</its:rubyText>
      </its:rubyRule>
    </its:rules>
  </head>
  <body>
    
    ...
  </body>
</text>
```

[Source file: [examples/xml/EX-ruby-legacy-1.xml](#)]

LOCAL: In a document, the [Ruby](#) data category is realized with a `ruby` element. It contains the following:

[Ed. note: Not sure if the following is correct and understandable. Also Ruby model was recently extended in HTML5, we should align to this probably.]

- The ruby base text or `span` element that contains the ruby base text and allows for [local ITS markup](#).
- An `rp` element that contains the ruby parenthesis. It is used in case of simple markup to specify characters that can denote the beginning and end of ruby text when user agents do not have other ways to present ruby text distinctively from the base text.
- An `rt` element that contains the ruby text and allows for [local ITS markup](#).

All these elements share the attributes of the `span` element.

Example 50: The `Ruby` data category expressed locally

```
<text its:version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <head> ... </head>
  <body>
    <p>この本は
      <its:ruby>
        慶?義塾大学
        <its:rp>(</its:rp><its:rt>けいおうぎじゅくだいがく
      </its:rt><its:rp></its:rp>
      </its:ruby>の歴史を?明するものです。</p>
    </body>
  </text>
```

[Source file: [examples/xml/EX-ruby-implementation-1.xml](#)]

Note:

The structure of the content model for the `ruby` element is identical with the structure of ruby markup as defined in [\[HTML5\]](#).

8.7 Language Information



8.7.1 Definition



The element `langRule` is used to express the language of a given piece of content. The `langPointer` attribute points to the markup which expresses the language of the text selected by the selector attribute. This markup **MUST** use values that conform to [\[BCP47\]](#). The recommended way to specify language identification is to use `xml:lang` in XML, and `lang` in HTML. The `langRule` element is intended only as a fall-back mechanism for documents where language is identified with another construct.

Example 51: Pointing to language information via `langRule`

The following `langRule` element expresses that the content of all `p` elements (including attribute values and textual content of child elements) are in the language indicated by `mylangattribute`, which is attached to the `p` elements, and expresses language using values conformant to [\[BCP47\]](#).

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:langRule selector="//p" langPointer="@mylangattribute"/>
</its:rules>
```

[Source file: [examples/xml/EX-lang-definition-1.xml](#)]

Note:

The [Language Information](#) data category only provides for rules to be expressed at a global level. Locally users are able to use `xml:lang` (which is defined by XML), or `lang` in HTML, or an attribute specific to the format in question (as in [Example 51](#)).

In XML `xml:lang` is the preferable means of language identification. To ease the usage of `xml:lang`, a declaration for this attribute is part of the non-normative XML DTD and XML Schema document for ITS markup declarations. There is no declaration of `xml:lang` in the non-normative RELAX NG document for ITS, since in RELAX NG it is not necessary to declare attributes from the XML namespace.

Applying the [Language Information](#) data category to `xml:lang` attributes using global rules is not necessary, since `xml:lang` is the standard way to specify language information in XML. `xml:lang` is defined in terms of [RFC 3066 or its successor \(BCP47\)](#) is the "Best Common Practice" for language identification and encompasses [RFC 3066](#) and its successors.)

In HTML `lang` is the mandated means of language identification.

8.7.2 Implementation

The [Language Information](#) data category can be expressed only with global rules. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements and attributes. There is no default.

GLOBAL: The `langRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `langPointer` attribute that contains a [relative selector](#) pointing to a node that contains language information.

8.8 Elements Within Text

8.8.1 Definition

The [Elements Within Text](#) data category reveals if and how an element affects the way text content behaves from a linguistic viewpoint. This information is for example relevant to provide basic text segmentation hints for tools such as translation memory systems. The values associated with this data category are:

- "yes" : The element and its content are part of the flow of its parent element. For example the element `strong` in [XHTML 1.0](#):

```
<strong>Appaloosa horses</strong> have spotted coats.
```
- "nested" : The element is part of the flow of its parent element, its content is an independent flow. For example the element `fn` in [DITA 1.0](#):

```
Palouse horses<fn>A Palouse horse is the same as an Appaloosa.</fn>
have spotted coats.
```
- "no" : The element splits the text flow of its parent element and its content is an independent text flow. For example the element `p` when inside the element `li` in DITA or XHTML:

```
<li>Palouse horses: <p>They have spotted coats.</p> <p>They have
been bred by the Nez Perce.</p> </li>
```

8.8.2 Implementation

The [Elements Within Text](#) data category can be expressed with global rules, or locally on an individual element. There is no inheritance. The default is that elements are not within text.

GLOBAL: The `withinTextRule` element contains the following:

- A required **selector** attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required **withinText** attribute with the value "yes", "no" or "nested".

Example 52: Specifying elements within text with a **withinTextRule** element

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
  <its:withinTextRule withinText="yes" selector="//b | //em | //i"/>
</its:rules>
```

[Source file: [examples/xml/EX-within-text-implementation-1.xml](#)]

LOCAL: The following local markup is available for the [Elements Within Text](#) data category:

- A **withinText** attribute with the values "yes", "no" or "nested".

Example 53: The [Elements Within Text](#) data category expressed locally

```
<text xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <body>
    <par>Text with <b bold its:withinText="yes">bald</b>.</par>
  </body>
</text>
```

[Source file: [examples/xml/EX-within-text-local-1.xml](#)]

Example 54: The [Elements Within Text](#) data category expressed locally in HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Within text test: Default</title>
  </head>
  <body>
    <p>Text with <span its-within-text='yes'>bald</span>.</p>
  </body>
</html>
```

[Source file: [examples/html5/EX-within-text-local-html5-1.html](#)]

8.9 Domain



8.9.1 Definition



The [Domain](#) data category is used to identify the topic or subject of a given content. Such information allows to make more relevant linguistic choices during various processes.

Examples of usage include:

- Allowing machine translation systems to select the most appropriate engine and rules to translate the content.
- Providing a general indication of what terminology collection should be used by a translator.

This data category addresses various challenges:

- Often domain-related information already exist in the document (e.g. keywords in the HTML `meta` element). The [Domain](#) data category provides a mechanism to point to this information.
- There are many flat or structured lists of domain related values, keywords, key phrases, classification codes, ontologies, etc. The [Domain](#) data category does not propose its own given list. Instead it provides a mapping mechanism to associate the values in the document with the values used by the consumer tool.

8.9.2 Implementation



The [Domain](#) data category can be expressed only with global rules. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements and attributes. There is no default.

The information provided by this data category is a comma-separated list of one or more values which is obtained by applying the following algorithm:

- STEP 1: Set the initial value of the resulting string as an empty string.
- STEP 2: Get the list of nodes resulting of the evaluation of the `domainPointer` attribute.
- STEP 3: For each node:
 - STEP 3-1: If the node value contains a COMMA (U+002C):
 - STEP 3-1-1: Split the node value into separate strings using the COMMA (U+002C) as separator.
 - STEP 3-1-2: For each string:
 - STEP 3-1-2-1: Trim the leading and trailing white spaces of the string.
 - STEP 3-1-2-2: If the first character of the value is an APOSTROPHE (U+0027) or a QUOTATION MARK (U+0022): Remove it.
 - STEP 3-1-2-3: If the last character of the value is an APOSTROPHE (U+0027) or a QUOTATION MARK (U+0022): Remove it.
 - STEP 3-1-2-4: If the value is empty: Go to STEP 3-1-2.
 - STEP 3-1-2-5: Check if there is a mapping for the string (the mapping is case-insensitive):
 - STEP 3-1-2-5-1. If a mapping is found: Add the corresponding value to the result string.
 - STEP 3-1-2-5-2. Else (if no mapping is found): Add the string (in its original cases) to the result string.
 - STEP 3-2: Else (if the node value does not contain a COMMA (U+002C)):
 - STEP 3-2-1: Trim the leading and trailing white spaces of the string.
 - STEP 3-2-2: If the first character of the value is an APOSTROPHE (U+0027) or a QUOTATION MARK (U+0022): Remove it.
 - STEP 3-2-3: If the last character of the value is an APOSTROPHE (U+0027) or a QUOTATION MARK (U+0022): Remove it.
 - STEP 3-2-4: If the value is empty: Go to STEP 3.
 - STEP 3-2-5: Check if there is a mapping for the string (the mapping is case-insensitive):
 - STEP 3-2-5-1: If a mapping is found: Add the corresponding value to the result string.
 - STEP 3-2-5-2: Else (if no mapping is found): Add the string (in its original cases) to the result string.
- STEP 4: Remove duplicated values from the resulting string.

STEP 5: Return the resulting string.

GLOBAL: The `domainRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `domainPointer` attribute that contains a [relative selector](#) pointing to a node that contains the domain information.
- An optional `domainMapping` attribute that contains a comma separated list of mappings between values in the content and consumer tool specific values. The left part of the pair corresponds to the source content and is unique within the mapping and case-insensitive. The right part of the mapping belongs to the consumer tool. Several left parts can map to a single right part. The values in the left or the right part of the mapping may contain spaces; in that case they **MUST** be delimited by quotation marks, that is pairs of APOSTROPHE (U+0027) or QUOTATION MARK (U+0022).

Note:

Although the `domainMapping` attribute it is optional, its usage is recommended. Many commercial machine translation systems use their own domain definitions; the `domainMapping` attribute will foster interoperability between these definitions and metadata items like `keywords` or `dcterms.subject` in Web pages or other types of content.

Values used in the `domainMapping` attribute are arbitrary strings. In some consumer systems or existing content, the domain may be identified via an IRI like `http://example.com/domains/automotive`. The `domainMapping` allows for using IRIs too. For the mapping, they are regarded as ordinary string values.

Example 55: The `domainRule` element

The `domainRule` element expresses that the content of the HTML `body` element is in the domain expressed by the HTML `meta` element with the `name` attribute, value `keywords`. The `domainPointer` attribute points to that `meta` element.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
  xmlns:h="http://www.w3.org/1999/xhtml">
  <its:domainRule selector="/h:html/h:body"
    domainPointer="/h:html/h:head/h:meta[@name='keywords']"/>
</its:rules>
```

[Source file: [examples/xml/EX-domain-1.xml](#)]

Example 56: The `domainRule` element

The `domainRule` element expresses that the content of the HTML `body` element is in the domain expressed by associated values. The `domainPointer` attribute points to the values in the source content. In this case it points to the `meta` elements with the `name` attribute set to `"keywords"` or to `"dcterms.subject"`. These elements hold the values in their `content` attributes. The `domainMapping` attribute contains the comma separated list of mappings. In the example, `"automotive"` is available in the source content, and `"auto"` is used within the consumer tool, e.g. a machine translation system.

```
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
  xmlns:h="http://www.w3.org/1999/xhtml">
  <its:domainRule selector="/h:html/h:body"
    domainPointer="/h:html/h:head/h:meta[@name='dcterms.subject' or @name='keywor
```

```
domainMapping="automotive auto, medical medicine, 'criminal law' law, 'proper
</its:rules>
```

[Source file: <examples/xml/EX-domain-2.xml>]

Note:

In HTML the preferred way to express domain information is a `meta` element with the `name` attribute set to "keywords", see [standard metadata names in HTML](#). Alternatively, following the process for [other metadata names](#) the [extension value](#) of "dcterms.subject" can be used. The usage of both "keywords" and "dcterms.subject" is shown in example [Example 56](#).

In the area of machine translation (e.g. machine translation systems or systems harvesting content for machine translation training), there is no agreed upon set of value sets for domain. Nevertheless it is recommended to use a small set of values both in source content and within consumer tools, to foster interoperability. If larger value sets are needed (e.g. detailed terms in the law or medical domain), mappings to the smaller value set needed for interoperability should be provided. An example would be a `domainMapping` attribute for generalizing the law domain: `domainMapping="'criminal law' law, 'property law' law, 'contract law' law"`.

It is possible to have more than one domain associated with a piece of content. For example, if the consumer tool is a statistical machine translation engine, it could include corpora from all domains available in the source content in training the machine translation engine.

The consumer machine translation engine might choose to ignore the domain and take a one size fits all approach, or may be selective in which domains to use, based on the range of content marked with domain. For example, if the content has hundreds of sentences marked with domain "automotive" and "medical", but only a couple of sentences marked with additional domains "criminal law" and "property law", the consumer tool may opt to include its domains "auto" and "medicine", but not "law", since the extra training resources does not justify the improvement in the output.

8.10 Disambiguation



8.10.1 Definition



The [Disambiguation](#) data category is used to highlight (mark up) specific conceptual patterns that may require special treatment when localizing and translating content.

This data category can be used for several purposes, including, but not limited to:

- Informing a translation service that a certain fragment of text is subject to follow specific translation rules, e.g. for proper names, or officially regulated translations, as well as to conveying a very specific meaning of the fragment.
- Informing content management systems and translation services about the intended conceptual type of a textual entity in order to enable processing based on this specific type for source and target languages, for example, when dealing with personal names, product names, or geographic names, chemical compounds, protein names, and so forth.

Note:

The use case for disambiguation is distinct from that for the [Terminology](#) data category. Disambiguation may directly inform human and automated translation activities in settings where either explicit terminology information is not (yet) available or would be not appropriate (general

language case). The two data categories may also be complementary, e.g. when automatically generated disambiguation annotation provides input to a manual or automated term mining process that results in Terminology annotations.

Disambiguation support is achieved by associating a marked up fragment of text with an external web resource that can be dereferenced by a language review agent, i.e. by accessing the intended meaning or lexical choice of the fragment, and thereby contributing to its correct translation.

A fragment of text is disambiguated at different granularities: (1) lexical type, (2) ontological concept, or (3) named entity.

In the case of lexical type, the external resource may provide appropriate synonyms and example usage, such as what WordNet services do.

In the case of ontological concept, the external resource may provide a formalized conceptual definition arranged in a hierarchical framework of related concepts.

In the case of a named entity, the external resource may provide a fully fledged description of the associated real world entity. For instance, the word 'City' in the fragment 'I am going to the City' may be disambiguated on the basis of one of WordNet's synsets that can be represented by 'city', an ontological concept of 'City' that could represent a subclass of 'Populated Place' at the conceptual granularity level, or the central area of a particular city, e.g. 'City of London', as interpreted at the entity granularity level.

Linked data networks, such as DBpedia, further increase the interlinking of ontological concepts and named entity definitions for same things and in different languages, thereby offering the possibility to directly facilitate translation through a source language description.

Two types of disambiguation are possible:

- Disambiguation for target type class, which explicitly describes the type class of the underlying concept or entity of the fragment.
- Disambiguation for target identity, which implicitly describes the intended meaning of the fragment through a link to an external resource.

Text analysis engines, such as named entity recognizers, named entity, concept and word sense disambiguation components do offer appropriate solutions to create the needed information. Content management systems are also able to present and visualize this information, or employ it to index their content. Machine translation services may use this information for optimizing their language and translation models.

8.10.2 Implementation



The [Disambiguation](#) data category can be expressed with global rules, or locally on an individual element. There is no inheritance.

When using disambiguation specifying the target identity, the user **MUST** use only one of the two addressing modes:

1. Using `disambigSource` and one of `disambigIdent` or `disambigIdentPointer` (at a global rule) to specify the collection and the identifier itself.
2. Using one of `disambigIdentRef` or `disambigIdentRefPointer` (at a global rule) using an IRI for the disambiguation target.

GLOBAL: The `disambiguationRule` element contains the following:

- A required **selector** attribute that contains an [absolute selector](#) which selects the nodes to which this rule applies.
- An optional **disambigGranularity** attribute that contains a string, specifying the granularity level of the disambiguation. The value **MUST** be one of the following identifiers: "lexical-concept", "ontology-concept", or "entity". The default value is "entity".
- At least one of the following:
 - To specify the target type class, exactly one of the following:
 - A **disambigClassPointer** attribute that contains a [relative selector](#) pointing to a node specifying the type of entity or concept class behind the selector.
 - A **disambigClassRefPointer** attribute that contains a [relative selector](#) pointing to a node that holds an IRI that specifies the type of entity or concept class behind the selector.
 - To specify the target identity, exactly one of the following:
 - When using the addressing [mode 1](#):
 - A **disambigSourcePointer** attribute that contains a [relative selector](#) to a node that holds the string representing the disambiguation identifier collection source.
 - A **disambigIdentPointer** attribute that contains a [relative selector](#) to a node that holds the string, representing the disambiguation identifier for the disambiguation target that is valid within the specified disambiguation source.
 - When using the addressing [mode 2](#):
 - A **disambigIdentRefPointer** attribute that contains a [relative selector](#) pointing to a node that holds an IRI that represents a unique identifier for the disambiguation target.

For an example, see [Example 59](#).

LOCAL: The following local markup is available for the [Disambiguation](#) data category:

- An optional **disambigConfidence** attribute with the value of a rational number in the interval 0 to 1 (inclusive). The value follows the [XML Schema decimal data type](#) with the constraining facets [minInclusive](#) set to 0 and [maxInclusive](#) set to 1. **disambigConfidence** represents the confidence of the agents producing the annotation that the union of the values for the other disambiguation attributes in this instance are accurate. 1 represents the highest level of confidence.
- An optional **disambigGranularity** attribute that contains a string, specifying the granularity level of the disambiguation. The value **MUST** be one of the following identifiers: "lexical-concept", "ontology-concept", or "entity". The default value is "entity".
- At least one of the following:
 - To specify the target type class:
 - A **disambigClassRef** attribute that contains an IRI, specifying the type of entity or concept class behind the selector.
 - To specify the target identity, exactly one of the following:
 - When using the addressing [mode 1](#):
 - A **disambigSource** attribute that contains a string representing the disambiguation identifier collection source.
 - A **disambigIdent** attribute that contains a string, representing the disambiguation identifier for the disambiguation target that is valid within the specified disambiguation source.
 - When using the addressing [mode 2](#):

- A `disambigIdentRef` attribute that contains an IRI that represents a unique identifier for the disambiguation target.

Any node selected by the `disambiguation` data category with the `disambigConfidence` attribute specified **MUST** be contained in an element with the `annotatorsRef` (or in HTML `its-annotators-ref`) attribute specified for the `disambiguation` data category. For more information, see [Section 5.8: ITS Tools Annotation](#).

Example 57: Local mixed usage of Usage of `disambigClassRef`, `disambigGranularity`, and `disambigIdentRef` in HTML.

```
<!DOCTYPE html>
<html lang="en" its-annotators-ref="disambiguation|http://enrycher.ijs.si">
  <head>
    <meta charset="utf-8" />
    <title>Disambiguation: Local Test</title>
  </head>
  <body>
    <p><span
      its-disambig-confidence="0.7"
      its-disambig-class-ref="http://nerd.eurecom.fr/ontology#Place"
      its-disambig-ident-ref="http://dbpedia.org/resource/Dublin"
      its-disambig-granularity="entity">Dublin</span>
    is the <span
      its-disambig-source="Wordnet3.0"
      its-disambig-ident="301467919"
      its-disambig-granularity="lexical-concept"
      its-disambig-confidence="0.5"
      >capital</span> of Ireland.</p>
  </body>
</html>
```

[Source file: [examples/html5/EX-disambiguation-html5-local-1.html](#)]

Note:

For referring to `disambigClassRef` values, implementors are encouraged to use an existing repository of entity types as long as they satisfy their requirements. For example, the Named Entity Recognition and Disambiguation [\[NERD\]](#) ontology.

Furthermore, valid target types depend on the disambiguation granularity: types of entities are distinct from types of lexical concepts or ontology concepts. While this distinction exists, the specification does not prescribe a way of automatically inferring a disambiguation level from a target type.

When serializing the ITS mark-up in HTML, the preferred way is to serialize in RDFa Lite or Microdata due to the existing search and crawling infrastructure that is able to consume this kind of data.

Example 58: Local mixed usage of `disambigClassRefPointer`, `disambigIdentRefPointer`, `disambigGranularity` in HTML+RDFa Lite.

See [Example 59](#) for the companion document with the mapping data.

```
<!DOCTYPE html>
<html lang=en>
  <head>
```

```

<meta charset=utf-8>
<link href=EX-disambiguation-html5-rdfa.xml rel=its-rules>
<title>Entity: Local Test</title>
</head>
<body>
  <p><span property="http://xmlns.com/foaf/0.1/name"
    about="http://dbpedia.org/resource/Dublin"
    typeof="http://nerd.eurecom.fr/ontology#Place">Dublin</span> is
    the capital of Ireland.</p>
</body>
</html>

```

[Source file: <examples/html5/EX-disambiguation-html5-rdfa.html>]

Example 59: Companion document, having the mapping data for [Example 58](#).

```

<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
  <its:disambiguationRule selector="//*[@typeof and @about]"
    disambigClassRefPointer="@typeof" disambigIdentRefPointer="@about" disambigGr
</its:rules>

```

[Source file: <examples/html5/EX-disambiguation-html5-rdfa.xml>]

8.11 Locale Filter

8.11.1 Definition

The [Locale Filter](#) data category specifies that a node is only applicable to certain locales.

This data category can be used for several purposes, including, but not limited to:

- Include a legal notice only in locales for certain regions.
- Drop editorial notes from all localized output.

The [Locale Filter](#) data category associates with each selected node a list of extended language ranges conforming to [\[BCP47\]](#). The list is comma-separated and can include the wildcard extended language range "*". The list can also be empty. Whitespace surrounding language ranges is ignored.

Note:

To express that all locales should be included, one can use the wildcard "*" for the language range. To express that the content should not be included in any local, one can use the empty value.

8.11.2 Implementation

The [Locale Filter](#) data category can be expressed with global rules, or locally on an individual element. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements and attributes. The default is that the language range is "*".

Implementations [MUST NOT](#) combine lists of language ranges from multiple rules or local attributes.

GLOBAL: The `localeFilterRule` element contains the following:

- A required **selector** attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required **localeFilterList** attribute with a comma-separated list of extended language ranges, or an empty string value.

Example 60: The [Locale Filter](#) data category expressed globally

The **localeFilterRule** element specifies that certain legal notice elements should only be shown in the specified locales. Note that using the extended language range **"*-CA"** in the **localeFilterList** attribute would cover all Canadian locales, including various minority languages in Canada.

```
<book xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <info>
    <its:rules version="2.0">
      <its:localeFilterRule
        selector="//legalnotice[@role='Canada']"
        localeFilterList="en-CA, fr-CA"/>
    </its:rules>
    <legalnotice role="Canada">
      <para>This legal notice is only for English and French Canadian locales.</para>
    </legalnotice>
  </info>
</book>
```

[Source file: [examples/xml/EX-locale-filter-selector-1.xml](#)]

Example 61: The [Locale Filter](#) data category expressed globally

The **localeFilterRule** element specifies that editorial remarks should be removed from all translations.

```
<section xmlns:its="http://www.w3.org/2005/11/its">
  <info>
    <its:rules version="2.0">
      <its:localeFilterRule selector="//remark" localeFilterList=""/>
    </its:rules>
  </info>
  <remark>Note: This section will be written later.</remark>
</section>
```

[Source file: [examples/xml/EX-locale-filter-selector-2.xml](#)]

LOCAL: The following local markup is available for the [Locale Filter](#) data category:

- A **localeFilterList** attribute with a comma-separated list of extended language ranges, or an empty string value.

Example 62: The [Locale Filter](#) data category expressed locally

```
<book xmlns:its="http://www.w3.org/2005/11/its">
  <info>
    <legalnotice its:localeFilterList="en-CA, fr-CA">
      <para>This legal notice is only for English and French Canadian locales.</para>
    </legalnotice>
  </info>
</book>
```

[Source file: <examples/xml/EX-locale-filter-attribute-1.xml>]

8.12 Provenance



8.12.1 Definition



The [Provenance](#) data category is used to communicate the identity of agents that have been involved in the translation of the content or the revision of the translated content. This allows translation and translation revision consumers, such as post-editors, translation quality reviewers or localization workflow managers, to assess how the performance of these agents may impact the quality of the translation. Translation and translation revision agents can be identified as a person, a piece of software or an organization that has been involved in providing a translation that resulted in the selected content.

This data category offers three types of information. First, it allows to identify translation agents. Second, it allows to identify revision agents. Third, if provenance information is needed that includes temporal or sequence information about translation processes (e.g. multiple revision cycles) or requires agents that support a wider range of activities, the data category offers a mechanism to refer to external provenance information.

Note:

The specification does not define the format of external provenance information, but it is recommended that an open provenance or change logging format be used, e.g. the W3C provenance data model [\[PROV-DM\]](#).

Translation or translation revision tools, such as machine translation engines or computer assisted translation tools, may offer an easy way to create this information. Translation tools can then present this information to post-editors or translation workflow managers. Web applications may to present such information to consumers of translated documents.

The data category defines seven pieces of information:

Information	Description	Value
Human provenance information	Identification of a human translation agent	A string or an IRI (only for the <code>Ref</code> attributes)
Organisational provenance information	Identification of an organization acting as a translation agent	A string or an IRI (only for the <code>Ref</code> attributes)
Tool related provenance information	Identification of a software tool that was used in translating the selected content	A string or an IRI (only for the <code>Ref</code> attributes)
Human revision provenance information	Identification of a human translation revision agent	A string or an IRI (only for the <code>Ref</code> attributes)
Organisational revision provenance information	Identification of an organization acting as a translation revision agent	A string or an IRI (only for the <code>Ref</code> attributes)
Tool related revision provenance information	Identification of a software tool that was used in revising the translation of the selected content	A string or an IRI (only for the <code>Ref</code> attributes)
Reference to external provenance information	A reference to external provenance information	A space (U+0020) separated list of IRIs

Note:

The tool related provenance and tool related revision provenance pieces of information are not meant to express information about tools used for creating ITS annotations themselves. For this

purpose, ITS 2.0 provides a separate mechanism. See [Section 5.8: ITS Tools Annotation](#) for details, especially the [note on annotatorsRef usage scenarios](#).

8.12.2 Implementation



The [Provenance](#) data category can be expressed with global rules, or locally on individual elements. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements and attributes.

GLOBAL: The `provRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A `provenanceRecordsRefPointer` attribute that contains a [relative selector](#) pointing to a node containing a list of [provenance records](#). These are related to the content selected via the `selector` attribute.

The global rule does not apply to HTML as local markup is provided for direct annotation in HTML.

Example 63: The [Provenance](#) data category used globally with standoff provenance records.

This example expresses provenance information in a standoff manner using `provenanceRecords` elements. The `provRule` element specifies that for any element with a `ref` attribute, that `ref` attribute holds a reference to an associated `provenanceRecords` element where the provenance information is listed. The `legalnotice` element has been revised two times. Hence, the related `provenanceRecords` element contains two `provenanceRecord` child elements.

```
<text xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <dc:creator>John Doe</dc:creator>
  <its:provenanceRecords xml:id="pr1">
    <its:provenanceRecord
      toolRef="http://www.onlinemtex.com/2012/7/25/wsd1/"
      org="acme-CAT-v2.3"
      revToolRef="http://www.mycat.com/v1.0/download"
      revOrg="acme-CAT-v2.3"
      provRef="http://www.examplelsp.com/excontent987/production/prov/e6354"/>
  </its:provenanceRecords>
  <its:provenanceRecords xml:id="pr2">
    <its:provenanceRecord
      person="John Doe"
      orgRef="http://www.legaltrans-ex.com/"
      revPerson="Tommy Atkins"
      revOrgRef="http://www.vistatec.com/"
      provRef="http://www.vistatec.com/job-12-7-15-X31/reviewed/prov/re8573469"/>
    <its:provenanceRecord
      revPerson="John Smith"
      revOrgRef="http://john-smith.qa.example.com"/>
  </its:provenanceRecords>
  <its:rules version="2.0">
    <its:provRule selector="//*[@ref]" provenanceRecordsRefPointer="@ref"/>
  </its:rules>
  <title>Translation Revision Provenance Agent: Global Test in XML</title>
  <body>
    <par ref="#pr1"> This paragraph was translated from the machine.</par>
```

```

<legalnotice postediting-by="http://www.vistatec.com/" ref="#pr2">This text w
  translated directly by a person.</legalnotice>
</body>
</text>

```

[Source file: <examples/xml/EX-provenance-global-1.xml>]

LOCAL: Using the inline markup to represent the data category locally is limited to a single occurrence for a given content (e.g. one cannot have different `toolRef` attributes applied to the same span of text because the inner-most one would override the others). A local *standoff markup* is provided to allow such cases.

The following local markup is available for the [Provenance](#) data category:

- Either (inline markup): at least one of the following attributes:
 - A `person` or `personRef` attribute that implement the [human provenance information](#).
 - An `org` or `orgRef` attribute that implement the [organisational provenance information](#).
 - A `tool` or `toolRef` attribute that implement the [tool related provenance information](#).
 - A `revPerson` or `revPersonRef` attribute that implement the [human revision provenance information](#).
 - A `revOrg` or `revOrgRef` attribute that implement the [organisational revision provenance information](#).
 - A `revTool` or `revToolRef` attribute that implement the [tool related revision provenance information](#).
 - A `provRef` attribute that implements the [reference to external provenance descriptions](#).
- Or (standoff markup):
 - A `provenanceRecordsRef` attribute. Its value is a IRI pointing to the `provenanceRecords` element containing the list of [provenance records](#) related to this content.
 - An element `provenanceRecords` which contains:
 - One or more elements `provenanceRecord`, each of which contains at least one of the following attributes:
 - A `person` or `personRef` attribute that implement the [human provenance information](#).
 - An `org` or `orgRef` attribute that implement the [organisational provenance information](#).
 - A `tool` or `toolRef` attribute that implement the [tool related provenance information](#).
 - A `revPerson` or `revPersonRef` attribute that implement the [human revision provenance information](#).
 - A `revOrg` or `revOrgRef` attribute that implement the [organisational revision provenance information](#).
 - A `revTool` or `revToolRef` attribute that implement the [tool related revision provenance information](#).
 - A `provRef` attribute that implements the [reference to external provenance descriptions](#).

When the attributes `person`, `personRef`, `org`, `orgRef`, `tool`, `toolRef`, `revPerson`, `revPersonRef`, `revOrg`, `revOrgRef`, `revTool`, `revToolRef` and `provRef` are used in a standoff manner, the information they carry pertains to the content of the element that refers to the standoff annotation, not to the content of the element `provenanceRecord` where they are declared.

In HTML the standoff markup **MUST** be stored inside a `script` element. It **MUST** have a `type` attribute with the value `application/its+xml`. Its `id` attribute **MUST** be set to the same value as the `xml:id` attribute of the `provenanceRecords` element it contains.

Example 64: Annotating provenance information in XML with local inline markup

The provenance related attributes at the `par` and `legalnotice` elements are used to associate the provenance information directly with the content of these elements.

```
<text xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <title>Translation Revision Provenance Agent: Local Test in XML</title>
  <body>
    <par its:toolRef="http://www.onlinemt看.com/2012/7/25/wsd1/"
          its:org="acme-CAT-v2.3"
          its:revToolRef="http://www.mycat.com/v1.0/download"
          its:revOrg="acme-CAT-v2.3"
          its:provRef="http://www.example.lsp1.com/prov/e6354 http://www.example.lsp2.com/prov/e6354">
      >This paragraph was translated from the machine.</par>
    <legalnotice its:person="John Doe"
                  its:orgRef="http://www.legaltrans-ex.com/"
                  its:provRef="http://www.examplelsp.com/excontent987/legal/prov/e6354"
                  its:revPerson="Tommy Atkins"
                  its:revOrgRef="http://www.vistatec.com/">
      >This text was translated directly by a person.</legalnotice>
  </body>
</text>
```

[Source file: [examples/xml/EX-provenance-local-1.xml](#)]

Example 65: Annotating provenance information in HTML with local inline markup

In this example several spans of content are associated with provenance information.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Provenance Agent: Local Test in HTML5</title>
  </head>
  <body>
    <p its-tool-ref="http://www.onlinemt看.com/2012/7/25/wsd1/"
      its-org="acme-CAT-v2.3"
      its-prov-ref="http://www.examplelsp.com/excontent987/production/prov/e6354"
      its-rev-org="acme-CAT-v2.3">
      >This paragraph was translated from the machine.</par>
    <p class="legal-notice"
      its-person="John Doe"
      its-org-ref="http://www.legaltrans-ex.com/"
      its-prov-ref="http://www.examplelsp.com/excontent987/legal/prov/e6354"
      its-rev-person="Tommy Atkins" its-rev-org-ref="http://www.vistatec.com/">
      >This text was translated directly by a person.</legalnotice>
```

```
</body>
</html>
```

[Source file: <examples/html5/EX-provenance-html5-local-1.html>]

Example 66: Annotating provenance information in HTML with local standoff markup

The following example shows a document using local standoff markup to encode provenance information. The `p` elements delimits the content to markup. They hold `its-provenance-records-ref` attributes that point to the standoff information inside the `script` elements.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Test</title>
    <script id=pr1 type=application/its+xml>
      <its:provenanceRecords xml:id="pr1" xmlns:its="http://www.w3.org/2005/11/its">
        <its:provenanceRecord
          toolRef="http://www.onlinemtex.com/2012/7/25/wsd1/"
          org="acme-CAT-v2.3"
          provRef="http://www.examplelsp.com/excontent987/production/prov/e6354"/>
          revToolRef="http://www.mycat.com/v1.0/download"
          revOrg="acme-CAT-v2.3" />
        </its:provenanceRecord>
      </its:provenanceRecords>
    </script>
    <script id=pr2 type=application/its+xml>
      <its:provenanceRecords xml:id="pr2" xmlns:its="http://www.w3.org/2005/11/its">
        <its:provenanceRecord
          person="John Doe"
          orgRef="http://www.legaltrans-ex.com/"
          provRef="http://www.examplelsp.com/excontent987/legal/prov/e6354"
          revPerson="Tommy Atkins"
          revOrgRef="http://www.vistatec.com/" />
        <its:provenanceRecord
          revPerson="John Smith"
          revOrgRef="http://john-smith.qa.example.com" />
        </its:provenanceRecord>
      </its:provenanceRecords>
    </script>
  </head>
  <body>
    <p its-provenance-records-ref="#pr1">This paragraph was translated from the m
    <p its-provenance-records-ref="#pr2">This text was translated directly by a p
  </body>
</html>
```

[Source file: <examples/html5/EX-provenance-html5-local-2.html>]

8.13 External Resource



8.13.1 Definition



The [External Resource](#) data category indicates that a node represents or references potentially translatable data in a resource outside the document. Examples of such resources are external images and audio or video files.

8.13.2 Implementation



The [External Resource](#) data category can be expressed only with global rules. There is no inheritance. There is no default.

GLOBAL: The `externalResourceRefRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `externalResourceRefPointer` attribute that contains a [relative selector](#) pointing to a node that provides the IRI of the external resource.

Example 67: The `externalResourceRefRule` element

The `externalResourceRefRule` element expresses that the `imagedata`, `audiodata` and `videodata` elements contain references to external resources. These references are expressed via a `fileref` attribute. The `externalResourceRefPointer` attribute points to that attribute.

```
<doc xmlns:its="http://www.w3.org/2005/11/its"
      xmlns:db="http://docbook.org/ns/docbook">
  <its:rules version="2.0">
    <its:externalResourceRefRule
      selector="//db:imagedata | //db:audiodata | //db:videodata"
      externalResourceRefPointer="@fileref"/>
  </its:rules>
  <db:mediaobject>
    <db:videoobject>
      <db:videodata fileref="movie.avi"/>
    </db:videoobject>
    <db:imageobject>
      <db:imagedata fileref="movie-frame.gif"/>
    </db:imageobject>
    <db:textobject>
      <db:para>This video illustrates the proper way to assemble an inverting
        time distortion device. </db:para>
      <db:warning>
        <db:para>It is imperative that the primary and secondary temporal
          couplings not be mounted in the wrong order. Temporal catastrophe is
          the likely result. The future you destroy may be your own. </db:para>
      </db:warning>
    </db:textobject>
  </db:mediaobject>
</doc>
```

[Source file: [examples/xml/EX-externalresource-1.xml](#)]

Example 68: Two `externalResourceRefRule` elements used for external resources associated with HTML `video` elements

The two `externalResourceRefRule` elements select the `src` and the `poster` attributes at HTML `video` elements. These attributes identify different external resources, and at the same time contain the references to these resources. For this reason, the `externalResourceRefPointer` attributes point to the value of `src` and `poster` respectively. The underlying HTML document is given in [Example 69](#).

```
<its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its"
  xmlns:html="http://www.w3.org/1999/xhtml">
  <its:externalResourceRefRule selector="//html:video/@src"
    externalResourceRefPointer="."/>
  <its:externalResourceRefRule selector="//html:video/@poster"
    externalResourceRefPointer="."/>
</its:rules>
```

[Source file: [examples/xml/EX-externalresource-2.xml](#)]

Example 69: An HTML document that can be used for [Example 68](#).

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Video element example</title>
  </head>
  <body>
    <video
      height=360
      poster=video-image.png
      src=http://www.example.com/video/v2.mp
      width=640>
      <p>If your browser doesn't support the <code>video</code> element, you can
    </video>
  </body>
</html>
```

[Source file: [examples/html5/EX-externalresource-html5-1.html](#)]

8.14 Target Pointer



8.14.1 Definition



Some formats, such as those designed for localization or for multilingual resources, hold the same content in different languages inside a single document. The [Target Pointer](#) data category is used to associate the node of a given source content (i.e. the content to be translated) and the node of its corresponding target content (i.e. the source content translated into a given target language).

This specification makes no provision regarding the presence of the target nodes or their content: A target node may or may not exist and it may or may not have content.

This data category can be used for several purposes, including but not limited to:

- Extract the source content to translate and put back the translation at its proper location.
- Compare source and target content for quality verification.

- Re-use existing translations when localizing the new version of an existing document.
- Access aligned bi-lingual content to build memories, or to train machine translation engines.

Note:

In general, it is recommended to avoid developing formats where the same content is stored in different languages in the same document, unless for very specific use cases. See the best practices “[Working with multilingual documents](#)” from [\[XML i18n BP\]](#) for further guidance.

8.14.2 Implementation

The [Target Pointer](#) data category can be expressed only with global rules. There is no inheritance. There is no default.

GLOBAL: The `targetPointerRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `targetPointer` attribute. It contains a [relative selector](#) that points to the node for the target content corresponding to the selected source node.

Note:

The source node and the target node may be of different types, but the target node must be able to contain the same content of the source node (e.g. an attribute node cannot be the target node of a source node that is an element with children).

Example 70: Defining the target location of a source content with the `targetPointerRule` element

```
<file>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:translateRule selector="/file" translate="no"/>
    <its:translateRule selector="//source" translate="yes"/>
    <its:targetPointerRule selector="//source" targetPointer="../target"/>
  </its:rules>
  <entry id="one">
    <source>Remember last folder</source>
    <target/>
  </entry>
  <entry id="two">
    <source>Custom file filter:</source>
    <target/>
  </entry>
</file>
```

[Source file: [examples/xml/EX-target-pointer-global-1.xml](#)]

8.15 Id Value**8.15.1 Definition**

The [Id Value](#) data category indicates a value that can be used as unique identifier for a given part of the content.

The recommended way to specify a unique identifier is to use `xml:id` or `id` in HTML (See the best practice “[Defining markup for unique identifiers](#)” from [\[XML i18n BP\]](#)). The `idValueRule` element is intended only as a fall-back mechanism for documents where unique identifiers are available with another construct.

Providing a unique identifier that is maintained in the original document can be useful for several purposes, for example:

- Allow automated alignment between different versions of the source document, or between source and translated documents.
- Improve the confidence in leveraged translation for exact matches.
- Provide back-tracking information between displayed text and source material when testing or debugging.

Note:

- The [Id Value](#) data category only provides for rules to be expressed at a global level. Locally, users are able to use `xml:id` (which is defined by XML) or `id` in HTML, or an attribute specific to the format in question (as in [Example 73](#)).
- Applying the [Id Value](#) data category to `xml:id` (in XML) or `id` (in HTML) attributes in global rules is not necessary, since these attributes are the recommended way to specify an identifier.

8.15.2 Implementation



The [Id Value](#) data category can be expressed only with global rules. There is no inheritance. There is no default.

GLOBAL: The `idValueRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `idValue` attribute. It contains an XPath expression which constructs a string corresponding to the identifier of the node to which this rule applies. The identifier **MUST** be unique at least within the document. If the attribute `xml:id` is present or `id` in HTML for the selected node, the value of the `xml:id` attribute or `id` in HTML **MUST** take precedence over the `idValue` value.

Example 71: Pointing to an ID value with the `idValueRule` element

The `idValueRule` element indicates that the unique identifier for each `<text>` element is the value of the attribute `name` of its parent element.

```
<?xml version="1.0"?>
<resources>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:translateRule translate="no" selector="/resources"/>
    <its:translateRule translate="yes" selector="//text"/>
    <its:idValueRule selector="//text" idValue="../@name"/>
  </its:rules>
  <entry name="btn.OK">
    <text>OK</text>
    <pos>1, 1</pos>
    <trig>sendOK</trig>
  </entry>
  <entry name="btn.CANCEL">
```

```

    <text>Cancel</text>
    <pos>2, 1</pos>
    <trig>cancelAll</trig>
  </entry>
</resources>

```

[Source file: <examples/xml/EX-idvalue-element-1.xml>]

Example 72: Constructing ID values using the `idValueRule` element.

The `idValue` attribute allows to build composite values based on different attributes, element or event hard-coded text. Any of the String functions offered by XPath can be used. In the document below, the two elements `<text>` and `<desc>` are translatable, but they have only one corresponding identifier, the `name` attribute in their parent element.

To make sure the identifier is unique for both the content of `<text>` and the content of `<desc>`, the XPath expression `concat(..@name, '_t')` gives the identifier "settingsMissing_t" for the content of `<text>` and the expression `concat(..@name, '_d')` gives the identifier "settingsMissing_d" for the content of `<desc>`.

```

<?xml version="1.0"?>
<doc>
  <its:rules version="2.0" xmlns:its="http://www.w3.org/2005/11/its">
    <its:idValueRule selector="//text" idValue="concat(..@name, '_t')"/>
    <its:idValueRule selector="//desc" idValue="concat(..@name, '_d')"/>
  </its:rules>
  <msg name="settingsMissing">
    <text>Can't find settings file.</text>
    <desc>The module cannot find the default settings file. You need to
      re-initialize the system.</desc>
  </msg>
</doc>

```

[Source file: <examples/xml/EX-idvalue-element-2.xml>]

Example 73: Using `xml:id` and `idValueRule`

When an `xml:id` attribute is present for a node selected by an `idValueRule` element, the value of `xml:id` takes precedence over the value defined by the `idValueRule` element. In the example below, the unique ID to use is "btnAgain" for the first `<res>` element, and "retryTip" for the second `<res>` element.

```

<?xml version="1.0"?>
<file>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:idValueRule selector="//res" idValue="@name"/>
  </its:rules>
  <res name="retryBtn" xml:id="btnAgain">Try Again</res>
  <res name="retryTip">click this to re-run the process with the current
    settings.</res>
</file>

```

[Source file: <examples/xml/EX-idvalue-attribute-1.xml>]

8.16 Preserve Space



8.16.1 Definition



The [Preserve Space](#) data category indicates how whitespace should be handled in content. The possible values for this data category are "default" and "preserve" and carry the same meaning as the corresponding values of the [xml:space](#) attribute. The default value is "default".

8.16.2 Implementation



The [Preserve Space](#) data category can be expressed with global rules, or locally using the `xml:space` attribute. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements and attributes.

Note:

The [Preserve Space](#) data category is not applicable to HTML documents because `xml:space` (and by extension [Preserve Space](#)) has no effect in documents parsed as text/html.

GLOBAL: The `preserveSpaceRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `space` attribute with the value "default" or "preserve".

Example 74: The [Preserve Space](#) data category expressed globally

The `preserveSpaceRule` element specifies that whitespace in all verse elements must be treated literally.

```
<book>
  <info>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:preserveSpaceRule selector="//verse" space="preserve"/>
    </its:rules>
  </info>
  <verse>
    'Twas brillig, and the slithy toves
    Did gyre and gimble in the wabe;
    All mimsy were the borogoves,
    And the mome raths outgrabe.
  </verse>
</book>
```

[Source file: [examples/xml/EX-preserveSpace-global-1.xml](#)]

LOCAL: The `xml:space` attribute, as defined in section 2.10 of [\[XML 1.0\]](#), maps exactly to the [Preserve Space](#) data category.

Example 75: The [Preserve Space](#) data category expressed locally

The standard `xml:space` attribute specifies that the whitespace in the verse element must be treated literally.

```

<book>
  <verse xml:space="preserve">
'Twas brillig, and the slithy toves
  Did gyre and gimble in the wabe;
All mimsy were the borogoves,
  And the mome raths outgrabe.
  </verse>
</book>

[Source file: examples/xml/EX-preserve-space-local-1.xml]

```

8.17 Localization Quality Issue



8.17.1 Definition



The [Localization Quality Issue](#) data category is used to express information related to localization quality assessment tasks. Such tasks can be conducted on the translation of some source text into a target language or on the source text itself where its quality may impact on the localization process.

This data category can be used in a number of ways, including the following example scenarios:

- An automatic quality checking tool flags a number of potential quality issues in an XML or HTML file and marks them up using ITS 2.0 markup. Other tools in the workflow then examine this markup and decide whether the file needs to be reviewed manually or passed on for further processing without a manual review stage.
- A quality assessment process identifies a number of issues and adds the ITS markup to a rendered HTML preview of an XML file along with CSS styling that highlights these issues. The resulting HTML file is then sent back to the translator to assist his or her revision efforts.
- A human reviewer working with a web-based tool adds quality markup, including comments and suggestions, to a localized text as part of the review process. A subsequent process examines this markup to ensure that changes were made.

The data category defines five pieces of information:

Information	Description	Value	Notes
Type	A set of broad types of issues into which tool-specific issues can be categorized.	One of the values defined in list of type values .	ITS 2.0-compliant tools that use these types MUST map their internal values to these types. If the type of the issue is set to <code>uncategorized</code> , a comment MUST be specified as well.
Comment	A human-readable description of the quality issue.	Text	
Severity	A decimal value representing the severity of the issue, as defined by the	A rational number in the interval 0 to 100 (inclusive). The value follows the XML Schema decimal data type with the constraining facets minInclusive set to 0 and maxInclusive set to	It is up to tools to map the values of this to their own system to this scale. If needed, the original value can be passed along using a custom namespace for XML, or a <code>data-</code> attribute for HTML.

Information	Description	Value	Notes
	model generating the metadata.	100. The higher values represent greater severity.	
Profile Reference	A reference to a document describing the quality assessment model used for the issue.	An IRI pointing to the reference document.	The use of resolvable IRI is strongly recommended as it provides a way for human evaluators to learn more about the quality issues in use.
Enabled	A flag indicating whether the issue is enabled or not.	A value <code>yes</code> or <code>no</code> , with the default value being <code>yes</code> .	This flag is used to activate or deactivate issues. There is no prescribed behavior associated with activated or deactivated issues. One example of usage is a tool that allows the user to deactivate false positives so they are not displayed again each time the document is re-checked.

8.17.2 Implementation



The [Localization Quality Issue](#) data category can be expressed with global rules, or locally on individual elements. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements, but excluding attributes.

GLOBAL: The `locQualityIssueRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- Exactly one of the following:
 - Exactly one of the following:
 - A `locQualityIssuesRef` attribute. Its value is an IRI pointing to the `locQualityIssues` element containing the [list of issues](#) related to this content.
 - A `locQualityIssuesRefPointer` attribute that contains a [relative selector](#) pointing to a node with the exact same semantics as `locQualityIssuesRef`.
 - At least one of the following:
 - A `locQualityIssueType` attribute that implements the [type information](#).
 - A `locQualityIssueComment` attribute that implements the [comment information](#).
- An optional `locQualityIssueSeverity` attribute that implements the [severity information](#).
- An optional `locQualityIssueProfileRef` attribute that implements the [profile reference information](#).
- An optional `locQualityIssueEnabled` attribute that implements the [enabled information](#).

Note:

The attribute `locQualityIssuesRefPointer` does not apply to HTML as local markup is provided for direct annotation in HTML.

Example 76: Annotating an issue in XML with `locQualityIssueRule` element

The `locQualityIssueRule` element associates the issue information with the value of the `text` attribute.

```
<?xml version="1.0"?>
<doc>
  <header>
    <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
      <its:locQualityIssueRule selector="//image[@id='i1']/@text"
        locQualityIssueType="typographical"
        locQualityIssueComment="Sentence without capitalization"
        locQualityIssueSeverity="50"/>
    </its:rules>
  </header>
  <para>Click the button <image id="i1" src="button.png"
    text="start button"/>.</para>
</doc>
```

[Source file: <examples/xml/EX-locQualityIssue-global-1.xml>]

Example 77: Annotating an issue in XML with local standoff markup and a global rule

The following example shows a document using local standoff markup to encode several issues. But because, in this case, the `mrk` element does not allow attributes from another namespace we cannot use `locQualityIssuesRef` directly. Instead, a global rule is used to map the function of `locQualityIssuesRef` to a non-ITS construct, here the `ref` attribute of any `mrk` elements that has its attribute `type` set to "x-itslq".

```
<?xml version="1.0"?>
<doc xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <file>
    <header>
      <its:rules version="2.0">
        <its:locQualityIssueRule selector="//mrk[@type='x-itslq']"
          locQualityIssuesRefPointer="@ref"/>
      </its:rules>
    </header>
    <unit id="1">
      <segment>
        <source>This is the content</source>
        <target><mrk type="x-itslq" ref="#lq1">c'es</mrk> le contenu</target>
      </segment>
      <its:locQualityIssues xml:id="lq1">
        <its:locQualityIssue locQualityIssueType="misspelling"
          locQualityIssueComment="'c'es' is unknown. Could be 'c'est'"
          locQualityIssueSeverity="50"/>
        <its:locQualityIssue locQualityIssueType="typographical"
          locQualityIssueComment="Sentence without capitalization"
          locQualityIssueSeverity="30"/>
      </its:locQualityIssues>
    </unit>
  </file>
</doc>
```

[Source file: <examples/xml/EX-locQualityIssue-global-2.xml>]

LOCAL: Using the inline markup to represent the data category locally is limited to a single occurrence for a given content (e.g. one cannot have different `locQualityIssueType` attributes applied to the same span of text because the inner-most one would override the others). A local *standoff markup* is provided to allow such cases.

The following local markup is available for the [Localization Quality Issue](#) data category:

- Either (inline markup):
 - At least one of the following attributes:
 - A `locQualityIssueType` attribute that implements the [type information](#).
 - A `locQualityIssueComment` attribute that implements the [comment information](#).
 - An optional `locQualityIssueSeverity` attribute that implements the [severity information](#).
 - An optional `locQualityIssueProfileRef` attribute that implements the [profile reference information](#).
 - An optional `locQualityIssueEnabled` attribute that implements the [enabled information](#).
- Or (standoff markup):
 - A `locQualityIssuesRef` attribute. Its value is an IRI pointing to the `locQualityIssues` element containing the [list of issues](#) related to this content.
 - An element `locQualityIssues` with a `xml:id` attribute set to the identifier specified in the `locQualityIssuesRef` attribute. The `locQualityIssues` element contains:
 - One or more elements `locQualityIssue`, each of which contains:
 - At least one of the following attributes:
 - A `locQualityIssueType` attribute that implements the [type information](#).
 - A `locQualityIssueComment` attribute that implements the [comment information](#).
 - An optional `locQualityIssueSeverity` attribute that implements the [severity information](#).
 - An optional `locQualityIssueProfileRef` attribute that implements the [profile reference information](#).
 - An optional `locQualityIssueEnabled` attribute that implements the [enabled information](#).

When the attributes `locQualityIssueType`, `locQualityIssueComment`, `locQualityIssueSeverity`, `locQualityIssueProfileRef` and `locQualityIssueEnabled` are used in a standoff manner, the information they carry pertains to the content of the element that refers to the standoff annotation, not to the content of the element `locQualityIssue` where they are declared.

In HTML the standoff markup **MUST** be stored inside a `script` element. It **MUST** have a `type` attribute with the value `application/its+xml`. Its `id` attribute **MUST** be set to the same value as the `xml:id` attribute of the `locQualityIssues` element it contains.

Example 78: Annotating an issue in XML with local inline markup

The attributes `locQualityIssueType`, `locQualityIssueComment` and `locQualityIssueSeverity` are used to associate the issue information directly with a selected span of content.

```
<?xml version="1.0"?>
<doc xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <para><span its:locQualityIssueType="typographical"
    its:locQualityIssueComment="Sentence without capitalization"
    its:locQualityIssueSeverity="50">this</span> is an example</para>
</doc>
```

[Source file: <examples/xml/EX-locQualityIssue-local-1.xml>]

Example 79: Annotating an issue in HTML with local inline markup

In this example several spans of content are associated with a quality issue.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Telharmonium 1897</title>
    <style type=text/css>
      [its-loc-quality-issue-type]{
        background-color:yellow;
        margin:2px;
      }
      [its-loc-quality-issue-severity = "100"]{
        border: 2px solid red;
      }
    </style>
  </head>
  <body>
    <h1>Telharmonium (1897)</h1>
    <p>
      <span
        data-mytool-qacode=named_entity_not_found
        its-loc-quality-issue-comment="Should be Thomas Cahill."
        its-loc-quality-issue-profile-ref=http://example.org/qaMove1/v1
        its-loc-quality-issue-severity=100
        its-loc-quality-issue-type=inconsistent-entities>Christian Bale</span> (18
        from a power plant for hundreds of miles to listeners over telegraph wiring
        1889 the sound quality of regular telephone concerts was very poor on accou
        generated by carbon-granule microphones. As a result Cahill decided to set
        perfection of sound <span
          its-loc-quality-issue-comment="should be 'quality'"
          its-loc-quality-issue-profile-ref=grammar
          its-loc-quality-issue-severity=50
          its-loc-quality-issue-type=spelling>qulaity</span> with his instrument, a
        would overcome all the flaws of traditional instruments.</p>
    </body>
  </html>
```

[Source file: <examples/html5/EX-locQualityIssue-html5-local-1.html>]

Example 80: Annotating an issue in XML with local standoff markup

The following example shows a document using local standoff markup to encode several issues. The `mrk` element delimits the content to markup and holds a

`locQualityIssuesRef` attribute that points to the `locQualityIssues` element where the issues are listed.

```
<?xml version="1.0"?>
<xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2"
  xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <file original="example.doc" source-language="en" datatype="plaintext">
    <body>
      <trans-unit id="1">
        <source xml:lang="en">This is the content</source>
        <target xml:lang="fr"><mrk mtype="x-itslq"
          its:locQualityIssuesRef="#lq1">c'es</mrk> le contenu</target>
        <its:locQualityIssues xml:id="lq1">
          <its:locQualityIssue locQualityIssueType="misspelling"
            locQualityIssueComment="'c'es' is unknown. Could be 'c'est'"
            locQualityIssueSeverity="50"/>
          <its:locQualityIssue locQualityIssueType="typographical"
            locQualityIssueComment="Sentence without capitalization"
            locQualityIssueSeverity="30"/>
        </its:locQualityIssues>
      </trans-unit>
    </body>
  </file>
</xliff>
```

[Source file: <examples/xml/EX-locQualityIssue-local-2.xml>]

Example 81: Annotating an issue in HTML with local standoff markup

The following example shows a document using local standoff markup to encode several issues. The `span` element delimits the content to markup and holds a `loc-quality-issues-ref` attribute that points to a special `span` element where the issues are listed within a set of other special `span` elements.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=utf-8>
    <title>Test</title>
    <script src=qaissues.js type=text/javascript></script>
    <script type=application/its+xml id=lq1>
      <its:locQualityIssues xml:id="lq1" xmlns:its="http://www.w3.org/2005/11/its">
        <its:locQualityIssue
          locQualityIssueType="misspelling"
          locQualityIssueComment="'c'es' is unknown. Could be 'c'est'"
          locQualityIssueSeverity="50"/>
        <its:locQualityIssue
          locQualityIssueType="typographical"
          locQualityIssueComment="Sentence without capitalization"
          locQualityIssueSeverity="30"/>
      </its:locQualityIssues>
    </script>
    <style type=text/css>.qaissue { background-color: yellow; } </style>
  </head>
  <body onload=addqaissueattrs()>
    <p>
```

```
<span its-loc-quality-issues-ref=#lq1>c'es</span> le contenu</p>
</body>
</html>
```

[Source file: <examples/html5/EX-locQualityIssue-html5-local-2.html>]

8.18 Localization Quality Rating

8.18.1 Definition

The [Localization Quality Rating](#) data category is used to express an overall measurement of the localization quality of a document or an item in a document.

This data category allows to specify a quality score or a voting result for a given item or document, as well as to indicate what constitutes a passing score or vote. It also allows to point to a profile describing the quality assessment model used for the scoring or the voting.

8.18.2 Implementation

The [Localization Quality Rating](#) data category is only expressed locally on individual elements. The data category information [inherits](#) to the textual content of the element, *including* child elements, but *excluding* attributes.

LOCAL: The following local markup is available for the [Localization Quality Rating](#) data category:

- Exactly one of the following:
 - A `locQualityRatingScore` attribute. Its value is a rational number in the interval 0 to 100 (inclusive). The value follows the [XML Schema decimal data type](#) with the constraining facets [minInclusive](#) set to 0 and [maxInclusive](#) set to 100. The higher values represent better quality.
 - A `locQualityRatingVote` attribute. Its value is a signed integer with higher values indicating a better vote.
- If `locQualityRatingScore` is used:
 - an optional `locQualityRatingScoreThreshold` attribute indicating the lowest score that constitutes a passing score in the profile used. Its value is a rational number in the interval 0 to 100 (inclusive). The value follows the [XML Schema decimal data type](#) with the constraining facets [minInclusive](#) set to 0 and [maxInclusive](#) set to 100.
- If `locQualityRatingVote` is used:
 - an optional `locQualityRatingVoteThreshold` attribute indicating the lowest value that constitutes a passing vote in the profile used. Its value is a signed integer.
- An optional `locQualityRatingProfileRef` attribute. Its value is an IRI pointing to the reference document describing the quality assessment model used for the scoring.

Example 82: The [Localization Quality Rating](#) data category expressed locally in XML

The `locQualityRatingScore`, `locQualityRatingThreshold` and `locQualityRatingProfileRef` are used to score the quality of the document.

```
<?xml version="1.0"?>
<doc xml:lang='nl'
  xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0"
```

```

its:locQualityRatingScore="100"
its:locQualityRatingScoreThreshold="95"
its:locQualityRatingProfileRef="http://example.org/qaModel/v13">
<title>De lotgevallen van Tom Sawyer</title>
<para>Hij kwam vrij laat te huis, en toen hij voorzichtig het raam insprong,
viel hij in eene hinderlaag, in de persoon van zijne tante, bij wie, toen zij
den staat zag, waarin zijne kleederen verkeerden, het besluit om zijn vrijen
Zaterdag in een gevangenschap met dwangarbeid te veranderen, onherroepelijk
vaststond.</para>
</doc>

```

[Source file: <examples/xml/EX-locQualityRating-local-1.xml>]

Example 83: The Localization Quality Rating data category expressed locally in HTML

The `its-loc-quality-rating-score`, `its-loc-quality-rating-score-threshold` and `its-loc-quality-rating-profile-ref` are used to score the quality of the document.

```

<html lang=fr
  its-loc-quality-rating-profile-ref=http://example.org/qaModel/v13
  its-loc-quality-rating-score=90
  its-loc-quality-rating-score-threshold=80
>
<head>
  <meta charset=utf-8>
  <title>Rikki-tikki-tavi</title>
</head>
<body>
  <p>C'est l'histoire de la grande guerre que Rikki-Tikki-Tavi a combattu tout se
à travers les salles de bain du grand bungalow au cantonnement Segowlee. Darze
le tailbird, l'a aidé, et Chuchundra, le rat musqué, qui ne sort jamais jusqu'
milieu du plancher, mais se glisse toujours contre la paroi, lui donnait des
conseils, mais Rikki-Tikki-Tavi fait le véritable combat.</p>
</body>
</html>

```

[Source file: <examples/html5/EX-locQualityRating-html5-local.html>]

8.19 MT Confidence

8.19.1 Definition

The MT Confidence data category is used to communicate the self-reported confidence score from a machine translation engine of the accuracy of a translation it has provided. It is not intended to provide a score that is comparable between machine translation engines and platforms. This data category does NOT aim to establish any sort of correlation between the self-reported confidence score and either human evaluation of MT usefulness, or post-editing cognitive effort. For harmonization's sake, MT Confidence is provided as a rational number in the interval 0 to 1 (inclusive).

Note:

Implementers are expected to interpret the floating point number and present it to human and other consumers in a convenient form, such as percentage (0-100%) with up to 2 decimal digits, font or background color coding, etc.

This data category can be used for several purposes, including, but not limited to:

- Automated prioritising of raw machine translated text for further processing based on empirically set thresholds.
- Providing readers, translators, post-editors, reviewers and proof-readers of machine translated text with self-reported relative accuracy prediction.

MT confidence scores can be displayed e.g. on websites machine translated on the fly, by simple web-based translation editors or on Computer Aided Translation (CAT) tools.

8.19.2 Implementation



The [MT Confidence](#) category can be expressed with global rules or locally on individual elements. For elements, the data category information [is inherited](#) by the textual content of the element, *including* child elements, but *excluding* attributes.

Any node selected by the [MT Confidence](#) data category [MUST](#) be contained in an element with the [annotatorsRef](#) (or in HTML, [its-annotators-ref](#)) attribute specified for the [MT Confidence](#) data category. For more information, see [Section 5.8: ITS Tools Annotation](#).

GLOBAL: The `mtConfidenceRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- A required `mtConfidence` attribute with a value that represents the translation confidence score as a rational number in the interval 0 to 1 (inclusive). The value follows the [XML Schema decimal data type](#) with the constraining facets [minInclusive](#) set to 0 and [maxInclusive](#) set to 1.

Example 84: Global usage of `mtConfidenceRule` in a HTML document to specify the confidence scores for the translation into English of the `title` attributes of two `img` elements.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <link href=EX-mtconfidence-global-html5-1-external-rules.xml rel=its-rules>
    <title>Machine translated title attributes of img elements give MT
      confidence scores using global rules</title>
  </head>
  <body its-annotators-ref="mt-confidence|file:///tools.xml#T1">
    <p>
      
      
    </p>
  </body>
</html>
```

[Source file: [examples/html5/EX-mtConfidence-global-html5-1.html](#)]

Where the external ITS rules file is as shown:

Example 85: XML file with external rules references from an HTML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0"
  xmlns:h="http://www.w3.org/1999/xhtml">
  <its:mtConfidenceRule mtConfidence="0.785" selector="//h:img[@title='Front ga
  <its:mtConfidenceRule mtConfidence="0.805" selector="//h:img[@title='A tart w
</its:rules>
```

[Source file: <examples/html5/EX-mtconfidence-global-html5-1-external-rules.xml>]

LOCAL: the following local markup is available for the [MT Confidence](#) data category:

- A `mtConfidence` attribute with a value that represents the translation confidence score as a rational number in the interval 0 to 1 (inclusive). The value follows the [XML Schema decimal data type](#) with the constraining facets [minInclusive](#) set to 0 and [maxInclusive](#) set to 1.

Example 86: The [MT Confidence](#) data category expressed locally for the content of a span in an XML document.

```
<text xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0" its:annotatorsR
  <body>
    <p>
      <span its:mtConfidence="0.8982">Dublin is the capital city of
        Ireland.</span>
    </p>
  </body>
</text>
```

[Source file: <examples/xml/EX-mtConfidence-local-1.xml>]

Example 87: The [MT Confidence](#) data category expressed locally for the content of two separate spans in a HTML document.

```
<!DOCTYPE html>
<html lang=en >
  <head>
    <meta charset=utf-8>
    <title>Sentences about Dublin and Prague
      machine translated from Czech with mtConfidence locally.</title>
  </head>
  <body its-annotators-ref="mt-confidence|file:///tools.xml#T1">
    <p>
      <span its-mt-confidence=0.8982>Dublin is the capital of Ireland.</span>
      <span its-mt-confidence=0.8536 >The capital of the Czech Republic is Pra
    </p>
  </body>
</html>
```

[Source file: <examples/html5/EX-mtConfidence-html5-local-1.html>]

8.20 Allowed Characters



8.20.1 Definition



The [Allowed Characters](#) data category is used to specify what characters are allowed in a given piece of content.

This data category can be used for various purposes, including the following examples:

- Limiting the characters that may be used in the UI of a game due to font restrictions.
- Preventing illegal characters from being entered as text content that represents file or directory names.
- Controlling what characters can be used when translating examples of a login name in content.

Note:

The [Allowed Characters](#) data category is not intended to disallow HTML markup. The purpose is to restrict the content to various characters only, e.g., when the content is to be used for URL or filename generation. In most Content Management Systems, content is divided into several fields, some of which may be restricted to plain text, while in other fields HTML fragments may be allowed. Enforcing such restrictions is outside the scope of this data category. For further information see [Section 1.3.1.4: Content producers](#).

The set of characters that are allowed is specified using a regular expression. That is, each character in the selected content [MUST](#) be included in the set specified by the regular expression.

The regular expression is a character class construct as defined in the section [Character Classes](#) of XML Schema [\[XML Schema Part 2\]](#), with the assumption that the `.` metacharacter matches also CARRIAGE RETURN (U+000D) and LINE FEED (U+000F). That is with the *dot-all* option set.

Example of expressions (shown as XML source):

- `"[abc]"` : allows the characters 'a', 'b' and 'c'.
- `"[a-c]"` : allows the characters 'a', 'b' and 'c'.
- `"[a-zA-Z]"` : allows the characters from 'a' to 'z' and from 'A' to 'Z'.
- `"[^abc]"` : allows any characters except 'a', 'b', and 'c'.
- `"[^a-c]"` : allows any characters except 'a', 'b', and 'c'.
- `"\w"` : allows any character except the set of "punctuation", "separator" and "other" characters.
- `"[--[\<:>:"\\\/\|?*"]"` : allows only the characters valid for Windows file names.
- `"."` : allows any character.
- `""` : allows no character.
- `"[a-ÿ-[\s]"` : allows all characters between U+0061 and U+00FF except the characters SPACE (U+0020), TABULATION (U+0009), CARRIAGE RETURN (U+000D) and LINE FEED (U+000F).

8.20.2 Implementation



The [Allowed Characters](#) data category can be expressed with global rules, or locally on individual elements. For elements, the data category information [inherits](#) to the textual content of the element, *including* child elements, but *excluding* attributes.

GLOBAL: The `allowedCharactersRule` element contains the following:

- A required `selector` attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- Exactly one of the following:
 - An `allowedCharacters` attribute that contains the regular expression indicating the allowed characters.
 - An `allowedCharactersPointer` attribute that contains a [relative selector](#) pointing to a node with the exact same semantics as `allowedCharacters`.

Example 88: The [Allowed Characters](#) data category expressed globally in XML

The `allowedCharactersRule` element states that the translated content of elements content must not contain the characters `*` and `+`.

```
<?xml version="1.0"?>
<myRes xmlns:its="http://www.w3.org/2005/11/its">
  <head>
    <its:rules version="2.0">
      <its:allowedCharactersRule allowedCharacters="^[^*+]" selector="//content"/>
    </its:rules>
  </head>
  <body>
    <content>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
      nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed
      diam voluptua.</content>
  </body>
</myRes>
```

[Source file: [examples/xml/EX-allowedCharacters-global-1.xml](#)]

Example 89: Mapping the [Allowed Characters](#) data category in XML

The attribute `allowedCharactersPointer` is used to map the data category to the non-ITS attribute `set` in this document. The attribute has the same semantics as `allowedCharacters`.

```
<?xml version="1.0"?>
<res xmlns:its="http://www.w3.org/2005/11/its">
  <head>
    <its:rules version="2.0">
      <its:allowedCharactersRule selector="//record" allowedCharactersPointer="@set">
    </its:rules>
  </head>
  <record id="a1" set="[ &#xFF01;-&#xFF5E;]">FULL WIDTH ONLY</record>
</res>
```

[Source file: [examples/xml/EX-allowedCharacters-global-2.xml](#)]

LOCAL: the following local markup is available for the [Allowed Characters](#) data category:

- A `allowedCharacters` attribute that contains the regular expression indicating the allowed characters.

Example 90: The [Allowed Characters](#) data category expressed locally in XML

The local `allowedCharacters` attribute specifies that the translated content of element `panelmsg` must contain only Unicode characters between U+0020 and U+00FE.

```
<?xml version="1.0"?>
<messages xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <msg num="123">Click the <panelmsg its:allowedCharacters=" [&#x0020;-&#x00FE;] ">
    >CONTINUE</panelmsg> Button on the printer panel</msg>
</messages>
```

[Source file: [examples/xml/EX-allowedCharacters-local-1.xml](#)]

Example 91: The [Allowed Characters](#) data category expressed locally in HTML

The local `its-allowed-characters` attribute specifies that the translated content of element `code` must not contain the characters other than 'a' to 'z' in any case and the characters underscore and minus.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Example</title>
  </head>
  <body>
    <p>Login names can only use letters from A to Z (upper or lowercase)
    and the character underscore (_) and minus (-).
    For example: <code its-allowed-characters=[a-zA-Z_\-]>Huck_Finn</code>.</p>
  </body>
</html>
```

[Source file: [examples/html5/EX-allowedCharacters-html5-local-1.html](#)]

8.21 Storage Size

8.21.1 Definition

The [Storage Size](#) data category is used to specify the maximum storage size of a given content.

This data category can be used for various purposes, including the following examples:

- Verify during translation if a string fits into a fixed-size database field.
- Control the size of a string that is stored in a fixed-size memory buffer at run-time.

The storage size is expressed in bytes and is provided along with the character set encoding used to store the content.

8.21.2 Implementation

The [Storage Size](#) data category can be expressed with global rules, or locally on individual elements. There is no inheritance. The default value of the character set encoding is UTF-8.

GLOBAL: The `storageSizeRule` element contains the following:

- A required **selector** attribute. It contains an [absolute selector](#) which selects the nodes to which this rule applies.
- Exactly one of the following:
 - A **storageSize** attribute. It contains the maximum number of bytes the text of the selected node is allowed in storage.
 - A **storageSizePointer** attribute that contains a [relative selector](#) pointing to a node with the exact same semantics as **storageSize**.
- None or exactly one of the following:
 - A **storageEncoding** attribute. It contains the name of the character set encoding used to calculate the number of bytes of the selected text. The name **MUST** be one of the names or aliases listed in the [IANA Character Sets registry \[IANA Character Sets\]](#). The default value is "UTF-8".
 - A **storageEncodingPointer** attribute that contains a [relative selector](#) pointing to a node with the exact same semantics as **storageEncoding**.
- An optional **lineBreakType** attribute. It indicates what type of line breaks the storage uses. The possible values are: **cr** for CARRIAGE RETURN (U+000D), **lf** for LINE FEED (U+000A), **crLf** for CARRIAGE RETURN (U+000D) followed by LINE FEED (U+000A), or **nel** for NEXT LINE (U+0085). The default value is **lf**.

Example 92: The [Storage Size](#) data category expressed globally in XML

The **storageSizeRule** element is used to specify that, when encoded in ISO-8859-1, the content of the **country** element must not be more than 25 bytes. The name "Papouasie-Nouvelle-Guinée" is 25 character long and fits because all characters in ISO-8859-1 are encoded as a single byte.

```
<?xml version="1.0"?>
<db>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:storageSizeRule selector="//country" storageSize="25"
      storageEncoding="ISO-8859-1"/>
  </its:rules>
  <data>
    <country id="123">Papouasie-Nouvelle-Guinée</country>
    <country id="139">République Dominicaine</country>
  </data>
</db>
```

[Source file: [examples/xml/EX-storageSize-global-1.xml](#)]

Example 93: Mapping the [Storage Size](#) data category in XML

The **storageSizePointer** attribute is used to map the non-ITS attribute **max** to the same functionality as **storageSize**. There is no character set encoding specified, so the default UTF-8 is assumed. Note that, while the name "Papouasie-Nouvelle-Guinée" is 25 character long, the character 'é' is encoded into two bytes in UTF-8. Therefore this name is one byte too long to fit in its storage destination.

```
<?xml version="1.0"?>
<fields>
  <its:rules xmlns:its="http://www.w3.org/2005/11/its" version="2.0">
    <its:storageSizeRule selector="//field" storageSizePointer="@max"/>
  </its:rules>
  <field type="country" id="123" max="25">Papouasie-Nouvelle-Guinée</field>
  <field type="country" id="139" max="25">République Dominicaine</field>
</fields>
```

[Source file: <examples/xml/EX-storageSize-global-2.xml>]

LOCAL: the following local markup is available for the [Storage Size](#) data category:

- A `storageSize` attribute. It contains the maximum number of bytes the text of the selected node is allowed in storage.
- An optional `storageEncoding` attribute. It contains the name of the character set encoding used to calculate the number of bytes of the selected text. The name **MUST** be one of the names or aliases listed in the [IANA Character Sets registry](#) [IANA Character Sets]. The default value is "UTF-8".
- An optional `lineBreakType` attribute. It indicates what type of line breaks the storage uses. The possible values are: `cr` for CARRIAGE RETURN (U+000D), `lf` for LINE FEED (U+000A), `crlf` for CARRIAGE RETURN (U+000D) followed by LINE FEED (U+000A), or `nel` for NEXT LINE (U+0085). The default value is `lf`.

Example 94: The [Storage Size](#) data category expressed locally in XML

The `storageSize` attribute allows to specify different the maximum storage sizes throughout the document.

```
<?xml version="1.0"?>
<messages xmlns:its="http://www.w3.org/2005/11/its" its:version="2.0">
  <var num="panelA1_Continue" its:storageSize="8" its:storageEncoding="UTF-16">CO
  <var num="panelA1_Stop" its:storageSize="8" its:storageEncoding="UTF-16">STOP</
  <var num="panelB5_Cancel" its:storageSize="12" its:storageEncoding="UTF-16">CAN
</messages>
```

[Source file: <examples/xml/EX-storageSize-local-1.xml>]

Example 95: The [Storage Size](#) data category expressed locally in HTML

The `its-storage-size` is used here to specify the maximum number of bytes the two editable strings can have in UTF-8.

```
<!DOCTYPE html>
<html lang=en>
  <head>
    <meta charset=utf-8>
    <title>Example</title>
  </head>
  <body>
    <p>String to translate:</p>
    <p contenteditable=true id=123 its-storage-size=25>Papua New-Guinea</p>
    <p contenteditable=true id=139 its-storage-size=25>Dominican Republic</p>
  </body>
</html>
```

[Source file: <examples/html5/EX-storageSize-html5-local-1.html>]

A References



BCP47

Addison Phillips, Mark Davis. [Tags for Identifying Languages](#), September 2009. Available at <http://www.rfc-editor.org/rfc/bcp/bcp47.txt>.

IANA Character Sets

[Character Sets](http://www.iana.org/assignments/character-sets) Available at <http://www.iana.org/assignments/character-sets>.

QAFRAMEWORK

Karl Dubost, Lynne Rosental, Dominique Hazaël-Massieux, Lofton Henderson. [QA Framework: Specification Guidelines](http://www.w3.org/TR/2005/REC-qaframe-spec-20050817/). W3C Recommendation 17 August 2005. Available at <http://www.w3.org/TR/2005/REC-qaframe-spec-20050817/>. The latest version of [QAFRAMEWORK](http://www.w3.org/TR/qaframe-spec/) is available at <http://www.w3.org/TR/qaframe-spec/>.

RELAX NG

Information technology -- Document Schema Definition Language (DSDL) -- Part 2: *Regular-grammar-based validation* -- RELAX NG. International Organization for Standardization (ISO) ISO/IEC 19757-2:2003.

RFC 2119

S. Bradner. [Key Words for use in RFCs to Indicate Requirement Levels](http://www.ietf.org/rfc/rfc2119.txt). IETF RFC 2119, March 1997. Available at <http://www.ietf.org/rfc/rfc2119.txt>.

RFC 3987

Martin Dürst, Michel Suignard. [Internationalized Resource Identifiers \(IRIs\)](http://www.ietf.org/rfc/rfc3987.txt). RFC 3987, January 2005. See <http://www.ietf.org/rfc/rfc3987.txt>.

XLink 1.1

Steve DeRose, Eve Maler, David Orchard, Norman Walsh. [XML Linking Language 1.1](http://www.w3.org/TR/2010/REC-xlink11-20100506/). W3C Recommendation 6 May 2010. Available at <http://www.w3.org/TR/2010/REC-xlink11-20100506/>. The latest version of [XLink 1.1](http://www.w3.org/TR/xlink11/) is available at <http://www.w3.org/TR/xlink11/>.

XML 1.0

Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, et al., editors. [Extensible Markup Language \(XML\) 1.0 \(Fourth Edition\)](http://www.w3.org/TR/2006/REC-xml-20060816/), W3C Recommendation 16 August 2006. Available at <http://www.w3.org/TR/2006/REC-xml-20060816/>. The latest version of [XML 1.0](http://www.w3.org/TR/REC-xml/) is available at <http://www.w3.org/TR/REC-xml/>.

XML Infoset

John Cowan, Richard Tobin. [XML Information Set \(Second Edition\)](http://www.w3.org/TR/2004/REC-xml-infoset-20040204/). W3C Recommendation 4 February 2004. Available at <http://www.w3.org/TR/2004/REC-xml-infoset-20040204/>. The latest version of [XML Infoset](http://www.w3.org/TR/xml-infoset/) is available at <http://www.w3.org/TR/xml-infoset/>.

XML Names

Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin. [Namespaces in XML \(Second Edition\)](http://www.w3.org/TR/2006/REC-xml-names-20060816/). W3C Recommendation 16 August 2006. Available at <http://www.w3.org/TR/2006/REC-xml-names-20060816/>. The latest version of [XML Names](http://www.w3.org/TR/REC-xml-names/) is available at <http://www.w3.org/TR/REC-xml-names/>.

XML Schema

Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn. [XML Schema Part 1: Structures Second Edition](http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/). W3C Recommendation 28 October 2004. Available at <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>. The latest version of [XML Schema](http://www.w3.org/TR/xmlschema-1/) is available at <http://www.w3.org/TR/xmlschema-1/>.

XML Schema Part 2

Paul V. Biron, Ashok Malhotra. [XML Schema Part 2: Datatypes Second Edition](http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/). W3C Recommendation 28 October 2004. Available at <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>. The latest version of [XML Schema](http://www.w3.org/TR/xmlschema-2/) is available at <http://www.w3.org/TR/xmlschema-2/>.

XPath 1.0

James Clark. [XML Path Language \(XPath\) Version 1.0](http://www.w3.org/TR/1999/REC-xpath-19991116/). W3C Recommendation 16 November 1999. Available at <http://www.w3.org/TR/1999/REC-xpath-19991116/>. The latest version of [XPath 1.0](http://www.w3.org/TR/xpath/) is available at <http://www.w3.org/TR/xpath/>.

B Internationalization Tag Set (ITS) MIME Type



This section defines a MIME type for Internationalization Tag Set (ITS) documents. It covers both ITS 1.0 and ITS 2.0.

Type name: application

Subtype name: its+xml

Required parameters: none

Optional parameters: charset

This parameter has identical semantics to the charset parameter of the "application/xml" media type as specified in IETF RFC 3023.

Encoding considerations: Identical to those of "application/xml" as described in IETF RFC 3023, section 3.2, as applied to an ITS document.

Security considerations: An ITS 1.0 or ITS 2.0 document may cause arbitrary URIs or IRIs to be dereferenced, via the @xlink:href attribute at the its:rules element. Therefore, the security issues of [RFC3987] Section 8 should be considered. In addition, the contents of resources identified by file: URIs can in some cases be accessed, processed and returned as results. An implementation of ITS global rules requires the support of XPath 1.0 or its successor. Hence, processing of global rules might encompass dereferencing of URIs or IRIs during computation of XPath expressions. Arbitrary recursion is possible, as is arbitrarily large memory usage, and implementations may place limits on CPU and memory usage, as well as restricting access to system-defined functions. ITS 1.0 and ITS 2.0 permit extensions. Hence it is possible that application/its+xml may describe content that has security implications beyond those described here.

Interoperability considerations: There are no known interoperability issues.

Published specification: <http://www.w3.org/TR/2007/REC-its-20070403/> and <http://www.w3.org/TR/its20/>.

Any XML document containing ITS 1.0 "its:rules" elements <http://www.w3.org/TR/its/#selection-global> can be labeled with application/its+xml. <http://www.w3.org/TR/its/EX-link-external-rules-2.xml> Provides an example of a document linking to a file with ITS 1.0 and ITS 2.0 "rules". The link target is at <http://www.w3.org/TR/its/EX-link-external-rules-1.xml>. There is no need that the link target has "its:rules" as a root element. The processing semantics is that rules are gathered in document order.

Applications that use this media type: This new media type is being registered to allow for deployment of ITS 1.0 and ITS 2.0 on the World Wide Web., e.g. by localization tools.

Additional information:

- Magic number(s): none
- File extension(s): its
- Macintosh file type code(s): TEXT

Person & email address to contact for further information: World Wide Web Consortium <web-human@w3.org>

Intended usage: COMMON

Restrictions on usage: none

Author / Change controller: The Internationalization Tag Set (ITS) 1.0 and 2.0 specifications are a work product of the World Wide Web Consortium's Internationalization Tag Set Working Group. The W3C has change control over this specification.

C Values for the Localization Quality Issue Type



The `locQualityIssueType` attribute provides a basic level of interoperability between different localization quality assurance systems. It offers a list of high-level quality issue types common in automatic and human localization quality assessment. Tools can map their internal types to these

types in order to exchange information about the kinds of issues they identify and take appropriate action even if another tool does not know the specific issues identified by the generating tool.

The values listed in the following table are allowed for `locQualityIssueType`. The values a tool implementing the data category produces for the attribute `MUST` match one of the values provided in this table and `MUST` be semantically accurate. If a tool can map its internal values to these types it `MUST` do so and `MUST NOT` use the value `other`, which is reserved strictly for values that cannot be mapped to these values.

Note:

The [ITS Interest Group](#) maintains an informative mappings of tools to localization quality issue types. [The ITS IG Wiki](#) provides information on [how to update that list](#).

Value	Description	Example	Scope	Notes
terminology	An incorrect term or a term from the wrong domain was used or terms are used inconsistently.	<ul style="list-style-type: none"> The localization had “Pen Drive” when corporate terminology specified that “USB Stick” was to be used. The localization text inconsistently used “Start” and “Begin”. A text renders the Hungarian term <i>recsegőék</i> as “buzzer bridge” in English to translate (a literal translation), but the term used in English should be “wedge block,” as specified in a terminology list supplied to the translator. 	S or T	This value <code>MUST NOT</code> be used for simple typographical errors or word choice not related to defined terminologies. For example, a mistyping of “pin” as “pen” or the use of “imply” instead of “infer” (mistaking two commonly confused words) would not count as terminology issues and should be categorized as either spelling errors or mistranslations, depending on the nature of the issue. Terminology refers <i>only</i> to cases where incorrect choices about terms (either formal or commonly defined in a domain) are involved.
mistranslation	The content of the target mistranslates the content of the source.	<ul style="list-style-type: none"> The English source reads “An ape succeeded in grasping a banana lying 	T	Issues related to translation of specific terms related to the domain or task-specific language should be categorized as <code>terminology</code> issues.

Value	Description	Example	Scope	Notes
		outside its cage with the help of a stick" but the Italian translation reads "l'ape riuscì a prendere la banana posta fuori dalla sua gabbia aiutandosi con un bastone" ("A bee succeeded...")		
omission	Necessary text has been omitted from the localization or source.	<ul style="list-style-type: none"> One or more segments found in the source that should have been translated are missing in the target. 	S or T	This type should not be used for missing whitespace or formatting codes, but instead should be reserved for linguistic content.
untranslated	Content that should have been translated was left untranslated.	<ul style="list-style-type: none"> The source segment reads "The Professor said to Smith that he would hear from his lawyer" but the Hungarian localization reads "A professzor azt mondta Smithnek, hogy he would hear from his lawyer." 	T	<code>omission</code> takes precedence over <code>untranslated</code> . Omissions are distinct in that they address cases where text is not present, while <code>untranslated</code> addresses cases where text has been carried from the source untranslated.
addition	The translated text contains inappropriate additions.	<ul style="list-style-type: none"> The translated text contains a note from the translator to himself to look up a term; the note should have been deleted but was not. 	T	
duplication	Content has been duplicated improperly.	<ul style="list-style-type: none"> A section of the target text was 	T	

Value	Description	Example	Scope	Notes
		inadvertently copied twice in a copy and paste operation.		
inconsistency	The text is inconsistent with itself (NB: not for use with terminology inconsistency).	<ul style="list-style-type: none"> The text states that an event happened in 1912 in one location but in another states that it happened in 1812. 	S or T	
grammar	The text contains a grammatical error (including errors of syntax and morphology).	<ul style="list-style-type: none"> The text reads "The guidelines says that users should use a static grounding strap." 	S or T	
legal	The text is legally problematic (e.g., it is specific to the wrong legal system).	<ul style="list-style-type: none"> The localized text is intended for use in Thailand but includes U.S. regulatory notices. A text translated into German contains comparative advertising claims that are not allowed by German law. 	S or T	
register	The text is written in the wrong linguistic register of uses slang or other language variants inappropriate to the text.	<ul style="list-style-type: none"> A financial text in U.S. English refers to dollars as "bucks". 	S or T	
locale-specific-content	The localization contains content that does not apply to the locale for which it was prepared.	<ul style="list-style-type: none"> A text translated for the Japanese market contains call center 	S or T	Legally inappropriate material should be classified as <code>legal</code> .

Value	Description	Example	Scope	Notes
		<p>numbers in Texas and refers to special offers available only in the U.S.</p>		
locale-violation	Text violates norms for the intended locale.	<ul style="list-style-type: none"> A text localized into German has dates in YYYY-MM-DD format instead of in DD.MM.YYYY. A text for the Irish market uses American-style foot and inch measurements instead of centimeters. A text intended for a U.S.-based audience uses U.K. spellings such as “centre” and “colour.” 	S or T	<p>This category should be used for spelling errors only if they relate specifically to locale expectations (e.g., a text consistently uses British instead of U.S. spellings for a text intended for the U.S.). If these errors are not systematic (e.g., a text uses U.S. spellings but has a single instance of “centre”), they should instead be counted as spelling errors.</p>
style	The text contains stylistic errors.	<ul style="list-style-type: none"> Company style guidelines dictates that all individuals be referred to as Mr. or Ms. with a family name, but the text refers to “Jack Smith”. 	S or T	
characters	The text contains characters that are garbled or incorrect or that are not used in the language in which the content appears.	<ul style="list-style-type: none"> A text should have a '•' but instead has a '¥' sign. A text translated into German omits the umlauts over 'ü', 'ö', and 'ä'. 	S or T	
misspelling	The text contains a misspelling.	<ul style="list-style-type: none"> A German text misspells the word 	S or T	

Value	Description	Example	Scope	Notes
		"Zustellung" as "Zustellüng".		
typographical	The text has typographical errors such as omitted/incorrect punctuation, incorrect capitalization, etc.	<ul style="list-style-type: none"> An English text has the following sentence: "The man whom, we saw, was in the Military and carried it's insignias". 	S or T	
formatting	The text is formatted incorrectly.	<ul style="list-style-type: none"> Warnings in the text are supposed to be set in italic face, but instead appear in bold face. Margins of the text are narrower than specified. 	S or T	
inconsistent-entities	The source and target text contain different named entities (dates, times, place names, individual names, etc.)	<ul style="list-style-type: none"> The name "Thaddeus Cahill" appears in an English source but is rendered as "Tamaš Cahill" in the Czech version. The date "February 9, 2007" appears in the source but the translated text has "2. September 2007". 	S or T	
numbers	Numbers are inconsistent between source and target.	<ul style="list-style-type: none"> A source text states that an object is 120 cm long, but the target text says it is 129 cm. long. 	S or T	Some tools may correct for differences in units of measurement to reduce false positives.
markup	There is an issue related to markup	<ul style="list-style-type: none"> The source segment has 	S or T	

Value	Description	Example	Scope	Notes
	or a mismatch in markup between source and target.	<p>five markup tags but the target has only two.</p> <ul style="list-style-type: none"> An opening tag in the text is missing a closing tag. 		
pattern-problem	The text fails to match a pattern that defines allowable content (or matches one that defines non-allowable content).	<ul style="list-style-type: none"> The tool disallows the regular expression pattern <code>["'"][\.,]</code> but the translated text contains "A leading "expert", a political hack, claimed otherwise." 	S or T	
whitespace	There is a mismatch in whitespace between source and target content or the text violates specific rules related to the use of whitespace..	<ul style="list-style-type: none"> A source segment starts with six space characters but the corresponding target segment has two non-breaking spaces at the start. The text uses a run of 12 space characters instead of a tab character to align numbers in a table. Two space characters appear after a period even though only a single period should be used. 	S or T	
internationalization	There is an issue related to the internationalization of content.	<ul style="list-style-type: none"> A line of programming code has embedded 	S or T	There are many kinds of internationalization issues. This category is therefore very

Value	Description	Example	Scope	Notes
		language-specific strings. <ul style="list-style-type: none"> A user interface element leaves no room for text expansion. A form allows only for U.S.-style postal addresses and expects five digit U.S. ZIP codes. 		heterogeneous in what it can refer to.
length	There is a significant difference in source and target length.	<ul style="list-style-type: none"> The translation of a segment is five times as long as the source. 	T or S	What constitutes a "significant" difference in length is determined by the model referred to in the locQualityIssueProfileRef
uncategorized	The issue has not been categorized.	<ul style="list-style-type: none"> A new version of a tool returns information on an issue that has not been previously checked and that is not yet classified. 	S or T	This category has two uses: <ol style="list-style-type: none"> A tool can use it to pass through quality data from another tool in cases where the issues from the other tool are not classified (for example, a localization quality assurance tool interfaces with a third-party grammar checker). A tool's issues are not yet assigned to categories, and, until an updated assignment is made, they may be listed as <code>uncategorized</code>. In this case it is recommended that issues be assigned to appropriate categories as soon as possible since <code>uncategorized</code> does not foster interoperability.
other	Any issue that cannot be assigned to any values listed above.		S or T	<ul style="list-style-type: none"> This category allows for the inclusion of any issues not included in the previously listed values. This value MUST NOT be used for any tool- or model-specific issues

Value	Description	Example	Scope	Notes
				<p>that can be mapped to the values listed above.</p> <ul style="list-style-type: none"> In addition, this value is not synonymous with uncategorized in that uncategorized issues may be assigned to another precise value, while other issues cannot. If a system has an "miscellaneous" or "other" category, it MUST be mapped to this value even if the specific instance of the issue might be mapped to another category.

D Schemas for ITS



The following schemas define ITS elements and attributes and can be used as building blocks when you want to integrate ITS markup into your own XML vocabulary. You can see examples of such integration in [Best Practices for XML Internationalization](#).

The following four schemas are provided:

1. *NVDL document*: The following [NVDL](#) document allows validation of ITS markup which has been added to a host vocabulary. Only ITS elements and attributes are checked. Elements and attributes of host language are ignored during validation against this NVDL document/schema.

Example 96: NVDL schema for ITS

```
<?xml version="1.0" encoding="UTF-8"?>
<rules xmlns="http://purl.oclc.org/dsdl/nvdl/ns/structure/1.0">
  <namespace ns="http://www.w3.org/2005/11/its">
    <validate schema="its20-elements.rng"/>
  </namespace>
  <namespace ns="http://www.w3.org/2005/11/its" match="attributes">
    <validate schema="its20-attributes.rng"/>
  </namespace>
  <anyNamespace>
    <allow/>
  </anyNamespace>
</rules>
```

[Source file: [schemas/its20.nvdl](#)]

2. *RELAX NG schema for elements and attributes*: The NVDL schema depends on the following two schemas: RELAX NG schema for ITS elements, and RELAX NG schema for all ITS local attributes.

Example 97: RELAX NG schema for ITS elements

```

<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0" xmlns="http://relaxng.org/ns/structure/1.0">
  <include href="its20.rng"/>
  <define name="its-local.attributes" combine="interleave">
    <a:documentation>enable all xml:* attributes</a:documentation>
    <zeroOrMore>
      <attribute>
        <nsName ns="http://www.w3.org/XML/1998/namespace"/>
      </attribute>
    </zeroOrMore>
  </define>
  <define name="its-local.nons.attributes" combine="interleave">
    <zeroOrMore>
      <attribute>
        <nsName ns="http://www.w3.org/XML/1998/namespace"/>
      </attribute>
    </zeroOrMore>
  </define>
  <start>
    <choice>
      <ref name="its-rules"/>
      <ref name="its-span"/>
      <ref name="its-ruby"/>
      <ref name="its-standoff"/>
    </choice>
  </start>
</grammar>

```

[Source file: [schemas/its20-elements.rng](#)]

(RELAX NG compact syntax version of schema)

Example 98: RELAX NG schema for all ITS local attributes

```

<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns="http://relaxng.org/ns/structure/1.0">
  <include href="its20.rng"/>
  <start>
    <group>
      <optional>
        <ref name="its-local.attributes"/>
      </optional>
      <optional>
        <ref name="its-attribute.version"/>
      </optional>
    </group>
  </start>
</grammar>

```

[Source file: [schemas/its20-attributes.rng](#)]

(RELAX NG compact syntax version of schema)

3. *Base RELAX NG schema for ITS*: All ITS elements and attributes referenced by previous two schemas are defined in the base RELAX NG schema for ITS.

Example 99: Base RELAX NG schema for ITS

```

<?xml version="1.0" encoding="UTF-8"?>
<grammar ns="http://www.w3.org/2005/11/its" xmlns:a="http://relaxng.org/ns/compat"
  <include href="its20-types.rng"/>
  <define name="its-attribute.translate">
    <attribute name="its:translate">
      <ref name="its-translate.type"/>
    </attribute>
  </define>
  <define name="its-attribute.translate.nons">
    <attribute name="translate">
      <ref name="its-translate.type"/>
    </attribute>
  </define>
  <define name="its-attribute.dir">
    <attribute name="its:dir">
      <ref name="its-dir.type"/>
    </attribute>
  </define>
  <define name="its-attribute.dir.nons">
    <attribute name="dir">
      <ref name="its-dir.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locNote">
    <attribute name="its:locNote">
      <ref name="its-locNote.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locNote.nons">
    <attribute name="locNote">
      <ref name="its-locNote.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locNoteType">
    <attribute name="its:locNoteType">
      <ref name="its-locNoteType.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locNoteType.nons">
    <attribute name="locNoteType">
      <ref name="its-locNoteType.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locNoteRef">
    <attribute name="its:locNoteRef">
      <ref name="its-locNoteRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locNoteRef.nons">
    <attribute name="locNoteRef">
      <ref name="its-locNoteRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.termInfoRef">
    <attribute name="its:termInfoRef">
      <ref name="its-termInfoRef.type"/>
    </attribute>
  </define>

```

```

    </attribute>
</define>
<define name="its-attribute.termInfoRef.nons">
  <attribute name="termInfoRef">
    <ref name="its-termInfoRef.type"/>
  </attribute>
</define>
<define name="its-attribute.term">
  <attribute name="its:term">
    <ref name="its-term.type"/>
  </attribute>
</define>
<define name="its-attribute.term.nons">
  <attribute name="term">
    <ref name="its-term.type"/>
  </attribute>
</define>
<define name="its-attribute.termConfidence">
  <attribute name="its:termConfidence">
    <ref name="its-termConfidence.type"/>
  </attribute>
</define>
<define name="its-attribute.termConfidence.nons">
  <attribute name="termConfidence">
    <ref name="its-termConfidence.type"/>
  </attribute>
</define>
<define name="its-attribute.withinText">
  <attribute name="its:withinText">
    <ref name="its-withinText.type"/>
  </attribute>
</define>
<define name="its-attribute.withinText.nons">
  <attribute name="withinText">
    <ref name="its-withinText.type"/>
  </attribute>
</define>
<define name="its-attribute.domainMapping">
  <attribute name="its:domainMapping">
    <ref name="its-domainMapping.type"/>
  </attribute>
</define>
<define name="its-attribute.domainMapping.nons">
  <attribute name="domainMapping">
    <ref name="its-domainMapping.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigGranularity">
  <attribute name="its:disambigGranularity">
    <ref name="its-disambigGranularity.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigGranularity.nons">
  <attribute name="disambigGranularity">
    <ref name="its-disambigGranularity.type"/>
  </attribute>
</define>

```

```

<define name="its-attribute.disambigConfidence">
  <attribute name="its:disambigConfidence">
    <ref name="its-disambigConfidence.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigConfidence.nons">
  <attribute name="disambigConfidence">
    <ref name="its-disambigConfidence.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigClassRef">
  <attribute name="its:disambigClassRef">
    <ref name="its-disambigClassRef.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigClassRef.nons">
  <attribute name="disambigClassRef">
    <ref name="its-disambigClassRef.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigIdent">
  <attribute name="its:disambigIdent">
    <ref name="its-disambigIdent.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigIdent.nons">
  <attribute name="disambigIdent">
    <ref name="its-disambigIdent.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigIdentRef">
  <attribute name="its:disambigIdentRef">
    <ref name="its-disambigIdentRef.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigIdentRef.nons">
  <attribute name="disambigIdentRef">
    <ref name="its-disambigIdentRef.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigSource">
  <attribute name="its:disambigSource">
    <ref name="its-disambigSource.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigSource.nons">
  <attribute name="disambigSource">
    <ref name="its-disambigSource.type"/>
  </attribute>
</define>
<define name="its-attribute.localeFilterList">
  <attribute name="its:localeFilterList">
    <ref name="its-localeFilterList.type"/>
  </attribute>
</define>
<define name="its-attribute.localeFilterList.nons">
  <attribute name="localeFilterList">

```

```

    <ref name="its-localeFilterList.type"/>
  </attribute>
</define>
<define name="its-attribute.person">
  <attribute name="its:person">
    <ref name="its-person.type"/>
  </attribute>
</define>
<define name="its-attribute.person.nons">
  <attribute name="person">
    <ref name="its-person.type"/>
  </attribute>
</define>
<define name="its-attribute.personRef">
  <attribute name="its:personRef">
    <ref name="its-personRef.type"/>
  </attribute>
</define>
<define name="its-attribute.personRef.nons">
  <attribute name="personRef">
    <ref name="its-personRef.type"/>
  </attribute>
</define>
<define name="its-attribute.org">
  <attribute name="its:org">
    <ref name="its-org.type"/>
  </attribute>
</define>
<define name="its-attribute.org.nons">
  <attribute name="org">
    <ref name="its-org.type"/>
  </attribute>
</define>
<define name="its-attribute.orgRef">
  <attribute name="its:orgRef">
    <ref name="its-orgRef.type"/>
  </attribute>
</define>
<define name="its-attribute.orgRef.nons">
  <attribute name="orgRef">
    <ref name="its-orgRef.type"/>
  </attribute>
</define>
<define name="its-attribute.tool">
  <attribute name="its:tool">
    <ref name="its-tool.type"/>
  </attribute>
</define>
<define name="its-attribute.tool.nons">
  <attribute name="tool">
    <ref name="its-tool.type"/>
  </attribute>
</define>
<define name="its-attribute.toolRef">
  <attribute name="its:toolRef">
    <ref name="its-toolRef.type"/>
  </attribute>
</define>

```

```
</define>
<define name="its-attribute.toolRef.nons">
  <attribute name="toolRef">
    <ref name="its-toolRef.type"/>
  </attribute>
</define>
<define name="its-attribute.revPerson">
  <attribute name="its:revPerson">
    <ref name="its-revPerson.type"/>
  </attribute>
</define>
<define name="its-attribute.revPerson.nons">
  <attribute name="revPerson">
    <ref name="its-revPerson.type"/>
  </attribute>
</define>
<define name="its-attribute.revPersonRef">
  <attribute name="its:revPersonRef">
    <ref name="its-revPersonRef.type"/>
  </attribute>
</define>
<define name="its-attribute.revPersonRef.nons">
  <attribute name="revPersonRef">
    <ref name="its-revPersonRef.type"/>
  </attribute>
</define>
<define name="its-attribute.revOrg">
  <attribute name="its:revOrg">
    <ref name="its-revOrg.type"/>
  </attribute>
</define>
<define name="its-attribute.revOrg.nons">
  <attribute name="revOrg">
    <ref name="its-revOrg.type"/>
  </attribute>
</define>
<define name="its-attribute.revOrgRef">
  <attribute name="its:revOrgRef">
    <ref name="its-revOrgRef.type"/>
  </attribute>
</define>
<define name="its-attribute.revOrgRef.nons">
  <attribute name="revOrgRef">
    <ref name="its-revOrgRef.type"/>
  </attribute>
</define>
<define name="its-attribute.revTool">
  <attribute name="its:revTool">
    <ref name="its-revTool.type"/>
  </attribute>
</define>
<define name="its-attribute.revTool.nons">
  <attribute name="revTool">
    <ref name="its-revTool.type"/>
  </attribute>
</define>
<define name="its-attribute.revToolRef">
```

```

    <attribute name="its:revToolRef">
      <ref name="its-revToolRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.revToolRef.nons">
    <attribute name="revToolRef">
      <ref name="its-revToolRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.provRef">
    <attribute name="its:provRef">
      <ref name="its-provRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.provRef.nons">
    <attribute name="provRef">
      <ref name="its-provRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.provenanceRecordsRef">
    <attribute name="its:provenanceRecordsRef">
      <ref name="its-provenanceRecordsRefPointer.type"/>
    </attribute>
  </define>
  <define name="its-attribute.provenanceRecordsRef.nons">
    <attribute name="provenanceRecordsRef">
      <ref name="its-provenanceRecordsRefPointer.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locQualityIssuesRef">
    <attribute name="its:locQualityIssuesRef">
      <ref name="its-locQualityIssuesRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locQualityIssuesRef.nons">
    <attribute name="locQualityIssuesRef">
      <ref name="its-locQualityIssuesRef.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locQualityIssueType">
    <attribute name="its:locQualityIssueType">
      <ref name="its-locQualityIssueType.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locQualityIssueType.nons">
    <attribute name="locQualityIssueType">
      <ref name="its-locQualityIssueType.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locQualityIssueComment">
    <attribute name="its:locQualityIssueComment">
      <ref name="its-locQualityIssueComment.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locQualityIssueComment.nons">
    <attribute name="locQualityIssueComment">
      <ref name="its-locQualityIssueComment.type"/>
    </attribute>
  </define>

```

```

    </attribute>
</define>
<define name="its-attribute.locQualityIssueSeverity">
  <attribute name="its:locQualityIssueSeverity">
    <ref name="its-locQualityIssueSeverity.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityIssueSeverity.nons">
  <attribute name="locQualityIssueSeverity">
    <ref name="its-locQualityIssueSeverity.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityIssueProfileRef">
  <attribute name="its:locQualityIssueProfileRef">
    <ref name="its-locQualityIssueProfileRef.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityIssueProfileRef.nons">
  <attribute name="locQualityIssueProfileRef">
    <ref name="its-locQualityIssueProfileRef.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityIssueEnabled">
  <attribute name="its:locQualityIssueEnabled">
    <ref name="its-locQualityIssueEnabled.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityIssueEnabled.nons">
  <attribute name="locQualityIssueEnabled">
    <ref name="its-locQualityIssueEnabled.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingScore">
  <attribute name="its:locQualityRatingScore">
    <ref name="its-locQualityRatingScore.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingScore.nons">
  <attribute name="locQualityRatingScore">
    <ref name="its-locQualityRatingScore.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingVote">
  <attribute name="its:locQualityRatingVote">
    <ref name="its-locQualityRatingVote.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingVote.nons">
  <attribute name="locQualityRatingVote">
    <ref name="its-locQualityRatingVote.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingScoreThreshold">
  <attribute name="its:locQualityRatingScoreThreshold">
    <ref name="its-locQualityRatingScoreThreshold.type"/>
  </attribute>
</define>

```

```

<define name="its-attribute.locQualityRatingScoreThreshold.nons">
  <attribute name="locQualityRatingScoreThreshold">
    <ref name="its-locQualityRatingScoreThreshold.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingVoteThreshold">
  <attribute name="its:locQualityRatingVoteThreshold">
    <ref name="its-locQualityRatingVoteThreshold.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingVoteThreshold.nons">
  <attribute name="locQualityRatingVoteThreshold">
    <ref name="its-locQualityRatingVoteThreshold.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingProfileRef">
  <attribute name="its:locQualityRatingProfileRef">
    <ref name="its-locQualityRatingProfileRef.type"/>
  </attribute>
</define>
<define name="its-attribute.locQualityRatingProfileRef.nons">
  <attribute name="locQualityRatingProfileRef">
    <ref name="its-locQualityRatingProfileRef.type"/>
  </attribute>
</define>
<define name="its-attribute.mtConfidence">
  <attribute name="its:mtConfidence">
    <ref name="its-mtConfidence.type"/>
  </attribute>
</define>
<define name="its-attribute.mtConfidence.nons">
  <attribute name="mtConfidence">
    <ref name="its-mtConfidence.type"/>
  </attribute>
</define>
<define name="its-attribute.allowedCharacters">
  <attribute name="its:allowedCharacters">
    <ref name="its-allowedCharacters.type"/>
  </attribute>
</define>
<define name="its-attribute.allowedCharacters.nons">
  <attribute name="allowedCharacters">
    <ref name="its-allowedCharacters.type"/>
  </attribute>
</define>
<define name="its-attribute.storageSize">
  <attribute name="its:storageSize">
    <ref name="its-storageSize.type"/>
  </attribute>
</define>
<define name="its-attribute.storageSize.nons">
  <attribute name="storageSize">
    <ref name="its-storageSize.type"/>
  </attribute>
</define>
<define name="its-attribute.storageEncoding">
  <attribute name="its:storageEncoding">

```

```

    <ref name="its-storageEncoding.type"/>
  </attribute>
</define>
<define name="its-attribute.storageEncoding.nons">
  <attribute name="storageEncoding">
    <ref name="its-storageEncoding.type"/>
  </attribute>
</define>
<define name="its-attribute.lineBreakType">
  <attribute name="its:lineBreakType">
    <ref name="its-lineBreakType.type"/>
  </attribute>
</define>
<define name="its-attribute.lineBreakType.nons">
  <attribute name="lineBreakType">
    <ref name="its-lineBreakType.type"/>
  </attribute>
</define>
<define name="its-attribute.annotatorsRef">
  <attribute name="its:annotatorsRef">
    <ref name="its-annotatorsRef.type"/>
  </attribute>
</define>
<define name="its-attribute.annotatorsRef.nons">
  <attribute name="annotatorsRef">
    <ref name="its-annotatorsRef.type"/>
  </attribute>
</define>
<define name="its-attribute.version">
  <attribute name="its:version">
    <a:documentation>Version of ITS</a:documentation>
    <ref name="its-version.type"/>
  </attribute>
</define>
<define name="its-attribute.version.nons">
  <attribute name="version">
    <a:documentation>Version of ITS</a:documentation>
    <ref name="its-version.type"/>
  </attribute>
</define>
<define name="its-attribute.queryLanguage">
  <attribute name="its:queryLanguage">
    <ref name="its-queryLanguage.type"/>
  </attribute>
</define>
<define name="its-attribute.queryLanguage.nons">
  <attribute name="queryLanguage">
    <ref name="its-queryLanguage.type"/>
  </attribute>
</define>
<define name="its-attribute.xlink.href">
  <attribute name="xlink:href">
    <data type="anyURI"/>
  </attribute>
</define>
<define name="its-attribute.xlink.type">
  <attribute name="xlink:type">

```

```

    <value>simple</value>
  </attribute>
</define>
<define name="its-attribute.selector">
  <attribute name="selector">
    <ref name="its-absolute-selector.type"/>
  </attribute>
</define>
<define name="its-rules">
  <element name="rules">
    <a:documentation>Container for global rules</a:documentation>
    <ref name="its-rules.content"/>
    <ref name="its-rules.attributes"/>
  </element>
</define>
<define name="its-rules.content">
  <zeroOrMore>
    <ref name="its-param"/>
  </zeroOrMore>
  <zeroOrMore>
    <choice>
      <ref name="its-translateRule"/>
      <ref name="its-locNoteRule"/>
      <ref name="its-termRule"/>
      <ref name="its-dirRule"/>
      <ref name="its-rubyRule"/>
      <ref name="its-langRule"/>
      <ref name="its-withinTextRule"/>
      <ref name="its-domainRule"/>
      <ref name="its-disambiguationRule"/>
      <ref name="its-localeFilterRule"/>
      <ref name="its-provRule"/>
      <ref name="its-locQualityIssueRule"/>
      <ref name="its-mtConfidenceRule"/>
      <ref name="its-externalResourceRefRule"/>
      <ref name="its-targetPointerRule"/>
      <ref name="its-idValueRule"/>
      <ref name="its-preserveSpaceRule"/>
      <ref name="its-allowedCharactersRule"/>
      <ref name="its-storageSizeRule"/>
    </choice>
  </zeroOrMore>
</define>
<define name="its-rules.attributes">
  <ref name="its-attribute.version.nons"/>
  <optional>
    <ref name="its-attribute.xlink.href"/>
  </optional>
  <optional>
    <ref name="its-attribute.xlink.type"/>
  </optional>
  <optional>
    <ref name="its-attribute.queryLanguage.nons"/>
  </optional>
</define>
<define name="its-param">
  <element name="param">

```

```

    <a:documentation>Declaration of variable used in selectors</a:documentation>
    <ref name="its-param.content"/>
    <ref name="its-param.attributes"/>
  </element>
</define>
<define name="its-param.content">
  <text/>
</define>
<define name="its-param.attributes">
  <attribute name="name">
    <data type="string"/>
  </attribute>
</define>
<define name="its-local.attributes">
  <interleave>
    <optional>
      <ref name="its-attribute.translate"/>
    </optional>
    <optional>
      <ref name="its-attribute.dir"/>
    </optional>
    <optional>
      <choice>
        <ref name="its-attribute.locNote"/>
        <ref name="its-attribute.locNoteRef"/>
      </choice>
    </optional>
    <optional>
      <ref name="its-attribute.locNoteType"/>
    </optional>
    </optional>
    <optional>
      <ref name="its-attribute.term"/>
    </optional>
    <optional>
      <ref name="its-attribute.termInfoRef"/>
    </optional>
    <optional>
      <ref name="its-attribute.termConfidence"/>
    </optional>
    </optional>
    <optional>
      <ref name="its-attribute.withinText"/>
    </optional>
    <optional>
      <ref name="its-attribute.disambigConfidence"/>
    </optional>
    <optional>
      <ref name="its-attribute.disambigGranularity"/>
    </optional>
    <optional>
      <ref name="its-attribute.disambigClassRef"/>
    </optional>
    <optional>
      <choice>
        <group>
          <ref name="its-attribute.disambigSource"/>
          <ref name="its-attribute.disambigIdent"/>
        </group>
      </choice>
    </optional>
  </interleave>
</define>

```

```

        <ref name="its-attribute.disambigIdentRef"/>
    </choice>
</optional>
<optional>
    <ref name="its-attribute.localeFilterList"/>
</optional>
<optional>
    <choice>
        <interleave>
            <optional>
                <choice>
                    <ref name="its-attribute.person"/>
                    <ref name="its-attribute.personRef"/>
                </choice>
            </optional>
            <optional>
                <choice>
                    <ref name="its-attribute.org"/>
                    <ref name="its-attribute.orgRef"/>
                </choice>
            </optional>
            <optional>
                <choice>
                    <ref name="its-attribute.tool"/>
                    <ref name="its-attribute.toolRef"/>
                </choice>
            </optional>
            <optional>
                <choice>
                    <ref name="its-attribute.revPerson"/>
                    <ref name="its-attribute.revPersonRef"/>
                </choice>
            </optional>
            <optional>
                <choice>
                    <ref name="its-attribute.revOrg"/>
                    <ref name="its-attribute.revOrgRef"/>
                </choice>
            </optional>
            <optional>
                <choice>
                    <ref name="its-attribute.revTool"/>
                    <ref name="its-attribute.revToolRef"/>
                </choice>
            </optional>
            <optional>
                <ref name="its-attribute.provRef"/>
            </optional>
        </interleave>
        <ref name="its-attribute.provenanceRecordsRef"/>
    </choice>
</optional>
<optional>
    <choice>
        <ref name="its-attribute.locQualityIssuesRef"/>
    </choice>
    <interleave>
        <optional>

```

```

        <ref name="its-attribute.locQualityIssueType"/>
      </optional>
    </optional>
    <ref name="its-attribute.locQualityIssueComment"/>
  </optional>
  </optional>
    <ref name="its-attribute.locQualityIssueSeverity"/>
  </optional>
  </optional>
    <ref name="its-attribute.locQualityIssueProfileRef"/>
  </optional>
  </optional>
    <ref name="its-attribute.locQualityIssueEnabled"/>
  </optional>
</interleave>
</choice>
</optional>
</optional>
  <choice>
    <group>
      <ref name="its-attribute.locQualityRatingScore"/>
      <optional>
        <ref name="its-attribute.locQualityRatingScoreThreshold"/>
      </optional>
    </group>
    <group>
      <ref name="its-attribute.locQualityRatingVote"/>
      <optional>
        <ref name="its-attribute.locQualityRatingVoteThreshold"/>
      </optional>
    </group>
  </choice>
  </optional>
  <ref name="its-attribute.locQualityRatingProfileRef"/>
</optional>
</optional>
  <ref name="its-attribute.mtConfidence"/>
</optional>
</optional>
  <ref name="its-attribute.allowedCharacters"/>
</optional>
</optional>
  <ref name="its-attribute.storageSize"/>
  <optional>
    <ref name="its-attribute.storageEncoding"/>
  </optional>
  </optional>
  <ref name="its-attribute.lineBreakType"/>
</optional>
</optional>
  <ref name="its-attribute.annotatorsRef"/>
</optional>
</interleave>
</define>
<define name="its-local.nons.attributes">

```

```

<interleave>
  <optional>
    <ref name="its-attribute.translate.nons"/>
  </optional>
  <optional>
    <ref name="its-attribute.dir.nons"/>
  </optional>
  <optional>
    <choice>
      <ref name="its-attribute.locNote.nons"/>
      <ref name="its-attribute.locNoteRef.nons"/>
    </choice>
  </optional>
  <optional>
    <ref name="its-attribute.locNoteType.nons"/>
  </optional>
</optional>
<optional>
  <ref name="its-attribute.term.nons"/>
  <optional>
    <ref name="its-attribute.termInfoRef.nons"/>
  </optional>
  <optional>
    <ref name="its-attribute.termConfidence.nons"/>
  </optional>
</optional>
<optional>
  <ref name="its-attribute.withinText.nons"/>
</optional>
<optional>
  <ref name="its-attribute.disambigConfidence.nons"/>
</optional>
<optional>
  <ref name="its-attribute.disambigGranularity.nons"/>
</optional>
<optional>
  <ref name="its-attribute.disambigClassRef.nons"/>
</optional>
<optional>
  <choice>
    <group>
      <ref name="its-attribute.disambigSource.nons"/>
      <ref name="its-attribute.disambigIdent.nons"/>
    </group>
    <ref name="its-attribute.disambigIdentRef.nons"/>
  </choice>
</optional>
<optional>
  <ref name="its-attribute.localeFilterList.nons"/>
</optional>
<optional>
  <choice>
    <interleave>
      <optional>
        <choice>
          <ref name="its-attribute.person.nons"/>
          <ref name="its-attribute.personRef.nons"/>
        </choice>
      </optional>
    </interleave>
  </choice>
</optional>

```

```

</optional>
<optional>
  <choice>
    <ref name="its-attribute.org.nons"/>
    <ref name="its-attribute.orgRef.nons"/>
  </choice>
</optional>
<optional>
  <choice>
    <ref name="its-attribute.tool.nons"/>
    <ref name="its-attribute.toolRef.nons"/>
  </choice>
</optional>
<optional>
  <choice>
    <ref name="its-attribute.revPerson.nons"/>
    <ref name="its-attribute.revPersonRef.nons"/>
  </choice>
</optional>
<optional>
  <choice>
    <ref name="its-attribute.revOrg.nons"/>
    <ref name="its-attribute.revOrgRef.nons"/>
  </choice>
</optional>
<optional>
  <choice>
    <ref name="its-attribute.revTool.nons"/>
    <ref name="its-attribute.revToolRef.nons"/>
  </choice>
</optional>
<optional>
  <ref name="its-attribute.provRef.nons"/>
</optional>
</interleave>
<ref name="its-attribute.provenanceRecordsRef"/>
</choice>
</optional>
<optional>
  <choice>
    <ref name="its-attribute.locQualityIssuesRef.nons"/>
    <interleave>
      <optional>
        <ref name="its-attribute.locQualityIssueType.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueComment.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueSeverity.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueProfileRef.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueEnabled.nons"/>
      </optional>
    </interleave>
  </choice>

```

```

        </interleave>
      </choice>
    </optional>
  </optional>
  <choice>
    <group>
      <ref name="its-attribute.locQualityRatingScore.nons"/>
      <optional>
        <ref name="its-attribute.locQualityRatingScoreThreshold.nons"/>
      </optional>
    </group>
    <group>
      <ref name="its-attribute.locQualityRatingVote.nons"/>
      <optional>
        <ref name="its-attribute.locQualityRatingVoteThreshold.nons"/>
      </optional>
    </group>
  </choice>
  <optional>
    <ref name="its-attribute.locQualityRatingProfileRef.nons"/>
  </optional>
</optional>
<optional>
  <ref name="its-attribute.mtConfidence.nons"/>
</optional>
<optional>
  <ref name="its-attribute.allowedCharacters.nons"/>
</optional>
<optional>
  <ref name="its-attribute.storageSize.nons"/>
  <ref name="its-attribute.storageEncoding.nons"/>
</optional>
  <ref name="its-attribute.lineBreakType"/>
</optional>
</optional>
  <ref name="its-attribute.annotatorsRef.nons"/>
</optional>
</interleave>
</define>
<define name="its-span">
  <element name="span">
    <a:documentation>Inline element to contain ITS information</a:documentation>
    <ref name="its-span.content"/>
    <ref name="its-span.attributes"/>
  </element>
</define>
<define name="its-span.content">
  <zeroOrMore>
    <choice>
      <text/>
      <ref name="its-ruby"/>
      <ref name="its-span"/>
    </choice>
  </zeroOrMore>
</define>
<define name="its-span.attributes">

```

```

    <ref name="its-local.nons.attributes"/>
  </define>
  <define name="its-translateRule">
    <element name="translateRule">
      <a:documentation>Rule about the Translate data category</a:documentation>
      <ref name="its-translateRule.content"/>
      <ref name="its-translateRule.attributes"/>
    </element>
  </define>
  <define name="its-translateRule.content">
    <empty/>
  </define>
  <define name="its-translateRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.translate.nons"/>
  </define>
  <define name="its-locNoteRule">
    <element name="locNoteRule">
      <a:documentation>Rule about the Localization Note data category</a:document
      <ref name="its-attribute.selector"/>
      <ref name="its-attribute.locNoteType.nons"/>
      <choice>
        <ref name="its-locNote"/>
        <ref name="its-attribute.locNotePointer.nons"/>
        <ref name="its-attribute.locNoteRef.nons"/>
        <ref name="its-attribute.locNoteRefPointer.nons"/>
      </choice>
    </element>
  </define>
  <define name="its-attribute.locNotePointer.nons">
    <attribute name="locNotePointer">
      <ref name="its-relative-selector.type"/>
    </attribute>
  </define>
  <define name="its-attribute.locNoteRefPointer.nons">
    <attribute name="locNoteRefPointer">
      <ref name="its-relative-selector.type"/>
    </attribute>
  </define>
  <define name="its-locNote">
    <element name="locNote">
      <a:documentation>Localization note</a:documentation>
      <ref name="its-locNote.content"/>
      <ref name="its-locNote.attributes"/>
    </element>
  </define>
  <define name="its-locNote.content">
    <zeroOrMore>
      <choice>
        <text/>
        <ref name="its-ruby"/>
        <ref name="its-span"/>
      </choice>
    </zeroOrMore>
  </define>
  <define name="its-locNote.attributes">
    <ref name="its-local.attributes"/>

```

```

</define>
<define name="its-termRule">
  <element name="termRule">
    <a:documentation>Rule about the Terminology data category</a:documentation>
    <ref name="its-termRule.content"/>
    <ref name="its-termRule.attributes"/>
  </element>
</define>
<define name="its-termRule.content">
  <empty/>
</define>
<define name="its-termRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.term.nons"/>
  <optional>
    <choice>
      <ref name="its-attribute.termInfoPointer.nons"/>
      <ref name="its-attribute.termInfoRef.nons"/>
      <ref name="its-attribute.termInfoRefPointer.nons"/>
    </choice>
  </optional>
</define>
<define name="its-attribute.termInfoPointer.nons">
  <attribute name="termInfoPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-attribute.termInfoRefPointer.nons">
  <attribute name="termInfoRefPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-dirRule">
  <element name="dirRule">
    <a:documentation>Rule about the Directionality data category</a:documentation>
    <ref name="its-dirRule.content"/>
    <ref name="its-dirRule.attributes"/>
  </element>
</define>
<define name="its-dirRule.content">
  <empty/>
</define>
<define name="its-dirRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.dir.nons"/>
</define>
<define name="its-rubyRule">
  <element name="rubyRule">
    <a:documentation>Rule about the Ruby data category</a:documentation>
    <ref name="its-rubyRule.content"/>
    <ref name="its-rubyRule.attributes"/>
  </element>
</define>
<define name="its-rubyRule.content">
  <optional>
    <ref name="its-rubyText"/>
  </optional>

```

```

</define>
<define name="its-rubyRule.attributes">
  <ref name="its-attribute.selector"/>
  <optional>
    <ref name="its-attribute.rubyPointer.nons"/>
  </optional>
  <optional>
    <ref name="its-attribute.rpPointer.nons"/>
  </optional>
  <optional>
    <ref name="its-attribute.rtPointer.nons"/>
  </optional>
</define>
<define name="its-attribute.rubyPointer.nons">
  <attribute name="rubyPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-attribute.rpPointer.nons">
  <attribute name="rpPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-attribute.rtPointer.nons">
  <attribute name="rtPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-rubyText">
  <element name="rubyText">
    <a:documentation>Ruby text</a:documentation>
    <ref name="its-rubyText.content"/>
    <ref name="its-rubyText.attributes"/>
  </element>
</define>
<define name="its-rubyText.content">
  <text/>
</define>
<define name="its-rubyText.attributes">
  <ref name="its-local.attributes"/>
</define>
<define name="its-ruby">
  <element name="ruby">
    <a:documentation>Ruby markup</a:documentation>
    <ref name="its-ruby.content"/>
    <ref name="its-ruby.attributes"/>
  </element>
</define>
<!-- FIXME: Allow nested ruby as in HTML5 -->
<define name="its-ruby.content">
  <oneOrMore>
    <oneOrMore>
      <choice>
        <text/>
        <ref name="its-span"/>
      </choice>
    </oneOrMore>
  </oneOrMore>
</define>

```

```

    <choice>
      <ref name="its-rt"/>
      <group>
        <ref name="its-rp"/>
        <ref name="its-rt"/>
        <ref name="its-rp"/>
      </group>
    </choice>
  </oneOrMore>
</define>
<define name="its-ruby.attributes">
  <ref name="its-local.attributes"/>
</define>
<define name="its-rt">
  <element name="rt">
    <a:documentation>Ruby text</a:documentation>
    <ref name="its-rt.content"/>
    <ref name="its-rt.attributes"/>
  </element>
</define>
<define name="its-rt.content">
  <zeroOrMore>
    <choice>
      <text/>
      <ref name="its-span"/>
    </choice>
  </zeroOrMore>
</define>
<define name="its-rt.attributes">
  <ref name="its-local.attributes"/>
</define>
<define name="its-rp">
  <element name="rp">
    <a:documentation>Used in the case of simple ruby markup to specify character
that can denote the beginning and end of ruby text when user
agents do not have other ways to present ruby text distinctively
from the base text. Typically contains parenthesis.</a:documentation>
    <ref name="its-rp.content"/>
    <ref name="its-rp.attributes"/>
  </element>
</define>
<define name="its-rp.content">
  <text/>
</define>
<define name="its-rp.attributes">
  <ref name="its-local.attributes"/>
</define>
<define name="its-langRule">
  <element name="langRule">
    <a:documentation>Rule about the Language Information data category</a:documentation>
    <ref name="its-langRule.content"/>
    <ref name="its-langRule.attributes"/>
  </element>
</define>
<define name="its-langRule.content">
  <empty/>
</define>

```

```

<define name="its-langRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.langPointer.nons"/>
</define>
<define name="its-attribute.langPointer.nons">
  <attribute name="langPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-withinTextRule">
  <element name="withinTextRule">
    <a:documentation>Rule about the Elements Within Text data category</a:documentation>
    <ref name="its-withinTextRule.content"/>
    <ref name="its-withinTextRule.attributes"/>
  </element>
</define>
<define name="its-withinTextRule.content">
  <empty/>
</define>
<define name="its-withinTextRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.withinText.nons"/>
</define>
<define name="its-domainRule">
  <element name="domainRule">
    <a:documentation>Rule about the Domain data category</a:documentation>
    <ref name="its-domainRule.content"/>
    <ref name="its-domainRule.attributes"/>
  </element>
</define>
<define name="its-domainRule.content">
  <empty/>
</define>
<define name="its-domainRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.domainPointer.nons"/>
  <optional>
    <ref name="its-attribute.domainMapping.nons"/>
  </optional>
</define>
<define name="its-attribute.domainPointer.nons">
  <attribute name="domainPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-disambiguationRule">
  <element name="disambiguationRule">
    <a:documentation>Rule about the Disambiguation data category</a:documentation>
    <ref name="its-disambiguationRule.content"/>
    <ref name="its-disambiguationRule.attributes"/>
  </element>
</define>
<define name="its-disambiguationRule.content">
  <empty/>
</define>
<define name="its-disambiguationRule.attributes">
  <ref name="its-attribute.selector"/>

```

```

<optional>
  <ref name="its-attribute.disambigGranularity.nons"/>
</optional>
<optional>
  <ref name="its-attribute.disambigSource.nons"/>
</optional>
<optional>
  <choice>
    <ref name="its-attribute.disambigClassPointer.nons"/>
    <ref name="its-attribute.disambigClassRefPointer.nons"/>
  </choice>
</optional>
<optional>
  <choice>
    <group>
      <ref name="its-attribute.disambigSourcePointer.nons"/>
      <ref name="its-attribute.disambigIdentPointer.nons"/>
    </group>
    <ref name="its-attribute.disambigIdentRefPointer.nons"/>
  </choice>
</optional>
</define>
<define name="its-attribute.disambigClassPointer.nons">
  <attribute name="disambigClassPointer">
    <ref name="its-disambigClassPointer.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigClassRefPointer.nons">
  <attribute name="disambigClassRefPointer">
    <ref name="its-disambigClassRefPointer.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigIdentPointer.nons">
  <attribute name="disambigIdentPointer">
    <ref name="its-disambigIdentPointer.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigSourcePointer.nons">
  <attribute name="disambigSourcePointer">
    <ref name="its-disambigSourcePointer.type"/>
  </attribute>
</define>
<define name="its-attribute.disambigIdentRefPointer.nons">
  <attribute name="disambigIdentRefPointer">
    <ref name="its-disambigIdentRefPointer.type"/>
  </attribute>
</define>
<define name="its-localeFilterRule">
  <element name="localeFilterRule">
    <a:documentation>Rule about the LocaleFilter data category</a:documentation>
    <ref name="its-localeFilterRule.content"/>
    <ref name="its-localeFilterRule.attributes"/>
  </element>
</define>
<define name="its-localeFilterRule.content">
  <empty/>
</define>

```

```

<define name="its-localeFilterRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.localeFilterList.nons"/>
</define>
<define name="its-provRule">
  <element name="provRule">
    <a:documentation>Rule about the Provenance data category</a:documentation>
    <ref name="its-provRule.content"/>
    <ref name="its-provRule.attributes"/>
  </element>
</define>
<define name="its-provRule.content">
  <empty/>
</define>
<define name="its-provRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.provenanceRecordsRefPointer.nons"/>
</define>
<define name="its-attribute.provenanceRecordsRefPointer.nons">
  <attribute name="provenanceRecordsRefPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-externalResourceRefRule">
  <element name="externalResourceRefRule">
    <a:documentation>Rule about the External Resource data category</a:documentation>
    <ref name="its-externalResourceRefRule.content"/>
    <ref name="its-externalResourceRefRule.attributes"/>
  </element>
</define>
<define name="its-externalResourceRefRule.content">
  <empty/>
</define>
<define name="its-externalResourceRefRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.externalResourceRefPointer.nons"/>
</define>
<define name="its-attribute.externalResourceRefPointer.nons">
  <attribute name="externalResourceRefPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-targetPointerRule">
  <element name="targetPointerRule">
    <a:documentation>Rule about the Target Pointer data category</a:documentation>
    <ref name="its-targetPointerRule.content"/>
    <ref name="its-targetPointerRule.attributes"/>
  </element>
</define>
<define name="its-targetPointerRule.content">
  <empty/>
</define>
<define name="its-targetPointerRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.targetPointer.nons"/>
</define>
<define name="its-attribute.targetPointer.nons">

```

```

    <attribute name="targetPointer">
      <ref name="its-relative-selector.type"/>
    </attribute>
  </define>
  <define name="its-idValueRule">
    <element name="idValueRule">
      <a:documentation>Rule about the Id Value data category</a:documentation>
      <ref name="its-idValueRule.content"/>
      <ref name="its-idValueRule.attributes"/>
    </element>
  </define>
  <define name="its-idValueRule.content">
    <empty/>
  </define>
  <define name="its-idValueRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.idValue.nons"/>
  </define>
  <define name="its-attribute.idValue.nons">
    <attribute name="idValue">
      <ref name="its-xpath-expression.type"/>
    </attribute>
  </define>
  <define name="its-preserveSpaceRule">
    <element name="preserveSpaceRule">
      <a:documentation>Rule about the Preserve Space data category</a:documentation>
      <ref name="its-preserveSpaceRule.content"/>
      <ref name="its-preserveSpaceRule.attributes"/>
    </element>
  </define>
  <define name="its-preserveSpaceRule.content">
    <empty/>
  </define>
  <define name="its-preserveSpaceRule.attributes">
    <ref name="its-attribute.selector"/>
    <ref name="its-attribute.space.nons"/>
  </define>
  <define name="its-attribute.space.nons">
    <attribute name="space">
      <choice>
        <value>default</value>
        <value>preserve</value>
      </choice>
    </attribute>
  </define>
  <define name="its-locQualityIssueRule">
    <element name="locQualityIssueRule">
      <a:documentation>Rule about the Localization Quality Issue data category</a:documentation>
      <ref name="its-locQualityIssueRule.content"/>
      <ref name="its-locQualityIssueRule.attributes"/>
    </element>
  </define>
  <define name="its-locQualityIssueRule.content">
    <empty/>
  </define>
  <define name="its-locQualityIssueRule.attributes">
    <ref name="its-attribute.selector"/>
  </define>

```

```

<choice>
  <choice>
    <ref name="its-attribute.locQualityIssuesRef.nons"/>
    <ref name="its-attribute.locQualityIssuesRefPointer.nons"/>
  </choice>
  <interleave>
    <optional>
      <ref name="its-attribute.locQualityIssueType.nons"/>
    </optional>
    <optional>
      <ref name="its-attribute.locQualityIssueComment.nons"/>
    </optional>
  </interleave>
</choice>
<optional>
  <ref name="its-attribute.locQualityIssueSeverity.nons"/>
</optional>
<optional>
  <ref name="its-attribute.locQualityIssueProfileRef.nons"/>
</optional>
<optional>
  <ref name="its-attribute.locQualityIssueEnabled.nons"/>
</optional>
</define>
<define name="its-attribute.locQualityIssuesRefPointer.nons">
  <attribute name="locQualityIssuesRefPointer">
    <ref name="its-relative-selector.type"/>
  </attribute>
</define>
<define name="its-mtConfidenceRule">
  <element name="mtConfidenceRule">
    <a:documentation>Rule about the MT Confidence data category</a:documentation>
    <ref name="its-mtConfidenceRule.content"/>
    <ref name="its-mtConfidenceRule.attributes"/>
  </element>
</define>
<define name="its-mtConfidenceRule.content">
  <empty/>
</define>
<define name="its-mtConfidenceRule.attributes">
  <ref name="its-attribute.selector"/>
  <ref name="its-attribute.mtConfidence.nons"/>
</define>
<define name="its-allowedCharactersRule">
  <element name="allowedCharactersRule">
    <a:documentation>Rule about the Allowed Characters data category</a:documentation>
    <ref name="its-allowedCharactersRule.content"/>
    <ref name="its-allowedCharactersRule.attributes"/>
  </element>
</define>
<define name="its-allowedCharactersRule.content">
  <empty/>
</define>
<define name="its-allowedCharactersRule.attributes">
  <ref name="its-attribute.selector"/>
  <choice>
    <ref name="its-attribute.allowedCharacters.nons"/>

```

```

        <ref name="its-attribute.allowedCharactersPointer.nons"/>
    </choice>
</define>
<define name="its-attribute.allowedCharactersPointer.nons">
    <attribute name="allowedCharactersPointer">
        <ref name="its-relative-selector.type"/>
    </attribute>
</define>
<define name="its-storageSizeRule">
    <element name="storageSizeRule">
        <a:documentation>Rule about the Allowed Characters data category</a:documenta
        <ref name="its-storageSizeRule.content"/>
        <ref name="its-storageSizeRule.attributes"/>
    </element>
</define>
<define name="its-storageSizeRule.content">
    <empty/>
</define>
<define name="its-storageSizeRule.attributes">
    <ref name="its-attribute.selector"/>
    <choice>
        <ref name="its-attribute.storageSize.nons"/>
        <ref name="its-attribute.storageSizePointer.nons"/>
    </choice>
    <optional>
        <choice>
            <ref name="its-attribute.storageEncoding.nons"/>
            <ref name="its-attribute.storageEncodingPointer.nons"/>
        </choice>
    </optional>
    <optional>
        <ref name="its-attribute.lineBreakType.nons"/>
    </optional>
</define>
<define name="its-attribute.storageSizePointer.nons">
    <attribute name="storageSizePointer">
        <ref name="its-relative-selector.type"/>
    </attribute>
</define>
<define name="its-attribute.storageEncodingPointer.nons">
    <attribute name="storageEncodingPointer">
        <ref name="its-relative-selector.type"/>
    </attribute>
</define>
<define name="its-standoff">
    <choice>
        <ref name="its-provenanceRecords"/>
        <ref name="its-locQualityIssues"/>
    </choice>
</define>
<define name="its-provenanceRecords">
    <element name="its:provenanceRecords">
        <a:documentation>Standoff markup for Provenance data category</a:documentat
        <oneOrMore>
            <ref name="its-provenanceRecord"/>
        </oneOrMore>
        <attribute name="xml:id">

```

```

        <data type="ID"/>
      </attribute>
    </element>
  </define>
  <define name="its-provenanceRecord">
    <element name="its:provenanceRecord">
      <a:documentation>Provenance record used in Provenance standoff markup</a:do
      <ref name="its-provenanceRecord.attributes"/>
    </element>
  </define>
  <define name="its-provenanceRecord.attributes">
    <interleave>
      <optional>
        <choice>
          <ref name="its-attribute.person.nons"/>
          <ref name="its-attribute.personRef.nons"/>
        </choice>
      </optional>
      <optional>
        <choice>
          <ref name="its-attribute.org.nons"/>
          <ref name="its-attribute.orgRef.nons"/>
        </choice>
      </optional>
      <optional>
        <choice>
          <ref name="its-attribute.tool.nons"/>
          <ref name="its-attribute.toolRef.nons"/>
        </choice>
      </optional>
      <optional>
        <choice>
          <ref name="its-attribute.revPerson.nons"/>
          <ref name="its-attribute.revPersonRef.nons"/>
        </choice>
      </optional>
      <optional>
        <choice>
          <ref name="its-attribute.revOrg.nons"/>
          <ref name="its-attribute.revOrgRef.nons"/>
        </choice>
      </optional>
      <optional>
        <choice>
          <ref name="its-attribute.revTool.nons"/>
          <ref name="its-attribute.revToolRef.nons"/>
        </choice>
      </optional>
      <optional>
        <ref name="its-attribute.provRef.nons"/>
      </optional>
    </interleave>
  </define>
  <define name="its-locQualityIssues">
    <element name="its:locQualityIssues">
      <a:documentation>Standoff markup for Localization Quality Issue data categor
      <oneOrMore>

```

```

        <ref name="its-locQualityIssue"/>
      </oneOrMore>
      <attribute name="xml:id">
        <data type="ID"/>
      </attribute>
    </element>
  </define>
  <define name="its-locQualityIssue">
    <element name="its:locQualityIssue">
      <a:documentation>Issue recorded in Localization Quality standoff markup</a:documentation>
      <ref name="its-locQualityIssue.attributes"/>
    </element>
  </define>
  <define name="its-locQualityIssue.attributes">
    <interleave>
      <optional>
        <ref name="its-attribute.locQualityIssueType.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueComment.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueSeverity.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueProfileRef.nons"/>
      </optional>
      <optional>
        <ref name="its-attribute.locQualityIssueEnabled.nons"/>
      </optional>
    </interleave>
  </define>
</grammar>

```

[Source file: [schemas/its20.rng](#)]

(RELAX NG compact syntax version of schema)

4. *Data type definitions*: All datatypes used in the base RELAX NG schema are defined the following schema.

Example 100: RELAX NG schema with datatypes for ITS

```

<?xml version="1.0" encoding="UTF-8"?>
<grammar xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0" xmlns="http://relaxng.org/ns/structure/1.0">
  <define name="its-version.type">
    <a:documentation>Version of ITS</a:documentation>
    <data type="string">
      <param name="pattern">[0-9]+\.[0-9]+</param>
    </data>
  </define>
  <define name="its-queryLanguage.type">
    <a:documentation>The query language to be used for processing the rules</a:documentation>
    <choice>
      <value>xpath</value>
      <value>css</value>
    </choice>
  </define>
</grammar>

```

```

    <text/>
  </choice>
</define>
<define name="its-absolute-selector.type">
  <data type="string" datatypeLibrary="">
    <a:documentation>Absolute selector</a:documentation>
  </data>
</define>
<define name="its-relative-selector.type">
  <data type="string" datatypeLibrary="">
    <a:documentation>Relative selector</a:documentation>
  </data>
</define>
<define name="its-xpath-expression.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-confidence.type">
  <data type="decimal">
    <param name="minInclusive">0</param>
    <param name="maxInclusive">1</param>
  </data>
</define>
<define name="its-translate.type">
  <a:documentation>The Translate data category information to be attached to the
  <choice>
    <value>yes</value>
    <a:documentation>The nodes need to be translated</a:documentation>
    <value>no</value>
    <a:documentation>The nodes must not be translated</a:documentation>
  </choice>
</define>
<define name="its-locNote.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-locNoteType.type">
  <a:documentation>The type of localization note</a:documentation>
  <choice>
    <value>alert</value>
    <a:documentation>Localization note is an alert</a:documentation>
    <value>description</value>
    <a:documentation>Localization note is a description</a:documentation>
  </choice>
</define>
<define name="its-locNoteRef.type">
  <data type="anyURI"/>
</define>
<define name="its-termInfoRef.type">
  <data type="anyURI"/>
</define>
<define name="its-term.type">
  <a:documentation>Indicates a term locally</a:documentation>
  <choice>
    <value>yes</value>
    <a:documentation>The value 'yes' means that this is a term</a:documentation>
    <value>no</value>
    <a:documentation>The value 'no' means that this is not a term</a:documentation>
  </choice>

```

```

</define>
<define name="its-termConfidence.type">
  <ref name="its-confidence.type"/>
</define>
<define name="its-dir.type">
  <a:documentation>The text direction for the context</a:documentation>
  <choice>
    <value>ltr</value>
    <a:documentation>Left-to-right text</a:documentation>
    <value>rtl</value>
    <a:documentation>Right-to-left text</a:documentation>
    <value>lro</value>
    <a:documentation>Left-to-right override</a:documentation>
    <value>rlo</value>
    <a:documentation>Right-to-left override</a:documentation>
  </choice>
</define>
<define name="its-withinText.type">
  <a:documentation>States whether current context is regarded as "within text"<
  <choice>
    <value>yes</value>
    <a:documentation>The element and its content are part of the flow of its pa
    <value>no</value>
    <a:documentation>The element splits the text flow of its parent element and
    <value>nested</value>
    <a:documentation>The element is part of the flow of its parent element, its
  </choice>
</define>
<define name="its-domainMapping.type">
  <a:documentation>A comma separated list of mappings between values in the con
  and workflow specific values. The values may contain spaces; in
  that case they MUST be delimited by quotation marks.</a:documentation>
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-disambigGranularity.type">
  <choice>
    <value>lexical-concept</value>
    <value>ontology-concept</value>
    <value>entity</value>
  </choice>
</define>
<define name="its-disambigConfidence.type">
  <ref name="its-confidence.type"/>
</define>
<define name="its-disambigClassPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-disambigClassRefPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-disambigClassRef.type">
  <data type="anyURI"/>
</define>
<define name="its-disambigIdentRef.type">
  <data type="anyURI"/>
</define>
<define name="its-disambigIdent.type">

```

```

    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-disambigSource.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-disambigIdentPointer.type">
    <ref name="its-relative-selector.type"/>
</define>
<define name="its-disambigIdentRefPointer.type">
    <ref name="its-relative-selector.type"/>
</define>
<define name="its-disambigSourcePointer.type">
    <ref name="its-relative-selector.type"/>
</define>
<define name="its-localeFilterList.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-provenanceRecordsRefPointer.type">
    <ref name="its-relative-selector.type"/>
</define>
<define name="its-person.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-personRef.type">
    <data type="anyURI"/>
</define>
<define name="its-org.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-orgRef.type">
    <data type="anyURI"/>
</define>
<define name="its-tool.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-toolRef.type">
    <data type="anyURI"/>
</define>
<define name="its-revPerson.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-revPersonRef.type">
    <data type="anyURI"/>
</define>
<define name="its-revOrg.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-revOrgRef.type">
    <data type="anyURI"/>
</define>
<define name="its-revTool.type">
    <data type="string" datatypeLibrary=""/>
</define>
<define name="its-revToolRef.type">
    <data type="anyURI"/>
</define>
<define name="its-provRef.type">

```

```

    <list>
      <oneOrMore>
        <data type="anyURI"/>
      </oneOrMore>
    </list>
  </define>
  <define name="its-externalResourceRefPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-targetPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-idValue.type">
    <data type="string" datatypeLibrary=""/>
  </define>
  <define name="its-space.type">
    <choice>
      <value>default</value>
      <value>preserve</value>
    </choice>
  </define>
  <define name="its-locQualityIssuesRef.type">
    <data type="anyURI"/>
  </define>
  <define name="its-locQualityIssuesRefPointer.type">
    <ref name="its-relative-selector.type"/>
  </define>
  <define name="its-locQualityIssueType.type">
    <choice>
      <value>terminology</value>
      <value>mistranslation</value>
      <value>omission</value>
      <value>untranslated</value>
      <value>addition</value>
      <value>duplication</value>
      <value>inconsistency</value>
      <value>grammar</value>
      <value>legal</value>
      <value>register</value>
      <value>locale-specific-content</value>
      <value>locale-violation</value>
      <value>style</value>
      <value>characters</value>
      <value>misspelling</value>
      <value>typographical</value>
      <value>formatting</value>
      <value>inconsistent-entities</value>
      <value>numbers</value>
      <value>markup</value>
      <value>pattern-problem</value>
      <value>whitespace</value>
      <value>internationalization</value>
      <value>length</value>
      <value>uncategorized</value>
      <value>other</value>
    </choice>
  </define>

```

```

<define name="its-locQualityIssueTypePointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-locQualityIssueComment.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-locQualityIssueCommentPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-locQualityIssueSeverity.type">
  <data type="decimal">
    <param name="minInclusive">0</param>
    <param name="maxInclusive">100</param>
  </data>
</define>
<define name="its-locQualityIssueSeverityPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-locQualityIssueProfileRef.type">
  <data type="anyURI"/>
</define>
<define name="its-locQualityIssueProfileRefPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-locQualityIssueEnabled.type">
  <choice>
    <value>yes</value>
    <value>no</value>
  </choice>
</define>
<define name="its-locQualityRatingScore.type">
  <data type="decimal">
    <param name="minInclusive">0</param>
    <param name="maxInclusive">100</param>
  </data>
</define>
<define name="its-locQualityRatingVote.type">
  <data type="integer"/>
</define>
<define name="its-locQualityRatingScoreThreshold.type">
  <data type="decimal">
    <param name="minInclusive">0</param>
    <param name="maxInclusive">100</param>
  </data>
</define>
<define name="its-locQualityRatingVoteThreshold.type">
  <data type="integer"/>
</define>
<define name="its-locQualityRatingProfileRef.type">
  <data type="anyURI"/>
</define>
<define name="its-mtConfidence.type">
  <ref name="its-confidence.type"/>
</define>
<define name="its-allowedCharacters.type">
  <data type="string" datatypeLibrary=""/>
</define>

```

```

<define name="its-allowedCharactersPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-storageSize.type">
  <data type="nonNegativeInteger"/>
</define>
<define name="its-storageSizePointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-storageEncoding.type">
  <data type="string" datatypeLibrary=""/>
</define>
<define name="its-storageEncodingPointer.type">
  <ref name="its-relative-selector.type"/>
</define>
<define name="its-lineBreakType.type">
  <choice>
    <value>cr</value>
    <value>lr</value>
    <value>crlf</value>
    <value>nel</value>
  </choice>
</define>
<define name="its-annotatorsRef.type">
  <data type="string" datatypeLibrary=""/>
</define>
</grammar>

```

[Source file: [schemas/its20-types.rng](#)]

(RELAX NG compact syntax version of schema)

[Ed. note: W3C XML Schema will be provided later.]

E References (Non-Normative)



Bidi Article

Richard Ishida. [What you need to know about the bidi algorithm and inline markup](#). Article of the [W3C Internationalization Activity](#), June 2005.

CheckMate Quality Check

Okapi Project. [CheckMate – Quality Check Configuration](#). Available at http://www.opentag.com/okapi/wiki/index.php?title=CheckMate_-_Quality_Check_Configuration.

CSS 2.1

Bert Bos, Tantek Çelik, Ian Hickson Håkon Wium Lie. [Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification](#). W3C Recommendation 7 June 2011. Available at <http://www.w3.org/TR/2011/REC-CSS2-20110607/>. The latest version of [CSS2](#) is available at <http://www.w3.org/TR/CSS21/>.

DITA 1.0

Michael Priestley, JoAnn Hackos, et. al., editors. [OASIS Darwin Information Typing Architecture \(DITA\) Language Specification v1.0](#). OASIS Standard 9 May 2005. Available at <https://www.oasis-open.org/committees/download.php/15316/dita10.zip>.

DocBook

Norman Walsh and Leonard Muellner. [DocBook: The Definitive Guide](#). Available at <http://www.docbook.org/>.

I10n i18n

Richard Ishida, Susan Miller. [Localization vs. Internationalization](#). Article of the [W3C Internationalization Activity](#), January 2006.

HTML5

Ian Hickson [HTML5 – A vocabulary and associated APIs for HTML and XHTML](#). W3C Working Draft 29 March 2012. Available at <http://www.w3.org/TR/html5/>.

ISO 30042

(International Organization for Standardization). *TermBase eXchange (TBX)*. [Geneva]: International Organization for Standardization, 2008.

ITS REQ

Yves Savourel. [Internationalization and Localization Markup Requirements](#). W3C Working Draft 18 May 2006. Available at <http://www.w3.org/TR/2006/WD-itsreq-20060518/>. The latest version of [ITS REQ](#) is available at <http://www.w3.org/TR/itsreq/>.

Localizable DTDs

Richard Ishida, Yves Savourel [Requirements for Localizable DTD Design](#). Working Draft 7 July 2003. Available at <http://people.w3.org/rishida/localizable-dtds/>.

CSS Selectors Level 3

Tantek Çelik, Erika J. Etemad, Daniel Glazman, Ian Hickson, Peter Linss, John Williams [Selectors Level 3](#). W3C Recommendation 29 September 2011. Available at <http://www.w3.org/TR/css3-selectors/>.

NERD

Named Entity Recognition and Disambiguation ontology (NERD) available at: <http://nerd.eurecom.fr/ontology>

NVDL

Information technology -- Document Schema Definition Languages (DSDL) -- Part 4: *Namespace-based Validation Dispatching Language (NVDL)*. International Organization for Standardization (ISO) ISO/IEC 19757-4:2003.

OpenDocument

Michael Brauer et al. [OASIS Open Document Format for Office Applications \(OpenDocument\)](#). Oasis Standard 1 May 2005. Available at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office. The latest version of [OpenDocument](#) is available at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office.

PROV-DM

Editors tbd. [Provenance data model](#). Details to be completed. *[Ed. note: Need to complete entry for provenance data model.]*

RFC 3066

Harald Alvestrand. [Tags for the Identification of Languages](#). RFC 3066, January 2001. Available at <http://www.ietf.org/rfc/rfc3066.txt>.

Ruby-TR

Marcin Sawicki (until 10 October, 1999), Michel Suignard, Masayasu Ishikawa (石川 雅康), Martin Dürst, Tex Texin, [Ruby Annotation](#). W3C Recommendation 31 May 2001. Available at <http://www.w3.org/TR/2001/REC-ruby-20010531/>. The latest version of [Ruby Annotation](#) is available at <http://www.w3.org/TR/ruby/>.

Schematron

Information technology -- Document Schema Definition Languages (DSDL) -- Part 3: *Rule-based validation -- Schematron*. International Organization for Standardization (ISO) ISO/IEC 19757-3:2003.

TEI

Lou Burnard and Syd Bauman (eds). [Text Encoding Initiative Guidelines development version \(P5\)](#). TEI Consortium, Charlottesville, Virginia, USA, Text Encoding Initiative.

XHTML 1.0

Steven Pemberton et al. [XHTML™ 1.0 The Extensible HyperText Markup Language \(Second Edition\)](#). W3C Recommendation 26 January 2000, revised 1 August 2002. Available at <http://www.w3.org/TR/2002/REC-xhtml1-20020801/>. The latest version of [XHTML 1.0](#) is available at <http://www.w3.org/TR/xhtml1/>.

XLIFF

XLIFF reference - tbd.

XML i18n BP

Yves Savourel, Jirka Kosek, Richard Ishida. [Best Practices for XML Internationalization](#). Available at <http://www.w3.org/TR/2008/NOTE-xml-i18n-bp-20080213/>. The latest version of [xml-i18n-bp](#) is available at <http://www.w3.org/TR/xml-i18n-bp/>.

XMLSPEC

[The XML Spec Schema and Stylesheets](http://www.w3.org/2002/xmlspec/). Available at <http://www.w3.org/2002/xmlspec/>.

XSLT 1.0

James Clark. [XSL Transformations \(XSLT\) Version 1.0](http://www.w3.org/TR/1999/REC-xslt-19991116). W3C Recommendation 16 November 1999. Available at <http://www.w3.org/TR/1999/REC-xslt-19991116>. The latest version of [XSLT 1.0](http://www.w3.org/TR/xslt) is available at <http://www.w3.org/TR/xslt>.

XUL

[exTensible User Interface Language](http://www.xulplanet.com/). Available at <http://www.xulplanet.com/>.

F Checking ITS Markup Constraints With Schematron (Non-Normative)



[Ed. note: Should this be removed? Brief discussion at Prague f2f seemed to say "yes", need to check with Jirka.] [Ed. note: Jirka: I think that conclusion was that I will update this to cover ITS 2.0]

This section is informative.

Several constraints of ITS markup cannot be validated with ITS schemas. The following [\[Schematron\]](#) document allows for validating some of these constraints.

Example 101: Testing constraints in ITS markup

```
<schema xmlns="http://www.ascc.net/xml/schematron">
  <!-- Schematron document to test constraints for global and local ITS markup.
  For ITS markup definitions, see http://www.w3.org/TR/its/ . -->
  <ns prefix="its" uri="http://www.w3.org/2005/11/its"/>
  <pattern name="Check ITS Global Rules and Local Constraints, and Version Constr
  <rule context="*">
    <!-- Tests for locNoteRule -->
    <report test="self::its:locNoteRule and child::its:locNote and @its:locNote
      locNoteRule error: A locNoteRule element must not have both a locNote chi
      locNotePointer attribute.</report>
    <report test="self::its:locNoteRule and @its:locNoteRef and @its:locNoteRef
      locNoteRule error: A locNoteRule element must not have both a locNoteRef
      locNoteRefPointer attribute.</report>
    <report test="self::its:locNoteRule and child::its:locNote and @its:locNote
      error: A locNoteRule element must not have both a locNote child element a
      attribute.</report>
    <!-- Test for termRule -->
    <report test="self::its:termRule and @its:termInfoRef and @its:termInfoRef
      error: A termRule element must not have both a termInfoRef attribute and
      termInfoRefPointer attribute.</report>
    <report test="self::its:termRule and @its:termInfo and @its:termInfoPointer
      A termRule element must not have both a termInfo attribute and a termInfo
      attribute.</report>
    <report test="self::its:termRule and @its:termInfoRef and @its:termInfoPoin
      error: A termRule element must not have both a termInfoRef attribute and
      attribute.</report>
    <!-- Test for rubyRule -->
    <report test="self::its:rubyRule and child::its:rubyText and @its:rtPointer
      A rubyRule element must not have both a rubyText child element and a rtPo
      attribute.</report>
    <!-- Test for locNote (local) -->
    <report test="@its:locNote and @its:locNoteRef"> Local ITS usage error: The
      and the locNoteRef attribute must not be used together.</report>
    <!-- Test for term (local) -->
    <report test="@its:termInfoRef and not(its:term) and not(self::its:termRule
```

```

        error: A termInfoRef attribute must not appear locally without a term attribute
    <!-- Version attribute test -->
    <report test="/*/@its:version != @its:version"> The version attribute at the
        at the rules element must not specify different versions of ITS.</report>
    </rule>
</pattern>
</schema>

```

[Source file: [examples/xml/its-constraints-check-schematron.xml](#)]

G Conversion NIF2ITS (Non-Normative)



The following algorithm relies on [Example 25](#). It is assumed that the example has been converted to NIF, leading to the [output](#) exemplified for the [ITS2NIF conversion algorithm](#).

As a natural language processing (NLP) tool, we choose [DBpedia Spotlight](#). For this example let's assume DBpedia Spotlight linked "Ireland" to DBpedia:

```

<http://example.com/exampledoc.html#offset_21_28>
  rdf:type                str:String ;
  itsrdf:disambigIdentRef <http://dbpedia.org/resource/Ireland> .
<http://dbpedia.org/resource/Ireland>
  rdf:type                <http://nerd.eurecom.fr/ontology#Country> .

```

The conversion algorithm to generate ITS out of NIF consists of two steps.

- STEP 1: Send the text to any NIF web service, which creates the NLP annotation. The output of the Web service will be a NIF representation.
- STEP 2: Use the mapping from ITS2NIF (available after [step 7](#) of the ITS2NIF algorithm) to reintegrate annotations in the original ITS annotated document.

For step 2, three cases can occur.

[Ed. note: Need to check that the annotations shown for case 1 and case 2 are conform to the latest definition of "disambiguation".]

CASE 1: The NLP annotation created in NIF matches the text node. Solution: Attach the annotation to the parent element of the text node.

```

# Based on:
<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/b[1]/text()[1])>
  itsrdf:nif <http://example.com/exampledoc.html#offset_21_28> .
# and:
<http://example.com/exampledoc.html#offset_21_28>
  itsrdf:disambigIdentRef <http://dbpedia.org/resource/Ireland> .
# we can attach the metadata to the parent node:
<b its-disambig-ident-ref="http://dbpedia.org/resource/Dublin"
  translate="no">Ireland</b>

```

CASE 2: The NLP annotation created in NIF is a substring of the text node. Solution: Create a new element, e.g. for HTML "span". A different input example is given below as case 2 is not covered in the original example input.

```
# Input:

<html>
  <body>
    <h2>Welcome to Dublin in Ireland!</h2>
  </body>
</html>

# ITS2NIF

<http://example.com/exampledoc.html#xpath(/html/body[1]/h2[1]/text()[1])>
  itsrdf:nif <http://example.com/exampledoc.html#offset_0_29>

# DBpedia Spotlight returns:

<http://example.com/exampledoc.html#offset_21_28>
  itsrdf:disambigIdentRef <http://dbpedia.org/resource/Ireland> .

# NIF2ITS

<html>
  <body>
    <h2 >Welcome to Dublin in <span
      its-disambig-ident-ref="http://dbpedia.org/resource/Ireland" >Ireland<
    </body>
  </html>
```

Case 3: The NLP annotation created in NIF starts in one region and ends in another. Solution: No straight mapping is possible; a mapping can be created if both regions have the same parent.

H List of ITS 2.0 Global Elements and Local Attributes (Non-Normative)



The following table lists global ITS 2.0 elements inside **rules** element and local ITS 2.0 markup in XML and HTML. Note that for the local markup there are various constraints on what local attributes should be used together. Here these constraints are expressed via occurrence indicators: optional "?", alternatives "|", or groups "(...)". Please check the related sub sections in [Section 8: Description of Data Categories](#) defining local markup normatively.

In addition to below markup, ITS 2.0 provides a means to refer to the tools used to generate the markup: for XML the **annotatorsRef** attribute and for HTML the **annotators-ref** attribute. See [Section 5.8: ITS Tools Annotation](#) for details, especially the [note on annotatorsRef usage scenarios](#).

Data category	Global element inside rules element	Local XML attributes in ITS namespace	HTML attributes
Translate	translateRule	translate	translate
Localization Note	locNoteRule	(locNote locNoteRef), locNoteType ?	(its-loc-note loc-note-ref),

Data category	Global element inside rules element	Local XML attributes in ITS namespace	HTML attributes
			loc-note-type?
Terminology	termRule	term, termInfoRef?, termConfidence?	its-term, its-term-info-ref?, its-term-confidence?
Directionality	dirRule	dir	dir
Ruby	rubyRule	-	-
Language Information	langRule	xml:lang	lang
Elements Within Text	withinTextRule	withinText	its-within-text
Domain	domainRule	-	-
Disambiguation	disambiguationRule	disambigConfidence?, disambigGranularity?, at least one of (disambigClassRef, ((disambigSource, disambigIdent) disambigIdentRef))	its-disambig-confidence?, its-disambig-granularity?, at least one of (its-disambig-class-ref, ((its-disambig-source, its-disambig-ident) its-disambig-ident-ref))
Locale Filter	localeFilterRule	localeFilterList	its-locale-filter-list
Provenance	provRule	(At least one of ((person personRef), (org orgRef), (tool toolRef), (revPerson revPersonRef), (revOrg revOrgRef), (revTool revToolRef), provRef)) provenanceRecordsRef	(At least one of ((its-person its-person-ref), (its-org its-org-ref), (its-tool its-tool-ref), (its-rev-person its-rev-person-ref), (its-rev-org its-rev-org-ref), (its-rev-tool its-rev-tool-ref), its-

Data category	Global element inside rules element	Local XML attributes in ITS namespace	HTML attributes
			prov-ref)) its- provenance- records-ref
External Resource	externalResourceRefRule	-	-
Target Pointer	targetPointerRule	-	-
Id Value	idValueRule	xml:id	id
Preserve Space	preserveSpaceRule	xml:space	-
Localization Quality Issue	locQualityIssueRule	(at least one of (locQualityIssueType, locQualityIssueComment), locQualityIssueSeverity?, locQualityIssueProfileRef?, locQualityIssueEnabled?) locQualityIssuesRef	(at least one of (its-loc- quality- issue-type, its-loc- quality- issue- comment), its-loc- quality- issue- severity?, its-loc- quality- issue- profile-ref?, its-loc- quality- issue- enabled?) its-loc- quality- issues-ref
Localization Quality Rating	-	(locQualityRatingScore, locQualityRatingScoreThreshold?) (locQualityRatingVote, locQualityRatingVoteThreshold?), locQualityRatingProfileRef?	(its-loc- quality- rating- score, its- loc-quality- rating- score- threshold?) (its-loc- quality- rating-vote, its-loc- quality- rating-vote- threshold?), its-loc- quality- rating- profile-ref?

Data category	Global element inside rules element	Local XML attributes in ITS namespace	HTML attributes
MT Confidence	mtConfidenceRule	mtConfidence	its-mt-confidence
Allowed Characters	allowedCharactersRule	allowedCharacters	its-allowed-characters
Storage Size	storageSizeRule	storageSize, storageEncoding?, lineBreakType?	its-storage-size, its-storage-encoding?, lits-line-break-type?

I Revision Log (Non-Normative)



The following log records major changes that have been made to this document since the [ITS 2.0 Working Draft 23 October 2012](#).

1. Clarified usage of [Domain](#) data category in HTML in response to [issue-56](#).
2. Added the [enabled information](#) in [Section 8.17: Localization Quality Issue](#).
3. Updated the [Disambiguation](#) data category.
4. Fine tuned the algorithm to compute the result values of the [Domain](#) data category.
5. Fix on [Example 81](#): id attribute of `script` element now the same as of containing XML.
6. NIF example fix - see [action-284](#).
7. Added [a note](#) to mark CSS selectors as feature at risk, see [action-272](#).
8. Defined in [Section 5.3.2.2: Relative selector](#) that an XPath based relative selector can also be an absolute location path - see the `domainPointer` attribute in [Example 56](#) and [action-282](#).
9. Defined [Directionality](#) and [Ruby](#) as non-normative features. See [Section 1.1.1: Relation to ITS 1.0](#), [note on directionality](#), [note on ruby](#), and [action-250](#).
10. Update on [Disambiguation](#) example [Example 59](#). See [action-266](#) ([related discussion](#)).
11. Made a simplification of [Disambiguation used globally](#). See [action-267](#).
12. Added [Appendix B: Internationalization Tag Set \(ITS\) MIME Type](#), see [action-251](#).
13. Added [Section 8.19: MT Confidence](#), see [action-287](#) and [action-288](#).
14. Added [Section 5.8: ITS Tools Annotation](#) see [action-301](#).
15. Added confidence score attributes to [Disambiguation](#) and [MTConfidence](#) data categories - see [action-298](#) and [action-299](#).
16. Updated [Section 8.12: Provenance](#) - now called "*Provenance*" instead of "*Translation Agent Provenance*" - see [action-300](#).
17. Added [a note](#) to differentiate [Disambiguation](#) from [Terminology](#) data category - see [action-304](#).
18. Reworked the [Section 8.17: Localization Quality Issue](#) for global rules and standoff markup as per [action-303](#).
19. Removed placeholder for [text analysis annotation](#), since the [text analysis annotation requirement](#) is covered by the [local disambiguation](#) attribute `disambigConfidence`, in conjunction with [Section 5.8: ITS Tools Annotation](#).
20. Added explanations about ITS 2.0 and plain text in CMS to [Section 1.3.1.4: Content producers](#) and [Section 8.20.1: Definition](#) - see [action-262](#) and [action-302](#).
21. Various edits, see [summary mail](#) and [action-312](#) and [action-317](#).
22. Updated [list of pointer attributes](#) in [Section 5.3.2.2: Relative selector](#), see [action-308](#).

23. Checked [data category overview table](#), see [action-313](#), and various edits, see [summary mail](#).
24. Clarification of pointer attribute values in [Section 8.12.2: Implementation](#), see [mail for details](#).
25. Online editing call - see [call minutes](#) and [summary mail](#).
26. Updated [Section 8.12: Provenance](#) to remove all the pointers attributes, except `provenanceRecordsRefPointer`.
27. Updated [Section 8.18: Localization Quality Rating](#) to remove the global rules and adjust the thresholds.
28. Re-structured [Section 6.2: Global rules](#) and added XHTML example.
29. Made [Appendix D: Schemas for ITS](#) a normative section.
30. Moved list of data category identifiers from [Section 5.8: ITS Tools Annotation](#) to [data category overview table](#), see [action-330](#).
31. Added [Example 22](#): external rules with `rules` as the root element. See [action-328](#).
32. "HTML5" in document now replaced with "HTML", see [action-327](#).
33. Changed made during editing call 29 November, see [editing call minutes](#).
34. Made changes (see [detailed description](#)) to [descriptions of allowed values](#) for [Localization Quality Issue](#) (specifically *terminology*, *locale-violation*, and *whitespace* to respond to and clarify [points raised by Daniel Naber](#).
35. Added [Appendix H: List of ITS 2.0 Global Elements and Local Attributes](#), see [action-321](#).
36. Renaming attribute for [Section 5.8: ITS Tools Annotation](#). See [change description](#).
37. Changes related to `annotatorsRef`, see [Working Group call 2012-12-03](#) discussion.
38. Changes related to `disambigGranularity` attribute, see [Working Group call 2012-12-03](#) discussion and [action-359](#).

The following log records major changes that have been made to this document since the [ITS 2.0 Working Draft 29 August 2012](#).

1. Added a first draft of [Section 8.12: Provenance](#)
2. Added [Section 6: Using ITS Markup in HTML](#).
3. Removed inline markup declarations.
4. Addition of a `locQualityRatingVote` attribute and a `locQualityRatingVotePointer` attribute to [Section 8.18: Localization Quality Rating](#).
5. A [clarification](#) of ITS data category information and processing of content in [Section 8.1: Position, Defaults, Inheritance and Overriding of Data Categories](#).
6. Added [Section 8.20: Allowed Characters](#).
7. Added [Section 8.21: Storage Size](#).
8. Added [Section 8.19: MT Confidence](#).
9. Added [a note](#) about informative mappings of [Values for the Localization Quality Issue Type](#) to the [ITS IG wiki](#).
10. Added a [conformance clause](#) about HTML versus XML processing.
11. Added links to XML and HTML examples to the [data category overview table](#).
12. Added new kind of user to [Section 1.3.1: Potential Users of ITS](#).
13. Added the algorithm to obtain the value of the [Domain](#) data category.
14. Updated the [Allowed Characters](#) data category for the empty string case and the way to define "allow any characters"..
15. Added sections related to NIF conversion ([Section 5.7: Conversion to NIF](#) and [Appendix G: Conversion NIF2ITS](#)) and a related conformance clause [2-4](#).

The following log records major changes that have been made to this document since the [ITS 2.0 Working Draft 31 July 2012](#).

1. Added [Section 8.10: Disambiguation](#).
2. Added [Section 8.16: Preserve Space](#).
3. Added [Section 8.15: Id Value](#).
4. Added support for different query language and reworked whole XPath and CSS Selectors integration.
5. Added examples to [Section 8.13: External Resource](#).
6. Simplified [Section 8.11: Locale Filter](#).
7. Added a note about HTML and the attributes `dir` and `translate` to [Section 5.2.2: Local Selection in an XML Document](#).
8. Added definition of `param` element to [Section 5.2.1: Global, Rule-based Selection](#).
9. Added [Section 8.14: Target Pointer](#).
10. Original Ruby markup model changed to HTML5 Ruby model.
11. Updated references.
12. Added [Section 8.16: Preserve Space](#).
13. Added [Section 8.17: Localization Quality Issue](#) and the related [Appendix C: Values for the Localization Quality Issue Type](#).
14. Added [Section 8.18: Localization Quality Rating](#).
15. Added a placeholder [Section 8.19: MT Confidence](#).

The following log records major changes that have been made to this document since the [ITS 2.0 Working Draft 26 June 2012](#).

1. Various editorial changes (non-normative references update, style & grammar fixes).
2. Made clarifications to [Section 1.5: Out of Scope](#), [Section 1.6: Important Design Principles](#).
3. Added explanatory note on precedence and overriding in [Section 5.5: Precedence between Selections](#).
4. Reordered some components in [Section 1: Introduction](#).
5. Restructured [Section 1.1: Relation to ITS 1.0 and New Principles](#).
6. Added [Section 5.3.1: Choosing Query Language](#) as a stub.
7. Added [Section 8.11: Locale Filter](#).
8. Added [Section 8.9: Domain](#).
9. Added [Section 1.4.1: Support for legacy HTML content](#).
10. Added local markup in [Section 8.8: Elements Within Text](#).
11. Added [Section 8.13: External Resource](#).
12. Updated examples to use the `version` attribute with the value 2.0.

The following log records major changes that have been made to this document between the [ITS 1.0 Recommendation](#) and this document.

1. Clarified [introduction](#) to cover ITS 2.0
2. Added a subsection on the relation to ITS 1.0 to the introduction, see [Section 1.1.1: Relation to ITS 1.0](#)
3. Created HTML based declarations for various data categories, see e.g. HTML declarations for the Terminology data category and the summary for local data categories in [Section 5.2.2: Local Selection in an XML Document](#)
4. Created examples for these declarations, see e.g. [Example 44](#)
5. Added placeholders for new data categories to [Section 8: Description of Data Categories](#)
6. Added a placeholder section [Section 5.7: Conversion to NIF](#)

J Acknowledgements (Non-Normative)



This document has been developed with contributions by the MultilingualWeb-LT Working Group: Mihael Arcan (DERI Galway at the National University of Ireland, Galway, Ireland), Pablo Badía (Linguaserve), Aaron Beaton (Opera Software), Aljoscha Burchardt (German Research Center for Artificial Intelligence (DFKI) GmbH), Nicoletta Calzolari (CNR--Consiglio Nazionale delle Ricerche), Mauricio del Olmo (Linguaserve), Giuseppe Deriardi (Linguaserve), Pedro Luis Díez Orzas (Linguaserve), David Filip (University of Limerick), Leroy Finn (Trinity College Dublin), Karl Fritsche (Cocomore AG), Daniel Grasmick (Lucy Software and Services GmbH), Declan Groves (Centre for Next Generation Localisation), Moritz Hellwig (Cocomore AG), Manuel Honegger (University of Limerick), Dominic Jones (Trinity College Dublin), Milan Karásek (Moravia Worldwide), Jirka Kosek (University of Economics, Prague), Michael Kruppa (Cocomore AG), David Lewis (Trinity College Dublin), Fredrik Liden (ENLASO Corporation), Christian Lieske (SAP AG), Arle Lommel (German Research Center for Artificial Intelligence (DFKI) GmbH), Shaun McCance ((public) Invited expert), Jan Nelson (Microsoft Corporation), Pablo Nieto Caride (Linguaserve), Naoto Nishio (University of Limerick), Philip O'Duffy (University of Limerick), Des Oates (Adobe Systems Inc.), Carina Pellar (Cocomore AG), Georgios Petasis (Institute of Informatics & Telecommunications (IIT), NCSR), Georg Rehm (German Research Center for Artificial Intelligence (DFKI) GmbH), Phil Ritchie (VistaTEC), Thomas Rüdeshiem (Lucy Software and Services GmbH), Nieves Sande (German Research Center for Artificial Intelligence (DFKI) GmbH), Felix Sasaki (W3C Staff), Yves Savourel (ENLASO Corporation), Jörg Schütz (W3C Invited Experts), Ankit Srivastava (Centre for Next Generation Localisation), Tadej Štajner (Jozef Stefan Institute), Olaf-Michael Stefanov ((public) Invited expert), Najib Tounsi (Ecole Mohammadia d'Ingenieurs Rabat (EMI)), Clemens Weins (Cocomore AG).

A special thanks to Daniel Naber for introducing us to [LanguageTool](#) and for implementing [Localization Quality Issue Type](#) functionality in language tool.