|  |
| --- |
|  |

|  |
| --- |
| **TBX Starter Guide** |
| Validating TBX Files |
|  |
|  |
|  |
| **Formerly LISA Terminology Special Interest Group** |
| **10/18/2010** |
| **11/10/2010 (Rev 1)****12/20/2010 (Rev 2)****3/21/2013 (Rev 3)** |

[Chapter 1. Overview of TBX 5](#_Toc280617018)

[Overview 5](#_Toc280617019)

[Audience for this Guide 5](#_Toc280617020)

[History of TBX 5](#_Toc280617021)

[TBX-Default and TBX-Basic 6](#_Toc280617022)

[Benefits of TBX 6](#_Toc280617023)

[Structure of a Typical TBX File 7](#_Toc280617024)

[Validating TBX Files 7](#_Toc280617025)

[When to Validate TBX Files 7](#_Toc280617026)

[TBX Validation Resources 8](#_Toc280617027)

[Overview of TBX Resources 8](#_Toc280617028)

[The TBX Checker 8](#_Toc280617029)

[The Integrated RelaxNG Schema for TBX-Basic 8](#_Toc280617030)

[Sample TBX Files, the DTD, and the XCS File 8](#_Toc280617031)

[Steps for Validating TBX Files 8](#_Toc280617032)

[Error Messages 9](#_Toc280617033)

[Chapter 2. Downloading TBX Resources 11](#_Toc280617034)

[Requirements 11](#_Toc280617035)

[Downloading the TBX Checker Package 12](#_Toc280617036)

[Downloading the TBX Checker Executable File 13](#_Toc280617037)

[Chapter 3. Using the TBX Checker 15](#_Toc280617038)

[Starting the TBX Checker 15](#_Toc280617039)

[Overview of TBX Checking Demonstrations 15](#_Toc280617040)

[Demo: No Error 16](#_Toc280617041)

[Demo: Bad Attribute 16](#_Toc280617042)

[Demo: Bad Element 18](#_Toc280617043)

[Demo: Bad Element Content 20](#_Toc280617044)

[Demo: Bad Element Order 21](#_Toc280617045)

[Demo: Not Well-Formed Elements 24](#_Toc280617046)

[Chapter 4. Using the Integrated RNG Schema 27](#_Toc280617047)

[Requirements 27](#_Toc280617048)

[Organizing Resources in an <oXygen/> XML Editor Session 27](#_Toc280617049)

[Validating XML Project Documents 28](#_Toc280617050)

[Demo: Bad Attribute 29](#_Toc280617051)

[Demo: Bad Element 30](#_Toc280617052)

[Demo: Bad Element Content 30](#_Toc280617053)

[Demo: Bad Element Order 31](#_Toc280617054)

[Demo: Not Well-Formed 32](#_Toc280617055)

[Appendix A. Sample Structure of a TBX File 35](#_Toc280617056)

[Appendix B. Bibliography 39](#_Toc280617057)

[Appendix C. Glossary 40](#_Toc280617058)

#

# Chapter 1. Overview of TBX

# Overview

TermBase eXchange (TBX) is a markup language that is used to represent structured, concept-oriented terminological data in a database, which is known as a termbase. Based on XML, TBX can be used either as a native format for representing terminological data in a terminology management application or as an intermediary format for exchange purposes. TBX is an open standard and is implemented as a family of terminological markup languages (TMLs).

TBX can be used to facilitate the exchange of terminological data between two types of consumers:

* people, such as translators and terminologists
* applications and systems, such as terminology management tools and controlled authoring software

# Audience for this Guide

TBX implementers and TBX users are the primary audience for this guide. Any professional who works with a termbase might be interested in TBX file validation and analysis.

* A TBX implementer is an applications programmer who supports a company's termbase. An implementer validates TBX files and performs various programming tasks to ensure TBX compliance.
* TBX users are terminologists and other language specialists who need to analyze a terminological database for representation in TBX or who need to understand the content of a TBX file.

# History of TBX

TBX was first released by the Localization Industry Standards Association (LISA) in 2002. In 2007, LISA submitted TBX to the International Organization for Standardization (ISO) for adoption as an ISO standard. The TBX standard was co-published in December 2008 by ISO as ISO 30042:2008 and by LISA as TBX:2008. The LISA version and the ISO version of the TBX standard are identical. Though LISA was dissolved in 2011, ETSI continues to host the standards development committee, and the TBX specification is [still available online](http://www.gala-global.org/oscarStandards/tbx/tbx_oscar.pdf).

All TMLs that comply with TBX use the same core structure but might differ in which data categories are allowed.

# TBX-Default and TBX-Basic

The currently defined TBX TMLs are defined as follows:

* **TBX-Default**, which contains the following
	+ a document type definition (DTD), which is known as the TBX-Basic Core Structure DTD
	+ A complete set of data categories and their constraints. The data categories are specified in an eXtensible Constraint Specification (XCS) file.
* **TBX-Basic-** a TBX TML that contains fewer data categories than TBX-Default and some additional constraints on the core structure.
* **TBX-Glossary**- a TBX TML designed to support the interchange of glossary data among several formats: UTX-Simple, GlossML, the TBX family, and OLIF. It is designed to express only such essential data as can be unambiguously represented in all of these formats.

The focus of this guide is TBX-Basic.

# Benefits of TBX

Terminology management helps organizations to compete more easily in global markets, to maintain customer satisfaction, to control international brand, and to reduce support costs.

TBX provides the following benefits:

* **Standards for terminological data exchange.** TBX makes it easier to exchange complex terminological data among termbases by providing a standard intermediate representation.
* **Vendor neutrality.** TBX implements a standard that can be supported by all vendors of terminology management software. Also, TBX represents terminological data as generically as possible in order to maximize an application's ability to interpret and reuse the terminological data.
* **Reduced localization cost and faster time to market.** Sharing terminological data with translation and localization service providers helps them to improve accuracy and increase speed in translation. Sharing terminological data also reduces the time and cost of terminology research and revision.
* **Better control over corporate terminology assets.** TBX provides a machine-readable XML format for representing terminological data. The format improves control and reuse of terminological data across an enterprise.
* **Improved consistency and quality of translated and localized content.** Controlled terminology improves the accuracy of text, facilitating the translation process. TBX helps translators and localizers to maintain consistency and quality, which promotes customer satisfaction.

# Structure of a Typical TBX File

A TBX file, which is also known as a TBX document, is a single instance of a record of terminological data, many of which constitutes a termbase. A single TBX file, or entry in a termbase, typically contains a term, a definition, and other relevant details (such as the subject area to which the term belongs, the source of the data, the identity of the person who created the entry, and the term’s part of speech) that conform to the requirements and constraints of a particular TML, such as TBX-Basic.

For an example of the structure of a typical TBX file, see Appendix A.

Validating TBX Files

Validation is the process for determining whether a TBX file that is represented in a TBX TML is compliant with the TML. A single TBX file must meet the following requirements to be TBX compliant:

* It must be a well-formed XML file. For details about the requirements for well-formedness, see [Types of TBX Error Messages](#_Types_of_TBX).
* It must be valid according to the core structure of TBX and any additional constraints of the TBX TML.
* It must adhere to the constrained set of data categories that are specified in the XCS file.

# When to Validate TBX Files

TBX files should be validated at the following times:

* **At regular intervals during termbase development.** TBX validation could be automated to run at regular intervals and to deliver error reports in batch mode.
* **After a termbase is transferred to another organization for continued work.** Validation is especially important when the native languages that are used in the two organizations are different (for example, English and French).Validation ensures that the TBX file that contains the source language is error-free.
* **Before terminological data is imported from a source termbase to a target termbase.** The TBX files to be imported must be compared to the TBX files in the target termbase to ensure that the data categories are compatible. Modifications to the import files are typically necessary. Revalidation ensures that the import files are error-free before they are imported into the target termbase.

**Note**: Adding, deleting, or modifying data categories are external to the TBX validation process, and are outside the scope of this guide. See the [Bibliography](#_Appendix_B._Bibliography) for resources about this topic.

* **After terminological data is imported from the source termbase to the target termbase.** The target termbase must be validated for compliance to TBX again before authoring and editing activities resume. Revalidation ensures that the target termbase is structurally stable.

# TBX Validation Resources

## Overview of TBX Resources

As a service to TBX implementers and users, the Localization Industry Standards Association (LISA) provides at no charge software tools and sample files that support TBX/ISO 30042. Overviews follow of the software tools and sample files:

* The TBX Checker
* The Integrated RNG Schema
* Sample TBX files, DTDs, and XCS files

The TBX resources are available at <http://www.tbxconvert.gevterm.net>. Detailed instructions for accessing these resources are located in later sections in this guide.

## The TBX Checker

The TBX Checker is an open-source, cross-platform Java program that checks TBX files for compliance with well-formedness, core-structure validity, and XCS adherence. The TBX Checker's functionality is TBX-specific and exceeds that of a general-purpose XML editor.

For details, see [Chapter 3. Using the TBX Checker](#_Chapter_3._Using).

## The Integrated RelaxNG Schema for TBX-Basic

The Integrated RelaxNG (RNG) Schema is an alternative to the TBX Checker. In some instances, you might want to represent a TBX TML as an integrated schema, which combines the core structure constraints of the DTD and the additional data category constraints that are contained in a TBX-Basic XCS file.

The primary benefit of using an Integrated RNG schema is that a TBX file can be checked using a general-purpose XML tool rather than a TBX-specific tool such as the TBX Checker.

We provide a standard Integrated RNG Schema with embedded Schematron rules that can be used to validate TBX files. Schematron is a rule-based validation language that offers the primary benefit of conditionally controlling content in XML files.

In order to validate your TBX files against the Integrated RNG Schema, you must use an XML editor that supports the RelaxNG and Schematron languages. An example of such a product is the <oXygen/> ® XML editor. For details, see [Chapter 4. Using the Integrated RNG Schema](#_Chapter_4._Using).

# Sample TBX Files, the DTD, and the XCS File

For demonstration purposes, sample TBX files that contain deliberate errors are included in the TBX-Basic package available. Also, the TBX-Basic Core Structure DTD and the TBX-Basic XCS file are provided against which you can check the sample TBX files when using the TBX Checker.

# Steps for Validating TBX Files

The steps for validating TBX files can be demonstrated using the above TBX resources. Here are the basic steps:

1. Invoke a validation tool (for example, the TBX Checker or a general-purpose XML tool that supports RNG and Schematron, as applicable).
2. Specify the TBX file to be checked.
3. Make sure that the appropriate checking rules (the DTD and the XCS file, or an integrated RNG schema) are accessible.
4. Run the validation tool.
5. Evaluate the error messages and correct the errors.

The detailed tasks that you perform vary according to the specific validation tool and the resources that you use.

# Error Messages

Each type of error message that is reported by both of the validation tools contains the description of a problem and the location in the TBX file at which the error occurred. The location of the error in the file might be indicated by a line number or by a visual pointer to the line, depending on the validation tool that is used.

Most errors point to TBX elements that do not conform to the requirements that are specified in either the DTD file and the XCS file or the RNG schema. The following are the types of errors that can occur when you check TBX files:

**Bad attribute-** a type of error indicating that an element’s attribute is invalid. The following is an example of a bad attribute error message:

XCS Adherence Errors

 Unknown specification pair (admin, origin): termEntry id=c1 for the element [admin: null] (Start 37:27, End 37:50).

The type value “origin” for the <admin> element is invalid. An example of a valid type value is “source” for the <admin> element.

**Bad element** a type of error indicating that an element is invalid. The following is an example of a set of bad element error message:

XML Validation Major Errors

 Parse Exception:

 Line: 38

 Column: 42

 Message: Element type "transaction" must be declared.

 Embedded:

XML Validation Major Errors

 Parse Exception:

 Line: 41

 Column: 18

 Message: The content of element type "transacGrp" must match

 "(transac,(transacNote|date)\*)".

In this example, the element <transaction> is used erroneously. The correct element is <transac>.

**Bad element content**  a type of error indicating an invalid picklist value. The following is an example of a bad element content error message:

XCS Adherence Errors

 Invalid picklist entry: Value="preposition" in termEntry id=c1 for the element [termNote: null] (Start 57:37, End 57:59).

In this example, according to the XCS file, “masc” is an invalid value for the element <grammaticalGender>, whose valid picklist values are “masculine”, “feminine”, “neuter”, and “otherGender”. Picklist values must match exactly their representation in the XCS file.

**Bad element order** a type of error indicating a misordered element. The following is an example of a bad element order error message:

XML Validation Major Errors

 Parse Exception:

 Line: 67

 Column: 12

 Message: The content of element type "tig" must match "(term,termNote\*,(descrip|descripGrp|admin|transacGrp|note|ref|xref)\*)".

In this example, the element <termNote> must be ordered beneath the element <term>.

**Not well-formed elements**  a type of error indicating that the TBX file does not comply with the general rules of XML well-formedness.

An XML file is considered “well-formed” if it conforms to a set of syntax rules that are provided in the specification. Key features of well-formedness include the following:

* Certain characters (such as “<”) are used exclusively as special syntax characters in the XML markup language.
* The beginning, ending, and empty-element tags that are used as element delimiters are correctly nested, without missing element delimiters or overlapping delimiters.
* The names for element tags are case-sensitive, and beginning and end tags match exactly.
* A single root element contains all the other elements.

The following is an example of a not well-formed elements error message:

XML Wellformed Errors

 Parse Exception:

 Line: 44

 Column: 5

 Message: The element type "date" must be terminated by the matching end-tag "</date>".

Misspellings of XML tags or omitted closing tags are a common cause of ill-formed TBX files.

The TBX-Basic package contains the TBX-Basic specification (**TBX\_Basic\_datacategoriesVXX.pdf**), which explains the valid tags, attributes and ordering of a valid TBX-Basic file. This documentation is especially helpful for resolving validation issues.

#

# Chapter 2. Downloading TBX Resources

##

# Requirements

In order to download the TBX checking software, you must have sufficient resources:

* A Windows or UNIX operating environment.
* Access to the following Web sites: <http://www.tbxconvert.gevterm.net>
* A Java run-time environment, version 1.6 or higher. For details, see <http://java.sun.com/javase/downloads/index.jsp>

After the software has been downloaded, to use the tools for TBX checking, you must have access to or knowledge of these additional resources:

* To validate files using the TBX Checker, a simple text editor that includes a line-numbering feature; for example, TextPad, UltraEdit, or NotePad++.

 **Note**: You must undo line-numbering in the TBX file before you run the TBX Checker. Line numbers interfere with the checking procedure.

* To validate files using the RNG Schema, an XML editor that supports RNG and Schematron.

**Note**: Not all XML editors have the appropriate validation functionality. The <oXygen/>® XML editor is an example of a tool that supports RNG Schema. For details, see <http://www.oxygenxml.com>.

* The TBX-Basic specification, the DTD, and the XCS file. These resources are located in the TBX-Basic package.

# Downloading the TBX Checker Package

Follow these steps to download the TBX Checker package:

1. On your computer, create a folder in which to store the resources to be downloaded. For example, you might download resources into a folder named TBX on your C: drive.
2. Download the TBX Checker package.
	1. Download the following URL in your web browser: <http://www.tbxconvert.gevterm.net/TBXBasic-Package.zip>.
	2. Open the file **TBXBasic.zip** and extract it to the specified folder.

The package contains these files:

* + - File names that end with the **.tbx** extension are the TBX-Basic sample files that are used to demonstrate the checking process using the TBX Checker and the Integrated RNG Schema.
		- **TBXBasiccoreStrucV02.dtd** is the TBX-Basic Core Structure DTD, which is required in order to validate a TBX file when using the TBX Checker.
		- **TBXBasicRNGV02.rng** is the Integrated RNG Schema for TBX-Basic, which is required in order to validate a TBX-Basic file using a standard XML validator.
		- **TBXBasicXCSV02.xcs** is the TBX-Basic eXtensible Constraints Specification file, which is required in order to validate a TBX-Basic file when using the TBX Checker.
		- **tbxxcsdtd.dtd** is the DTD that is used to validate the XCS file. This file is useful if a user wants to create a customized XCS file for validating a customized TBX TML. It verifies that the XCS file is error-free.

# Downloading the TBX Checker Executable File

Follow these steps to download the TBX Checker executable file:

1. At the <http://sourceforge.net/projects/tbxutil/> Web site, click the big green **Download** button. The button will also contain text similar to **tbxcheck-1.2.9.jar**.
2. Save the executable file (for example, **tbxcheck-1.2.9.jar)** to the folder that contains the TBX Checker package, (C:\TBX in our above example).

For details about using the TBX Checker, see [Chapter 3. Using the TBX Checker](#_Chapter_3._Using). For details about using the Integrated RNG Schema, see [Chapter 4. Using the Integrated RNG Schema](#_Chapter_4._Using).

#

# Chapter 3. Using the TBX Checker

##

## Starting the TBX Checker

To start the TBX Checker, double-click the executable file previously downloaded (for example, **tbxcheck-1.2.9.jar)**. The user interface for the TBX Checker appears as follows:



Figure : TBX Checker User Interface

The Logging field is used to specify the level of error reporting at which to run the TBX Checker. The levels are: Severe, Warning, Info, Config, Fine, Finer, and Finest. INFO is the default. The **Open** button is used to navigate to the TBX file that you want to check.

## Overview of TBX Checking Demonstrations

The TBX package that you downloaded contains these sample TBX files, which illustrate typical TBX file-checking errors:

* [No error](#_Demo:_No_Errors)
* [Bad attribute](#_Demo:_Bad_Attributes)
* [Bad element](#_Demo:_Bad_Elements)
* [Bad element content](#_Demo:_Bad_Element)
* [Bad element order](#_Demo:_Bad_Element_1)
* [Not well-formed element](#_Demo:_Not_Well-Formed)

The next sections demonstrate each type of error and corrective steps.

## Demo: No Error

To check a TBX file means to check its conformance to the specified DTD and the XCS file. The default DTD that you downloaded is named **TBXBasiccoreStructV02.dtd.**

Follow these steps to check a TBX file:

1. Using the default INFO logging level, specify the TBX file to be validated by clicking **Open**.

The **Choose TBX File to Check** window appears.

1. Navigate to **tbx-basic-samples.tbx**, which is a valid file and contains no errors. Double-click the filename to run the TBX Checker.

The follow message indicates that the file was successfully validated. This file contains no errors.



Figure : Message for Successful TBX File Validation

1. Click **OK** for log details.



Figure : Informational Messages

The preceding informational messages indicate normal activities. This TBX file contains no errors.

1. Close the window.

## Demo: Bad Attribute

Follow these steps to check the demo file for bad attributes.

1. From the TBX Checker UI, click **Open** and select **tbx\_basic\_sample\_badattribute.tbx**. Double-click the filename to run the TBX Checker.

The TBX Checker checks the file and displays the following error messages:

XCS Adherence Errors

 Unknown specification pair (admin, origin): termEntry id=c1 for the element [admin: null] (Start 38:27, End 37:50).

XCS Adherence Errors

 Unknown specification pair (termNote, pos): termEntry id=c2 for the element [termNote: null] (Start 90:28, End 90:43).

The error message reports that the element attributes “origin” in line 37 and “pos” in line 89 are unknown.

1. View the TBX file to locate these errors.
	1. Using a simple text editor, open the **tbx\_basic\_sample\_badattribute.tbx** file, and apply line numbers.

**Note**: Undo line-numbering in the TBX file before you run the TBX Checker. Line numbers interfere with the checking procedure.

* 1. Locate lines 38 and 90. An excerpt follows:

36 <descripGrp>

37 <descrip type="definition">This is a sample definition at the entry level.</descrip>

38 <admin type="origin">Terminology SIG</admin>

39 </descripGrp>

.

.

.

90 <termNote type="pos">noun</termNote>

The XCS file does not support the values “origin” and “pos” for these attributes.

1. Check the XCS file for the valid values for the <admin> and <termNote> specifications.
	1. Using a simple text editor, open the file **TBXBasicXCSV02.xcs.**
	2. Search for the admin specification that contains the value “source.”

 An excerpt of the specification follows that contains the search string:

<adminSpec name="source" datcatId="ISO12620A-1019"> <contents/>

</adminSpec>

 The correct value is “source” rather than “origin” for this attribute.

* 1. Search for the <termNote> specification that contains the value “noun” because it is a known correct value.

 An excerpt of the specification follows that contains the search string:

<termNoteSpec name="partOfSpeech" datcatId="ISO12620A-020201">

 <contents datatype="picklist" forTermComp="yes">noun verb adjective adverb properNoun other</contents>

</termNoteSpec>

 The correct value is “partOfSpeech” rather than “pos” for this attribute.

1. Correct both of these errors in the **tbx­\_ basic\_sample\_badattribute.tbx** file and recheck the file.

The following message indicates that the file was successfully validated. It contains no errors.

C:\TBX2010\tbx\_basic\_sample\_badattribute-V1c.tbx is a TBX conformant file.

1. Click **OK** for log details.

The informational messages that appear in the log on your computer indicate normal activities. This TBX file contains no errors.

1. Close the window.

## Demo: Bad Element

Follow these steps to check the demo file for bad elements.

1. From the TBX Checker UI, click **Open** and select **tbx\_basic\_sample\_badelement.tbx.**  Double-click the filename to run the TBX Checker.

The TBX Checker checks the file and displays the following error messages:

XML Validation Major Errors

 Parse Exception:

 Line: 39

 Column: 42

 Message: Element type "transaction" must be declared.

 Embedded:

XML Validation Major Errors

 Parse Exception:

 Line: 42

 Column: 18

 Message: The content of element type "transacGrp" must match

 "(transac,(transacNote|date)\*)".

 Embedded:

XML Validation Major Errors

 Parse Exception:

 Line: 105

 Column: 16

 Message: Element type "comment" must be declared.

 Embedded:

XML Validation Major Errors

 Parse Exception:

 Line: 110

 Column: 12

 Message: The content of element type "tig" must match

 (term,termNote\*,(descrip|descripGrp|admin|transacGrp|note|ref|xref)\*)".

 Embedded:

**Note**: The error messages contain the valid declarations from the DTD.

The error message reports that elements within two element groupings (on lines 39 and 105) are undeclared.

1. View the TBX file to locate the first validation error.
	1. Using a simple text editor, open the **tbx\_basic\_sample\_badelementorder.tbx** file and apply line numbers.

 **Note**: Undo line-numbering in the TBX file before you run the TBX Checker. Line numbers interfere with the checking procedure.

* 1. Locate the <transacGrp> grouping that starts on line 38. An excerpt follows:

38 <transacGrp>

39 <transaction type="transactionType">origination</transaction>

40 <transacNote type="responsibility" target="US5001">Jane</transacNote>

41 <date>2007-07-22</date>

42 </transacGrp>

Recall the DTD declaration from the error message that indicates that the correct child element is <transac> rather than <transaction>.

Message: The content of element type "transacGrp" must match

 "(transac,(transacNote|date)\*)".

1. In line 39, correct the error by changing both instances of <transaction> to <transac>.
2. Locate the next validation error that occurs one line 105. An excerpt follows:

79 <tig>

.

.

.

105 <comment> This is a sample entry with some data categories at the term or

106 language level</comment>

.

.

.

110 </tig>

 The error messages indicate that the <comment> element is undeclared and that this particular statement does not conform to the correct syntax for a <tig> element.

1. Correct the error by changing <comment > to <note> in both instances in lines 105 and 106.
2. Recheck the file.

 The follow message indicates that the file was successfully validated. It contains no errors.

C:\TBX2010\tbx\_basic\_sample\_badelement-V1c.tbx is a TBX conformant file.

1. Click **OK** for log details.

The informational messages that appear in the log on your computer indicate normal activities. This TBX file contains no errors.

1. Close the window.

## Demo: Bad Element Content

Follow these steps to check the demo file for bad element content.

1. From the TBX Checker UI, click **Open** and select **tbx\_basic\_sample\_badelementcontent.tbx.**  Double-click the filename to run the checker.

The checker checks the file and displays the following error messages:

XCS Adherence Errors

 Invalid picklist entry: Value="preposition" in termEntry id=c1 for the element [termNote: null] (Start 58:37, End 58:59).

XCS Adherence Errors

 Invalid picklist entry: Value="deletion" in termEntry id=c2 for the element [transac: null] (Start 108:40, End 108:58).

The error message reports that the picklist values “preposition” in line 57 and “deletion” in line 107 are invalid.

1. View the TBX file to locate these errors.
	1. Using a simple text editor, open the **tbx\_basic\_sample\_badelementcontent.tbx** file and apply line numbers.

**Note**: Undo line-numbering in the TBX file before you run the TBX Checker. Line numbers interfere with the checking procedure.

* 1. Locate lines 58 and108. An excerpt follows:

56 <term>scheduled operation</term>

57 <termNote type="partOfSpeech">preposition</termNote>

58 <termNote type="termType">fullForm</termNote>

.

.

.

106 <transacGrp>

107 <transac type="transactionType">deletion</transac>

108 <transacNote type="responsibility" target=”US5002”>John</transacNote>

The XCS file does not support the values “preposition” and “deletion” for these attributes.

1. Check the XCS file for the valid values for the <partOfSpeech > and <transact type> attributes.
	1. Using a simple text editor, open the file **TBXBasicXCSV02.xcs.**
	2. Search for the “partOfSpeech” attribute.

 An excerpt of the specification follows that contains the search string:

<termNoteSpec name="partOfSpeech" datcatId="ISO12620A-020201">

 <contents datatype="picklist" forTermComp="yes">noun verb adjective adverb properNoun other</contents>

</termNoteSpec>

 The value “preposition” is unsupported.

* 1. Search for the “transactionType” attribute.

 An excerpt of the specification follows that contains the search string:

<transacSpec name="transactionType" datcatId="ISO12620A-1001">

 <contents datatype="picklist">origination modification</contents>

 </transacSpec>

 The value “deletion” is unsupported.

1. Correct both of these errors in the **tbx­\_ basic\_sample\_badelementcontent.tbx** file, choosing correct values, such as “noun” and “origination,” respectively, and recheck the file.

The follow message indicates that the file was successfully validated. It contains no errors.

C:\TBX2010\tbx\_basic\_sample\_badelementcontent-V1c.tbx is a TBX conformant file.

1. Click **OK** for log details.

The informational messages that appear in the log on your computer indicate normal activities. This TBX file contains no errors.

1. Close the window.

## Demo: Bad Element Order

Follow these steps to check the demo file for bad element order.

1. From the TBX Checker UI, click **Open** and select **tbx\_basic\_sample\_badelementorder.tbx.**  Double-click the filename to run the TBX Checker.

The TBX Checker checks the file and displays the following error messages:

XML Validation Major Errors

 Parse Exception:

 Line: 68

 Column: 12

 Message: The content of element type "tig" must match "(term,termNote\*,(descrip|descripGrp|admin|transacGrp|note|ref|xref)\*)".

 Embedded:

XML Validation Major Errors

 Parse Exception:

 Line: 111

 Column: 12

 Message: The content of element type "tig" must match "(term,termNote\*,(descrip|descripGrp|admin|transacGrp|note|ref|xref)\*)".

 Embedded:

XML Validation Major Errors

 Parse Exception:

 Line: 112

 Column: 15

 Message: The content of element type "langSet" must match "((descrip|descripGrp|admin|transacGrp|note|ref|xref)\*,tig+)".

 Embedded:

The error message reports the incorrect ordering of elements within two element groupings (on lines 68, 111, and 112).

1. View the TBX file to locate the first validation error.
	1. Using a simple text editor, open the **tbx\_basic\_sample\_badelementorder.tbx** file, and apply line numbers.

**Note**: Undo line-numbering in the TBX file before you run the TBX Checker. Line numbers interfere with the checking procedure.

* 1. Locate the <tig> grouping that ends on line 68. An excerpt follows:

53 <tig>

54 <term>scheduled operation</term>

55 <termNote type="partOfSpeech">verb</termNote>

56 <termNote type="termType">fullForm</termNote>

57 <termNote type="grammaticalGender">masculine</termNote>

58 <termNote type="administrativeStatus">preferredTerm-admn-sts</termNote>

59 <termNote type="geographicalUsage">Canada</termNote>

60 <descripGrp>

61 <descrip type="context">One hour is required between scheduled operations.</descrip>

62 <admin type="source">Tivoli Storage Manager Administrator's Guide</admin>

63 </descripGrp>

64 <termNote type="termLocation">menuItem</termNote>

65 <admin type="customerSubset">IBM</admin>

66 <admin type="source">IBM</admin>

67 <admin type="projectSubset">Tivoli Storage Manager</admin>

68 </tig>

The error message includes information about the correct syntax for a <tig> element. This TBX code includes a misordered <termNote> element on line 64.

1. Correct the error by moving the <termNote> to a location before the <descripGrp> element (cut line 64 and paste it beneath line 59), as follows:

 59 <termNote type="geographicalUsage">Canada</termNote>

 60 <termNote type="termLocation">menuItem</termNote>

 61 <descripGrp>

1. Locate the next validation error that occurs within the <langSet> language grouping level, between lines 78 and 116. An excerpt follows:

77 <langSet xml:lang="en">

78 <term>unscheduled operation</term>

79 <descrip type="definition">This is a sample definition at the language

80 level. This one has no source information required. Therefore, it

81 is not embedded in a descripGrp.</descrip>

82 <tig>

83 <termNote type="termType">fullForm</termNote>

84 <termNote type="grammaticalGender">masculine</termNote>

85 <termNote type="administrativeStatus">admittedTerm

86 sts</termNote>

87 <termNote type="geographicalUsage">en-US</termNote>

88 <termNote type="partOfSpeech">noun</termNote>

89 <termNote type="termLocation">radioButton</termNote>

90 <descrip type="context">Unscheduled operations should be recorded

91 in a log.</descrip>

92 <admin type="customerSubset">SAX Manufacturing</admin>

93 <admin type="source">Manufacturing Process Manual V2</admin>

94 <admin type="projectSubset">Service department</admin>

95 <admin type="customerSubset">SAX Manufacturing</admin>

96 <admin type="source">Manufacturing Process Manual V2</admin>

97 <admin type="projectSubset">Service department</admin>

98 <transacGrp>

99 <transac type="transactionType">origination</transac>

100 <transacNote type="responsibility"

101 target="US5001">Jane</transacNote>

102 <date>2007-07-22</date>

103 </transacGrp>

104 <transacGrp>

105 <transac type="transactionType">modification</transac>

106 <transacNote type="responsibility"

107 target="US5002">John</transacNote>

108 <date>2007-07-23</date>

109 </transacGrp>

110 <note>This is a sample entry with some data categories at the term

111 or language level</note>

112 <ref type="crossReference" target="c1">scheduled operation</ref>

113 <xref type="externalCrossReference"

114 target="http://www.lisa.org">LISA Web Site </xref>

115 </tig>

116 </langSet>

The error message includes information about the correct syntax for the <langSet> and <tig> elements. This TBX file includes a misordered <term> element on line 78.

1. Correct the error by moving the <term> element into the <tig> element (cut line 78 and paste it below line 82).

82 <tig>

83 <term>unscheduled operation</term>

84 <termNote type="termType">fullForm</termNote>

1. Recheck the file.

The following message indicates that the file was successfully validated. It contains no errors.

C:\TBX2010\tbx\_basic\_sample\_badelementorder-V1c.tbx is a TBX conformant file.

1. Click **OK** for log details.

The informational messages that appear in the log on your computer indicate normal activities. This TBX file contains no errors.

1. Close the window.

## Demo: Not Well-Formed Elements

Follow these steps to check the demo file for elements that are not well-formed.

1. From the TBX Checker UI, click **Open** and select **tbx\_basic\_sample\_notwellformed.tbx.**  Double-click the filename to run the checker.

The TBX Checker checks the file and displays the following error messages:

XML Wellformed Errors

 Parse Exception:

 Line: 45

 Column: 5

 Message: The element type "date" must be terminated by the matching end-tag "</date>".

The error message reports that the closing tag </date> on line 45 has been omitted.

1. View the TBX file to locate the validation error.
	1. Using a simple text editor, open the **tbx\_basic\_sample\_notwellformed.tbx** file, and apply line numbers.

**Note**: Undo line-numbering in the TBX file before you run the TBX Checker. Line numbers interfere with the checking procedure.

* 1. Locate the <date > element on line 44. An excerpt follows:

41 <transacGrp>

42 <transac type="transactionType">modification</transac>

43 <transacNote type="responsibility" target="US5002">John</transacNote>

44 <date>2007-07-23

45 </transacGrp>

The error message reports that the closing </date> tag has been omitted.

1. In line 44, correct the error by adding the closing tag </date>. Save the file.
2. Recheck the file.

The TBX Checker checks the file and displays the following error messages:

C:\TBX2010\tbx\_basic\_sample\_notwellformed-V1c.tbx is a TBX conformant file. XML Wellformed Errors

 Parse Exception:

 Line: 79

 Column: 56

 Message: The element type "termNote" must be terminated by the matching end-tag "</termNote>".

The error message reports that the closing </termNote > tag has been omitted.

1. View the TBX file again. In line 79, correct the error by changing the closing tag from </termnote> to </termNote>. As an XML format, TBX is case-sensitive. Save the file.

78 <term>unscheduled operation</term>

79 <termNote type="termType">fullForm</termnote>

80 <termNote type="grammaticalGender">masculine</termNote>

81 <termNote type="administrativeStatus">admittedTerm-admn-sts</termNote>

82 <termNote type="geographicalUsage">en-US</termNote>

83 <termNote type="partOfSpeech">noun</termNote>

1. Recheck the file.

The follow message indicates that the file was successfully validated. It contains no errors.

C:\TBX2010\tbx\_basic\_sample\_notwellformed-V1c.tbx is a TBX conformant file.

1. Click **OK** for log details.

The informational messages that appear in the log on your computer indicate normal activities. This TBX file contains no errors.

1. Close the window.

# Chapter 4. Using the Integrated RNG Schema

## Requirements

All the resources that you need to use the RNG Schema are located in the TBX Checker package. For complete requirements, see [Chapter 2. Downloading TBX Resources](#_Chapter_2._Downloading).

**Note:** The <oXygen/> XML editor does not recognize the TBX-Basic sample filenames that end with the .**tbx** extension, which is unique to TBX. Therefore, you must change the **.tbx** extension to **.xml** before you can use them in the validation demonstrations. For example, you must change the filename from **tbx\_basic\_samples.tbx** to **tbx\_basic\_samples.xml**.

## Organizing Resources in an <oXygen/> XML Editor Session

To validate the sample TBX files, the schema and the TBX files must be organized as a project, and they must be accessible from an <oXygen/> XML Editor session. The XML Editor is used in the following example.

Follow these steps to organize resources in an XML Editor session.

1. Invoke an <oXygen/> XML Editor session.
2. In the Project panel, create a new project by clicking the New Project icon . A generic project named **newProject.xpr** appears in the Project panel.
3. Add the schema file and the TBX demo files that you previously renamed using the **.xml** extension).
	1. In the Project panel, right-click **newProject.xpr** from the menu.
	2. In the Add Files window, navigate to the folder that contains the TBX package; in this example, the TBX2010 folder.
	3. Select the applicable files in the folder and click **Open**.

The selected files are copied to the new project in the XML editor session.

The following is a view of the TBX sample source files and the RNG schema in the Project panel:



Figure : TBX-Basic project files as seen in <oXygen/> XML Editor

1. Select all of the XML files in the project. Right click on the selection, choose **Validate** and then **Configure Validation Scenario(s)**. You’ll see the dialog below:



Figure : <oXygen/>'s dialog box to configure validation using the Integrated RNG schema

Select **New** and in the next dialog box specify the use of **TBXBasicRNGV02.rng** as the validating schema. Make sure to check the box labeled **Embedded Schematron Rules**. When you save the settings and close the dialog, you will be ready to validate the files using the RNG schema provided.

## Validating XML Project Documents

To validate all of the XML files in the project against the Integrated RNG Schema, select **Validate all project files** from the project menu. The associated icon looks similar to this: .

Errors are reported in the Batch Validation Errors panel. The following is an example of the error output:



Figure : Batch Validation Errors Panel

Note that no errors are reported for **tbx\_basic\_samples.xml**, which is an error-free file. Also, notice that <oXygen/> reports possible fixes for each error which is found.

The next sections demonstrate each type of error and recovery actions.

* [Bad attribute](#_Demo:_Bad_Attribute)
* [Bad element](#_Demo:_Bad_Element_2)
* [Bad element content](#_Demo:_Bad_Element_3)
* [Bad element order](#_Demo:_Bad_Element_4)
* [Not well-formed element](#_Demo:_Not_Well-Formed_1)

## Demo: Bad Attribute

Follow the steps in this section to correct the problems reported in these error messages.



Figure : Validation errors for tbx\_basic\_sample\_badattribute.xml

1. Click the first error message in the Batch Validation Errors panel.
2. The XML file opens in the Text view of the editor window. The cursor is positioned in the line that contains the error.

36 <descripGrp>

37 <descrip type="definition">This is a sample definition at the entry level.</descrip>

38 <admin type="origin">Terminology SIG</admin>

39 </descripGrp>

 On line 37, the value “origin” is unsupported for this element.

The correct value is “source.”

 Correct the error.

1. Click the next message in the error panel.

90 <termNote type="pos">noun</termNote>

 The cursor is positioned in line 90, where “pos” is the specified value for the attribute <termNote type=me>. The value “pos” is unsupported for this element. The correct value is “partOfSpeech”.

1. Correct the error, and save the file.
2. In the file panel, to re-validate this file, select the Validation icon .
3. The Batch Validation Errors panel is refreshed.

 No errors are reported for the **tbx\_basic\_sample\_badattribute.xml** file.

## Demo: Bad Element

Follow the steps in this section to correct the problems reported in these error messages:



Figure : Validation errors for tbx\_basic\_sample\_badelement.xml

1. Click the first message in the Batch Validation Errors panel.
2. The XML file opens in the Text view of the editor window. The cursor is positioned in the line that contains the error.

38 <transacGrp>

39 <transaction type="transactionType">origination</transaction>

40 <transacNote type="responsibility" target="US5001">Jane</transacNote>

41 <date>2007-07-22</date>

42 </transacGrp>

1. The message reports that the <transaction> element is unknown.

 Correct the error by changing <transaction> to <transac>.

1. In the error panel, click the next message, which reports that the <transacNote> element requires the <transac> element. You already recovered from this error by changing <transaction> to <transac> in the previous line.
2. In the error panel, click the next message, which reports that the <comment> element is unknown.

105 <comment> This is a sample entry with some data categories at the term or

106 language level</comment>

1. Correct the error by changing <comment> to <note>, and save the file.
2. In the file panel, to re-validate this file, select the validation icon .

 The Batch Validation Errors panel is refreshed.

 No errors are reported for the **tbx\_basic\_sample\_badelement.xml** file.

## Demo: Bad Element Content

Follow the steps in this section to correct the problems reported in these error messages:



Figure : Validation errors for tbx\_basic\_sample\_badelementcontent.xml

1. Click the first message in the Batch Validation Errors panel.
2. The XML file opens in the Text view of the editor window. The cursor is positioned in the line that contains the error.

57 <term>scheduled operation</term>

58 <termNote type="partOfSpeech">preposition</termNote>

59 <termNote type="termType">fullForm</termNote>

1. The message reports the correct picklist values for the <partOfSpeech> element.

 On line 58, the value “preposition” is an unsupported picklist value. An example of a correct value is “noun”.

 Correct the error by changing the value from “preposition” to “noun”.

1. In the error panel, click the next message, which reports the correct picklist values for the <transactionType> element.

 On line 108, the value “deletion” is also an unsupported picklist value. An example of a correct value is “origination”.

107 <transacGrp>

108 <transaction type="transactionType">deletion</transaction>

109 <transacNote type="responsibility" target="US5001">Jane</transacNote>

110 <date>2007-07-22</date>

111 </transacGrp>

 Correct the error by changing the value from “deletion” to “origination” and save the file.

1. In the file panel, to re-validate this file, select the validation icon .

 The Batch Validation Errors panel is refreshed.

 No errors are reported for the **tbx\_basic\_sample\_badelementcontent.xml** file.

## Demo: Bad Element Order

Follow the steps in this section to correct the problems reported in these error messages:

**Validation Errors: Bad Element Order**



Figure : Validation errors for tbx\_basic\_sample\_badelementorder.xml

1. Click the first message in the Batch Validation Errors panel.

 The XML file opens in the Text view of the editor window. The cursor is positioned in the line that contains the error.

63 </descripGrp>

64 <termNote type="termLocation">menuItem</termNote>

65 <admin type="customerSubset">IBM</admin>

66 <admin type="source">IBM</admin>

67 <admin type="projectSubset">Tivoli Storage Manager</admin>

 The message reports that on line 64 the <termNote> element is misordered.

1. Correct the error by moving the <termNote> to a location before the <descripGrp> element (cut line 64 and paste it beneath line 59).
2. In the error panel, click the next message, which reports that on line 77 the <term> element is misordered.

77 <langSet xml:lang="en">

78 <term>unscheduled operation</term>

79 <descrip type="definition">This is a sample definition at the language

80 level. This one has no source information required. Therefore, it 80 is not embedded in a descripGrp.</descrip>

81 <tig>

82 <termNote type="termType">fullForm</termNote>

1. Correct the error by moving the <term> element inside the <tig> level (cut line 78 and paste it below line 81).
2. Revalidate the file. In the file panel, select the validation icon .

 The Batch Validation Errors panel is refreshed.

 No errors are reported for the **tbx\_basic\_sample\_badelementorder.xml** file.

## Demo: Not Well-Formed

Follow the steps in this section to correct the problems reported in these error messages:

**Validation Errors: Not Well-Formed**



Figure : Validation errors for tbx\_basic\_sample\_notwellformed.xml

1. Click the first message in the Batch Validation Errors panel.

 The XML file opens in the Text view of the editor window. The cursor is positioned in the grouping that contains the error.

41 <transacGrp>

42 <transac type="transactionType">modification</transac>

43 <transacNote type="responsibility" target="US5002">John</transacNote>

44 <date>2007-07-23

45 </transacGrp>

 The error message reports that on line 45 the closing </date> tag has been omitted

1. Correct the error by adding the closing tag </date>.
2. In the file panel, select the validation icon  to re-validate this file.

1. The schema validates the file and displays the following error messages in the error panel:

 **Revalidation Errors: Not Well-Formed**



Figure : Validation errors for the modified version of tbx\_basic\_sample\_notwellformed.xml

1. The error message reports that the closing </termNote > tag has been omitted.
2. In the error panel, click the first message, which positions the cursor in line 79.

78 <term>unscheduled operation</term>

79 <termNote type="termType">fullForm</termnote>

80 <termNote type="grammaticalGender">masculine</termNote>

81 <termNote type="administrativeStatus">admittedTerm-admn-sts</termNote>

82 <termNote type="geographicalUsage">en-US</termNote>

83 <termNote type="partOfSpeech">noun</termNote>

1. Correct the error by changing the closing tag from </termnote> to </termNote>. TBX is case-sensitive.

 Save the file.

1. Revalidate the file.

 In the file panel, select the validation icon .

 The Batch Validation Errors panel is refreshed.

 No errors are reported for the **tbx\_basic\_sample\_badelementwellformed.xml** file.

# Appendix A. Sample Structure of a TBX File

<?xml version='1.0'?>

 <!DOCTYPE martif PUBLIC "ISO 12200:1999A//DTD MARTIF core (DXFcdV04)//EN" "TBXcdv04.dtd">

 <martif type="TBX" xml:lang="en-US" >

 <martifHeader>

 <fileDesc><sourceDesc><p>This is a sample TBX-Basic file from the LISA Terminology Special Interest Group (www.lisa.org/term). The entries in this file are for demonstration purposes only and do not reflect actual terminology data. Any references to real companies are fabricated for demonstration purposes only.</p></sourceDesc>

 </fileDesc>

 <encodingDesc><p type='DCSName'>SYSTEM "TBXBasic-XCS-v1a.xml"</p></encodingDesc>

 </martifHeader>

<text>

<body>

 <termEntry id="c1">

 <descrip type="subjectField">manufacturing</descrip>

 <descripGrp>

 <descrip type="definition">This is a sample definition at the entry level.

 </descrip>

 <admin type="source">Terminology SIG</admin>

 </descripGrp>

 <xref type="externalCrossReference" target="http://www.lisa.org">LISA Web site</xref>

 <ref type="cross-reference" target="c2">unscheduled operation</ref>

 <transacGrp>

 <transac type="terminologyManagementTransactions">origination</transac>

 <date>20070722</date>

 </transacGrp>

 <transacGrp>

 <transac type="terminologyManagementTransactions">origination</transac>

 <transacNote type="responsibility" target="US5001">Jane</transacNote>

 </transacGrp>

 <transacGrp>

 <transac type="terminologyManagementTransactions">modification</transac>

 <date>20070723</date>

 </transacGrp>

 <transacGrp>

 <transac type="terminologyManagementTransactions">modification</transac>

 <transacNote type="responsibility" target="US5002">John</transacNote>

 </transacGrp>

 <note>This is a sample entry with some data categories at the entry level.</note>

 <langSet xml:lang="en-US">

 <tig>

 <term>scheduled operation</term>

 <termNote type="partOfSpeech">verb</termNote>

 <termNote type="termType">fullForm</termNote>

 <termNote type="grammaticalGender">masculine</termNote>

 <termNote type="administrativeStatus">preferred</termNote>

 <termNote type="geographicalUsage">Canada</termNote>

 <descrip type="context">One hour is required between scheduled operations.</descrip>

 <descrip type="termLocation">menu item</descrip>

 <admin type="customerSubset">IBM</admin>

 <admin type="source">IBM</admin>

 <admin type="projectSubset">Marketing campaign X</admin>

 </tig>

 </langSet>

 </termEntry>

 <termEntry id="c2">

 <descrip type="subjectField">manufacturing</descrip>

 <langSet xml:lang="en-US">

 <descripGrp>

 <descrip type="definition">This is a sample definition at the language level.</descrip>

 <admin type="source">Dictionary of manufacturing</admin>

 </descripGrp>

 <tig>

 <term>unscheduled operation</term>

 <termNote type="termType">fullForm</termNote>

 <termNote type="grammaticalGender">masculine</termNote>

 <termNote type="administrativeStatus">admitted</termNote>

 <termNote type="geographicalUsage">Ireland</termNote>

 <termNote type="partOfSpeech">noun</termNote>

 <descrip type="context">Unscheduled operations should be recorded in a log.</descrip>

 <descrip type="termLocation">radio button</descrip>

 <ref type="cross-reference" target="c1">scheduled operation</ref>

 <admin type="customerSubset">SAX Manufacturing</admin>

 <admin type="source">Manufacturing Process Manual V2</admin>

 <admin type="projectSubset">Service department</admin>

 <xref type="externalCrossReference" target="http://www.lisa.org">LISA Web Site </xref>

 <transacGrp>

 <transac type="terminologyManagementTransactions">origination</transac>

 <date>20070722</date>

 </transacGrp>

 <transacGrp>

 <transac type="terminologyManagementTransactions">origination</transac>

 <transacNote type="responsibility" target="US5001">Jane</transacNote>

 </transacGrp>

 <transacGrp>

 <transac type="terminologyManagementTransactions">modification</transac>

 <date>20070723</date>

 </transacGrp>

 <transacGrp>

 <transac type="terminologyManagementTransactions">modification</transac>

 <transacNote type="responsibility" target="US5002">John</transacNote>

 </transacGrp>

 <note>This is a sample entry with some data categories at the term or language level</note>

 </tig>

 </langSet>

 </termEntry>

</body>

<back>

 <refObjectList type="respPerson">

 <refObject id="US5001">

 <item type="fname">Jane</item>

 <item type="surname">Doe</item>

 <item type="email">jane\_doe@mymail.com</item>

 <item type="role">approver</item>

 </refObject>

 <refObject id="US5002">

 <item type="fname">John</item>

 <item type="surname">Smith</item>

 <item type="email">john\_smith@mymail.com</item>

 <item type="role">inputter</item>

 </refObject>

 </refObjectList>

</back>

</text>

</martif>

# Appendix B. Bibliography

ISO 12620-1:2003 (E). *Terminology and other language resources — Data categories — Part 1:*

*Specification of data categories and management of a data category registry for language resources*

ISO 12620-2:2003 (E). *Terminology and other language resources — Data categories — Part 2: Data category selection (DCS) for electronic terminological resources (ETR)*

ISO 16642:2003. *Computer applications in terminology -- Terminological markup framework*

ISO 30042:2008. *Systems to manage terminology, knowledge and content -- TermBase eXchange (TBX)* (<http://www.ttt.org/oscarstandards/tbx/>)

# Appendix C. Glossary

**data category**

A classification of data for inclusion in a terminological database record. Examples of data categories include Context, Definition, Part of Speech, Gender, and Domain or Subject.

**document type description**

A file that specifies how the markup tags in a group of XML files should be interpreted by an application that displays, prints, or otherwise processes the documents. Short form: DTD.

**DTD**

See document type description.

**eXtensible Markup Language**

A markup language that structures information by tagging it for content, meaning, or use. Structured information contains both content (for example, words or numbers) and an indication of what role the content plays. For example, content in a section heading has a different meaning from content in a database table. Short form: XML.

**eXtensible Constraint Specification**

An XML file that identifies data categories and their constraints for a specific TBX TML. Short form: XCS.

**integrated RNG schema**

The combination of a Document Type Description and an eXtensible Constraint Specification into a single file. This schema is in RelaxNG format and contains embedded Schematron rules. The Localization Industry Standards Organization provides the Integrated RNG Schema at no charge as a service to TBX implementers and users.

**ISO 30042**

A standard for the representation of terminological data in TBX format. It was developed by the Localization Industry Standards Association and was submitted to the International Organization for Standardization for approval in 2007. The TBX standard was co-published in December 2008 by ISO as ISO 30042:2008 and by LISA as TBX:2008. The LISA version and the ISO version of the TBX standard are identical.

**localization**

The process of adapting a product to meet the language, cultural, and other requirements of a specific target environment or market so that customers can use their own languages and conventions when using the product. Translation of the user interface, system messages, and documentation is part of localization.

**localization service provider**

a person or an organization that can be contracted to provide a localization service for a customer. See also localization. Short form: LSP.

**RelaxNG Schema**

A simple schema language for XML that is based on Relax and TREX. A RelaxNG Schema specifies a pattern for the structure and content of an XML file. The RelaxNG Schema has been approved by the Organization for the Advancement of Structured Information Standards.

**RNG Schema**

See integrated RNG schema.

**Schematron**

a language for making assertions about the presence or absence of patterns in XML files. Schematron is an ISO standard that is commonly used for business rules validation and constraint checking.

**TBX**

See Term Base eXchange.

**TBX-Basic**

A subset of the TBX Standard modules; specifically, a limited set of data categories. Like TBX Standard, TBX-Basic is a Terminological Markup Language.

**TBX Checker**

An open-source, cross-platform tool written in Java that checks an instance of a TBX file for conformance to the specified files. The Localization Industry Standards Organization provides the RNG Generator at no charge as a service to TBX implementers and users. TBX Checker's functionality that is specific to TBX exceeds that of a general-purpose XML validating parser.

**TBX Standard**

a framework that includes a core structure, or a document type definition (DTD), and a set of data categories that are commonly used in termbases and their constraints in the form of an eXtensible constraint specification (XCS). TBX Standard is considered to be a terminological markup language.

**TBX validation**

The process of validating a TBX source file against a Document Type Description and an eXtensible Constraint Specification or an RNG schema, as applicable.

**Term Base eXchange**

An open standard for representing structured, concept-oriented terminological data for exchange purposes. The terminological data is often multilingual. TBX is implemented as a terminological markup framework language. The Localization Industry Standards Organization provides TBX at no charge as a service to TBX implementers and users. Short form: TBX.

**termbase**

See terminological database.

**terminological markup language**

An XML application for describing a terminological database that conforms to the constraints that are specified in ISO 16642. An example of a TML is TBX and the default XCS file, which is referred to as TBX default TML. TBX Basic is also a TML.

**terminology management system**

A collection of integrated tools and processes for harvesting, storing, retrieving, and managing terminological resources.

**terminological database**

A collection of terms, definitions, and associated metadata that is managed by a software application.

**TML**

See terminological markup language.

**well-formedness**

A quality of an XML file, demonstrating that the document conforms to the rules of a Document Type Definition or an XML Schema, both of which define the legal elements of an XML file.

**XCS**

See eXtensible Constraint Specification.

**XML**

See eXtensible Markup Language.