INTERNATIONAL STANDARD

ISO 30042

Second edition
2019-04

Management of terminology resources — TermBase eXchange (TBX)

Gestion des ressources terminologiques — TermBase eXchange (TBX)

Reference number
ISO 30042:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 37, *Language and terminology*, Subcommittee SC 3, *Management of terminology resources.*

This second edition cancels and replaces the first edition (ISO 30042:2008), which has been technically revised.

The main changes compared to the previous edition are as follows:

— industry-defined dialects consisting of data category selections corresponding to the needs of specific communities have been introduced;

— the XCS formalism has been removed and replaced with the requirement that the dialect be described and its name be declared on the root element of every TBX document instance;

— the DTD for the core structure has been replaced with a schema language-neutral definition;

— this document, containing the essential core and normative content, has been separated from ancillary content produced and distributed publicly by stakeholders;

— a simplified DCT (Data Category as Tag) style has been added alongside the traditional TBX style of DCA (Data Category as Attribute);

— xml namespaces have been introduced as a means for declaring the data categories used in a given TBX dialect (for DCT style).

NOTE Additional details about these and other changes are available on the TBX Info website[15].

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document defines a framework for representing structured terminological data, referred to as TermBase eXchange (TBX). Within this framework, a variety of industry standards, known as dialects, for specific types of terminology interchange scenarios and terminological data collections can be defined.

TBX is designed to support various types of processes involving terminological data, including analysis, descriptive representation, dissemination, and exchange in various computer environments. The primary purpose of TBX is the exchange of terminological data. For example, it facilitates:

- integrating or converting terminological data from multiple sources;
- comparing the contents of various terminological data collections;
- querying multiple terminological data collections by passing data through a common intermediate format on a batch or dynamic basis;
- placing data on an online site for download or public feedback;
- making terminology available dynamically in networked applications through a web service.

A TBX-compliant dialect can facilitate the exchange of terminological data between users, which include people such as translators and writers, as well as applications and systems, such as computer assisted translation tools and controlled authoring software. Therefore, it can be used for both human-oriented and machine-oriented terminological data processing. In this manner, it can enable the flow of terminological information between technologies and systems throughout the information production cycle, both inside an organization and with outside service providers.

TBX document instances of the same defined TBX dialect are interoperable and exchangeable with minimal loss or minimal need for negotiation, because they:

- adhere to the core structure;
- use, or have access to, the same data categories; and
- comply with the same dialect-specific constraints as other instances of the same dialect.

TBX document instances developed according to ISO 30042:2008 can be converted to comply with the current version of TBX by identifying a dialect with which the document instance complies and implementing the other changes in accordance with this document. A converter is available on the TBX Info website for such purposes[15].

NOTE Supplemental resources are available to assist implementers and users of TBX dialects on the TBX Info website[15].

TBX is limited in its ability to represent presentational markup (such as bold or italics). However, presentational markup can be autogenerated from descriptive markup in a TBX document instance.

Management of terminology resources — TermBase eXchange (TBX)

1 Scope

This document explains fundamental concepts and describes the metamodel, data categories, and XML styles: DCA (Data Category as Attribute) and DCT (Data Category as Tag). It also specifies the methodology for defining TBX dialects.

The audience for this document is anyone wishing to create a new dialect compliant with TBX. This document can also be used to analyze and to understand a terminological data collection or to design a new terminology database that complies with international standards and best practices. Typical users are programmers, software developers, terminologists, analysts, and other language professionals. Intended application areas include translation and authoring.

The TBX-Core dialect is described in detail in this document. All other industry-supported dialects are out of the scope of this document.

NOTE TBX dialects are defined by industry stakeholders. Any materials needed to implement currently shared dialects are publicly available as self-contained industry specifications (see for instance the TBX Info website[15]).

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601-1, *Date and time — Representations for information interchange — Part 1: Basic rules*

ISO 8601-2, *Date and time — Representations for information interchange — Part 2: Extensions*

ISO 12620, *Management of terminology resources — Data category specifications*

ISO 16642, *Computer applications in terminology — Terminological markup framework*

ISO 21720, *XLIFF (XML Localisation interchange file format)*

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at http://www.iso.org/obp

— IEC Electropedia: available at: http://www.electropedia.org

3.1
attribute class
group of one or more related attributes

3.2

child element

element that is subordinate to another element

3.3

classification element

element used to group data categories according to their function in a *concept entry* (3.5)

EXAMPLE <admin>, which expresses data categories for administrative information, such as <admin type="originatingPerson">.

Note 1 to entry: Data categories are instantiated as the value of the *type* attribute used with a given classification element.

3.4

complementary information

CI

information supplementary to that described in *concept entries* (3.5) and shared across the *terminological data collection* (3.29)

3.5

concept entry

terminological entry

entry

part of a *terminological data collection* (3.29) which contains the terminological data related to one concept

[SOURCE: ISO 16642:2017, 3.22, modified – new terms "concept entry" and "entry" added, synonym "TE" deleted, preferred term now is "concept entry" instead of "terminological entry", Note 1 to entry deleted.]

3.6

core structure

common structure and *data categories* (3.8) that are used in all TBX *dialects* (3.12)

Note 1 to entry: The core structure is compliant with ISO 16642 (TMF).

3.7

core structure module

core module

TBX-Core module

data category module (3.9) that contains only those *data categories* (3.8) that are part of the *core structure* (3.6)

3.8

data category

class of data items that are closely related from a formal or semantic point of view

EXAMPLE /part of speech/, /subject field/, /definition/.

Note 1 to entry: A data category can be viewed as a generalization of the notion of a field in a database.

Note 2 to entry: In running text, such as in this document, data category names are enclosed in forward slashes (e.g. /part of speech/).

3.9
data category module
module
list of permissible *data categories* (3.8) and constraints on them that are used in the design of a TBX-compliant *terminological data collection* (3.29)

EXAMPLE The TBX-Core module, which includes the data categories and structure common to all TBX dialects, the TBX-Min module, which adds a minimum number of data categories needed for simple glossaries, and the TBX-Basic module, which provides for a richer set of data categories.

3.10
DCA
data category as attribute
style of representing TBX data whereby most *data categories* (3.8) are expressed as the value of a *type* attribute on an XML element declared in the corresponding schema

EXAMPLE <termNote type="partOfSpeech">adjective</termNote>.

3.11
DCT
data category as tag
style of representing TBX data whereby most *data categories* (3.8) are expressed as XML generic identifiers

EXAMPLE <partOfSpeech>adjective</partOfSpeech>.

3.12
dialect
XML markup language that validates according to the *core structure* (3.6) of TBX and allows exactly those *data categories* (3.8) at those levels specified by a particular *data category module* (3.9) or set of data category modules and complies with all other relevant constraints

Note 1 to entry: "All other relevant constraints" refers to constraints that are necessary for the dialect in question but that are not expressible in either the core structure or the data category modules, such as date formats or conditional constraints. An example of this occurs in the dialect TBX-Basic, which requires a /definition/ OR a

/context/.

3.13
display name
name of a *data category* (3.8) as it appears on a software user interface or other medium

3.14
document instance
file containing *concept entries* (3.5) represented in a TBX *dialect* (3.12)

3.15
exchange
interchange
transaction involving exporting data from one *termbase* (3.28) and importing it into another termbase

3.16
global information
GI
technical and administrative information applying to the entire *terminological data collection* (3.29)

3.17
grouping element
XML element whose purpose is to group together a set of *child elements* (3.2)

3.18

object language

language being described

3.19

PID

persistent identifier

unique identifier that ensures permanent access for a digital object by providing access to it independently of its physical location or current ownership

[SOURCE: ISO 24619:2011, 3.2.4, modified – "persistent identifier" made second preferred term, Note 1 to entry deleted.]

3.20

private dialect

dialect (3.12) intended for private use that has not been described on a publicly accessible website

3.21

public dialect

dialect (3.12) that has been described on a publicly accessible website

Note 1 to entry: An example of a publicly accessible website is TBX Info[15].

3.22

root element

first element in a TBX *document instance* (3.14)

Note 1 to entry: The root element is <tbx>.

3.23

TBX agent

program or utility which generates, reads, edits, writes, processes, stores, renders or otherwise manipulates TBX-compliant *document instances* (3.14)

3.24

TBX export

process of creating a TBX *dialect* (3.12) *document instance* (3.14) from a *termbase* (3.28) or its subset

3.25

TBX import

process of inserting terminological data from one TBX *document instance* (3.14) into an existing *termbase* (3.29)

Note 1 to entry: The existing termbase can be empty or can already contain terminological entries.

3.26

TBX Module Description

TBXMD

formalism for identifying a set of *data categories* (3.8) and their constraints for a specific *data category module* (3.9)

3.27

term component

one of the words of a multi-word term, or one of the components of a single-word term (such as a morpheme)

3.28

termbase

terminology database

database comprising a *terminological data collection* (3.29)

3.29
terminological data collection
TDC
resource consisting of *concept entries* (3.5) with associated metadata and documentary information

EXAMPLE A TBX document instance, ISO 1087.

[SOURCE: ISO 16642:2017, 3.21, modified – in the definition, “concept entries” used instead of “terminological entries”, Example added.]

3.30
working language
metalanguage used in *concept entries* (3.5) to describe *object language* (3.18) content

4 Fundamental principles and concepts

TBX refers to a framework consisting of two interacting components: a core structure and a formalism for defining data category modules. The core structure is expressed in a schema definition language such as RelaxNG (RNG). (The core is also represented by its own data category module.) This component-based approach supports the varying types of terminological data, or data categories, that are included in different terminological data collections. The approach mirrors the terminological markup framework (TMF) in that the core structure shall reflect the abstract data model of TMF in accordance with ISO 16642. In addition, it facilitates an explicit description of what any two dialects within the TBX framework have in common (the core structure) and how they differ (expressed in their respective data category modules). The combination of these two components defines a particular dialect. “TBX” without a dialect indicator is not a file format, it is not a terminology markup language, and it is not itself a dialect.

The TBX framework assumes that, because terminological data collections vary significantly, no one dialect would satisfy all user requirements. All dialects within the TBX framework adhere to the core structure, which is described in Clause 8. A RelaxNG schema for the core structure is referenced in Annex C, and the elements and attributes are described in Annex A.

Dialects can differ with respect to which data categories are allowed, and at what levels of a concept entry these data categories may occur. These constraints on the core structure are formally represented in one or more data category modules.

A data category module, or simply *module*, is a list of permissible data categories and constraints on them that are used in the design of a TBX-compliant dialect. Constraints are the permissible content of a data category (including subsets of a standard picklist value domain) and the levels of the concept entry where the data category may occur (see Clause 8).

NOTE Sample data category modules are available on the TBX Info website[15].

It is recommended that implementers of TBX adhere to ISO standards and industry guidelines governing the principles and methodologies of terminology management and the content and quality of terminological data collections, such as those described in Clause 2 and the Bibliography.

The information represented in a TBX document instance should be concept-oriented. The terms in a single entry are assumed to be synonymous unless otherwise noted.

Furthermore, if two systems both fully support a given TBX dialect, then information in that dialect can be preserved when terminological data is exported from one and imported into the other. In the context of TBX, interoperability implies this preservation of data. When different dialects of TBX are used by two systems, interoperability is reduced, and loss of data categories and their content can occur. Thus, claiming compliance to TBX without indicating the dialect does not guarantee any degree of interoperability.

5 Data categories

5.1 General

Data categories represent information about terms and concepts, for instance, /part of speech/ and /definition/. A list of data categories commonly used in termbases is provided in Annex B.
A description of these and other data categories is available in the data category repository DatCatInfo[10]. If another data category repository is used to describe data categories, it shall also comply with ISO 12620.

In running text, such as in this document, data category names are enclosed in forward slashes (e.g. /part of speech/). In a TBX document instance, and in the data category modules where data categories for a TBX dialect are declared, camel case (e.g. partOfSpeech) shall be used. Industry-accepted names for data categories in camel case are available in DatCatInfo. If the data categories in Annex B are used in a TBX document instance, the names in Annex B shall be used.

5.2 Data categories specified in the core structure module

In TBX, the following data categories are declared in the TBX-Core structure, and therefore are available to all TBX dialects, and are represented in the same way in all styles (see Clause 6):

— /date/

— /term/

— /note/

5.3 Data categories specified in the data category modules

All data categories not included in TBX-Core that are required for a particular TBX dialect are documented in the dialect's data category module or modules (see Clause 9). Such additional data categories may include, for example, /definition/, /part of speech/, /context/, /term type/, and so forth. A simple data category module such as the Min module introduces /definition/, but an additional module, such as Basic, then further extends the model by adding /context/ and other data categories. Hence, the TBX-Basic dialect consists of modules for TBX-Core, TBX-Min, and TBX-Basic.

6 DCA and DCT styles

There are two XML styles that may be used to represent terminological data: DCA (data category as attribute) and DCT (data category as tag). DCA is the style used for the examples in this document.

— DCA: <termNote type="partOfSpeech">adjective</termNote>

— DCT: <partOfSpeech>adjective</partOfSpeech>

In DCA style, most data categories are expressed as the value of the type attribute (in the above case: /part of speech/) of one of the elements declared in the core structure (in this case, <termNote>).

In DCT style, most data categories are reflected in the element generic identifier name. The corresponding core-structure element with which this data category is associated may optionally be indicated as the value of the *metaType* attribute. For example:

```
<partOfSpeech metaType="termNote">adjective</partOfSpeech>
```

In both cases, the value of the data category is the content of the XML element. These two styles are isomorphic. That is, they can be converted back and forth by an algorithm without loss of information. Even if the *metaType* attribute is omitted (e.g. <partOfSpeech>adjective</partOfSpeech>), the two representations can still be converted from one to the other if the algorithm has access to a table that indicates the core-structure element associated with each data category.

DCA style emphasizes the similarity among TBX dialects. DCA also allows all TBX dialects to be validated, at a first level, against the same schema (the core structure), by using a general-purpose XML parser.

DCT style looks more familiar to XML users who are accustomed to distinct element names rather than refinement of elements through attribute values.

NOTE Additional information about DCT style is available on the TBX Info website[15].

7 Dialects

7.1 General

Few terminology collections or applications use exactly the same set of data categories. TBX is a flexible framework because it allows user groups to select their own data categories. By doing so, they can create their own dialect adapted to their requirements. A TBX dialect complies with the core structure and implements one or more defined data category modules.

7.2 Dialect naming

Dialect names shall start with the "TBX-" prefix and end with a dialect indicator, such as "Basic", i.e., TBX-Basic. Although dialects are not standardized, industry groups and companies have in the past declared their own data models for purposes of sharing in public environments. Public dialect names can be published and thus made available for collaborative use, for instance on the TBX Info website[15].

All TBX dialects are built upon the TBX-Core module, which contains the essential data categories described in Clause 8. TBX dialects extend the core by adding a set of data category modules. For instance, one common industry dialect, TBX-Basic, consists of three modules: TBX-Core, TBX-Min and TBX-Basic (see 9.6).

If a dialect has been extended through the addition of one or more data category modules, a meaningful name qualifier shall be added to the dialect name separated by a period ".".

EXAMPLE TBX-Basic.Seo, where Seo is the name of a module which adds one or more data categories to those in the TBX-Basic dialect, in this case to include data categories for search engine optimization (SEO).

NOTE 1 Suggested subset relationships between and among public TBX dialects or private extensions thereof are available on the TBX Info website[15].

NOTE 2 Modules and dialects both begin with the "TBX-" prefix and are distinguished by using the descriptors "module" or "dialect" in conjunction with their name.

7.3 Example of a dialect

This subclause describes a fictitious TBX dialect called TBX-Sample dialect. For illustrative purposes, this dialect allows minimal terminological information.

The TBX-Sample dialect is defined as the combination of the TBX-Core module plus the TBX-Fiction module:

TBX-Sample dialect = TBX-Core module + TBX-Fiction module

NOTE This example demonstrates how the module name and dialect name can be different.

The data categories (and their accompanying constraints) included in the TBX-Fiction module are expressed in Table 1:

Table 1 — Module definition for the TBX-Fiction module

Name	PID	Classification	Values	Levels
definition	www.datcatinfo.net/datcat/DC-168	<descrip>	noteText	conceptEntry, langSec
subjectField	www.datcatinfo.net/datcat/DC-489	<descrip>	finance, manufacturing	conceptEntry
termType	www.datcatinfo.net/datcat/DC-2677	<termNote>	abbreviatedForm, fullForm	

Note that any data category with the core-structure (classification) element of <termNote> is restricted by the core structure to the termSec level. Therefore, the level of /term type/ need not be indicated.

A TBX Module Description (TBXMD) for the TBX-Fiction module is available in Annex C.

The optional *module* attribute identifies the data categories as belonging to the TBX-Fiction module.

Example TBX-Sample dialect document instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<tbx xmlns:tbx3="urn:iso:std:iso:30042:ed-2" type="TBX-Sample" style="dca" xml:lang="en">
    <tbxHeader>
        <fileDesc>
            <sourceDesc>
                <p>A sample termbase consisting of one concept entry</p>
            </sourceDesc>
        </fileDesc>
    </tbxHeader>
    <text>
        <body>
            <conceptEntry id="c1">
                <descrip type="definition" module="Fiction">a sum of money that
                is expected to be paid back with interest</descrip>
                <descrip type="subjectField" module="Fiction">finance</descrip>
                <langSec xml:lang="en">
                    <termSec>
                        <term>loan</term>
                        <termNote type="termType" module="Fiction">
                        fullForm</termNote>
                    </termSec>
                </langSec>
                <langSec xml:lang="fr">
                    <termSec>
                        <term>prêt</term>
                        <termNote type="termType" module="Fiction">
                        fullForm</termNote>
                    </termSec>
                </langSec>
            </conceptEntry>
        </body>
    </text>
</tbx>
```

7.4 Requirements for a dialect to be TBX-compliant

Data in TBX document instances that adhere to the same dialect is interchangeable. By defining industry-recognized dialects and making them available for public use, it is expected that terminological data collections world-wide will move away from idiosyncratic implementations and towards increased interoperability.

For a dialect to be compliant with TBX, it shall:

1) be an XML markup language;

2) have a compliant dialect name (see 7.2);

3) consist of the core structure module plus zero or more additional compliant modules;

4) consist of non-overlapping modules (see Clause 9), in the event there is more than one; and

5) comply with all constraints of the core module and any other modules it includes.

NOTE Non-overlapping means that identical data categories do not appear in two modules in the same dialect. Data categories are deemed to be identical if they share the same PID (persistent identifier).

Normative constraints for a new TBX-compliant dialect are as follows:

1) a new dialect shall be based on the TBX-Core module (see Clause 8);

   NOTE If compatibility with a published industry dialect is desirable, one or more data category modules may be chosen from the published information for that dialect, such as from the TBX Info website[15].

2) if data categories are needed which are not included in any existing module, a new module shall be created by following the guidelines in Clause 9;

3) the dialect shall be named according to the naming conventions in 7.2.

A TBX dialect may exist as either a public dialect or a private dialect. Public dialects respond to the needs of an identifiable community of practice and are documented by a comprehensive dialect description.

NOTE Users who wish to share their dialects may post their information on a publicly accessible website (without requiring login), such as the TBX Info website[15].

7.5 Validating a TBX document instance

The extension for a TBX document instance is `.tbx`.

TBX is a framework for defining dialects and therefore does not require the use of any specific validation method or schema definition language. Implementers may use a variety of methods to validate TBX document instances.

For a TBX document instance to be compliant with TBX it shall meet the following criteria:

1) it shall be well-formed XML;

2) it shall comply with the TBX core structure as elaborated in Clause 8 and Annex A;

3) it shall specify the name of a compliant dialect as the value of the *type* attribute on the <tbx> root element;

   NOTE A TBX document instance that does not have a dialect name declared on the root element is not compliant with this document. Its content would be unpredictable and thus not interoperable.

4) it shall validate against an integrated schema (or equivalent) that incorporates the constraints of the specified dialect. If there is a discrepancy between schema representations when validating XML document instances, the requirements of this document shall be met;

   NOTE If the value of the *type* attribute on the <tbx> root element is TBX-Core, then the TBX document instance includes no data categories that are not in the core.

Although individual dialects are not themselves standardized, any specified dialect that claims compliance with this standard can be published as a public or private dialect on a readily accessible website such as TBX Info[15].

The schema used for validation shall use an established schema definition language such as RNG and Schematron in accordance with ISO/IEC 19757-2 and ISO/IEC 19757-3. A sample of the RNG schema for TBX-Core is available on the TBX Info website[15]. It may be extended to include the constraints of data category modules, thereby producing an integrated schema that can be used to validate a TBX document instance for a given dialect.

7.6 Requirements for compliant TBX agents

Common TBX agents include exporters, importers, modifiers and enrichers. Modifiers do not change the dialect of the TBX document instance, whereas enrichers extend a TBX dialect document instance with data from one or more additional modules so that the resulting TBX dialect document instance complies with a superset dialect definition.

For a TBX agent to export, import, modify or enrich TBX-compliant data the following requirements shall be met:

Exporter:

- shall produce a TBX document instance compliant with 7.5;
- shall export all data categories that are required in all modules of the declared dialect;
- for data categories from Annex B, shall use the data category names as shown in Annex B;

  EXAMPLE In TBX, the data category /context/ is a segment of text that includes the term. If a particular termbase uses /example/ for this purpose, the exporter converts all instances of /example/ to /context/.

- shall export data categories compliant with customary industry practice for TBX data category definitions (see Clause 5);
- may export subsets of the available terminological data.

  NOTE Subsets of the termbase (a certain number of entries) and subsets of entries themselves (a certain selection of data categories) qualify as long as the subset complies with the declared dialect name.

Importer:

- shall import all data categories according to the target termbase definition, if available in the imported TBX dialect document instance;
- may import a subset of data categories corresponding to a subset dialect, in the event the importer does not support the TBX dialect declared in the imported document instance;
- may convert data categories missing in the target termbase definition to /note/ by means of special converters;
- shall reject the importing TBX document instance if it does not comply with the dialect name specified on its <tbx> root element.

Modifier:

- shall not affect compliance to the dialect declared on the <tbx> root element;
- shall not change the dialect name specified as the value of the *type* attribute on the <tbx> root element;
- may add any data categories allowed by the target termbase definition while complying with its constraints;
- may add and delete entries. Added entries shall contain data for all mandatory data categories as defined for the target termbase definition.

Enricher:

- shall amend the dialect name specified as the value of the *type* attribute on the <tbx> root element in compliance with the dialect naming requirements (see 7.2).

EXAMPLE <tbx xmlns="urn:iso:std:iso:30042:ed-2" type="TBX-Basic.Seo" style="dca" xml:lang="en">

All processes should create relevant log files to capture any information loss or conflicts.

8 The core structure

8.1 General

The core structure defines the common structure and data categories that are used in all TBX dialects.

In case of conflict or dispute about some technical aspect of the core structure, the description in this clause takes precedence over all schema representations (RNG, DTD, XCS, and so forth).

NOTE 1 The TBX Info website[15] provides an RNG schema that complies with this clause and can serve as a model for other schemas if necessary.

NOTE 2 The core structure as described in this clause uses DCA style.

The TBX-Core namespace is: `urn:iso:std:iso:30042:ed-2`

The above namespace shall be used as the default namespace for TBX document instances of all dialects.

8.2 Metamodel

The core structure reflects the TMF (ISO 16642) metamodel, as shown in Figure 1. The terminological data collection (TDC) corresponds to a TBX document instance.

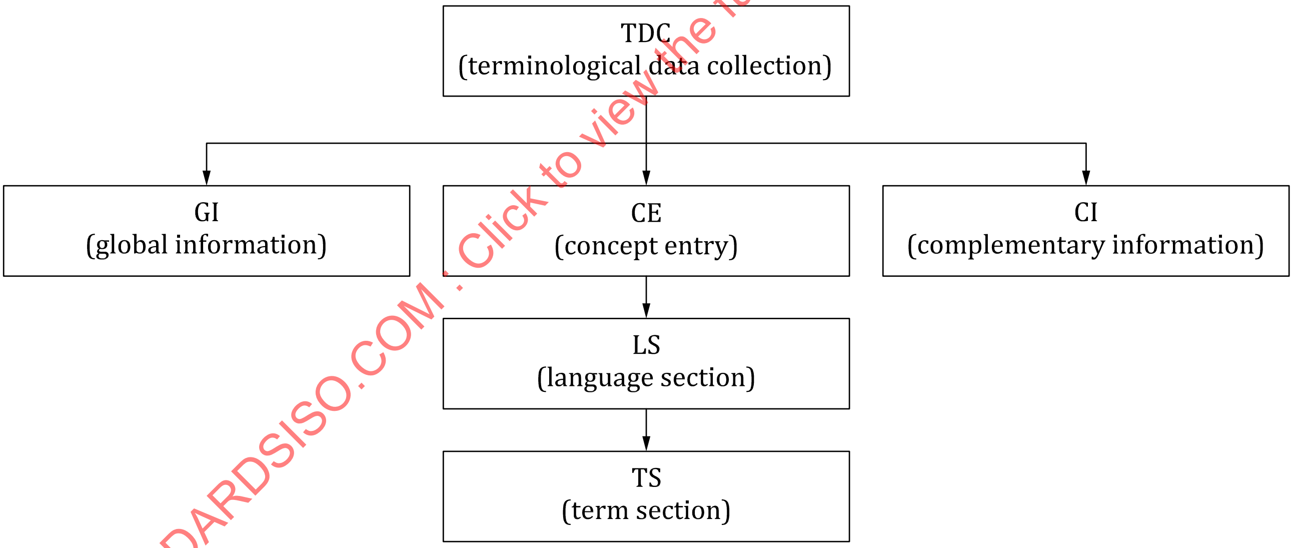


Figure 1 — TBX metamodel

Cardinality:

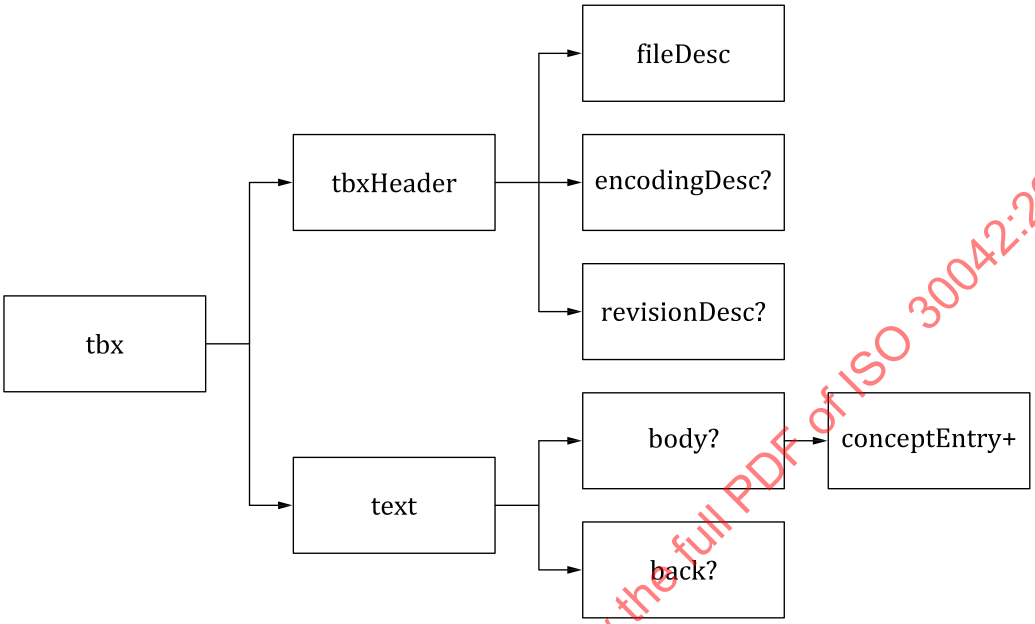
- A TDC shall contain exactly one GI (1..1), at most one CI (0..1) and any number of CE (0..*).
- A CE may contain any number of LS (0..*).
- An LS may contain any number of TS (0..*).

The highest-level XML element in a TBX document instance is the <tbx> element (which specifies the style and dialect in the attributes), which consists of a <tbxHeader> element and a <text> element (see Figure 2).

The <text> element in Figure 2 consists of concept entries, which are enclosed within one <body> element, and optional complementary information in the <back> element.

The <tbxHeader> element corresponds to global information in the TMF metamodel and consists of a description of the whole terminological data collection (in the <fileDesc> element) and a history of major revisions to the collection (in the <revisionDesc> element). The structure and content requirements of <tbxHeader> are found in Annex A.

Each concept entry in the <body> element documents all the terms and related information about a single concept. For this reason, the concept entry is enclosed in the core-structure element <conceptEntry>.



Key

?	= optional, but only one allowed
+	= at least one required
No sign	= shall appear exactly once

Figure 2 — TBX structure

8.3 Position of elements within a concept entry

8.3.1 Elements that may appear at multiple levels

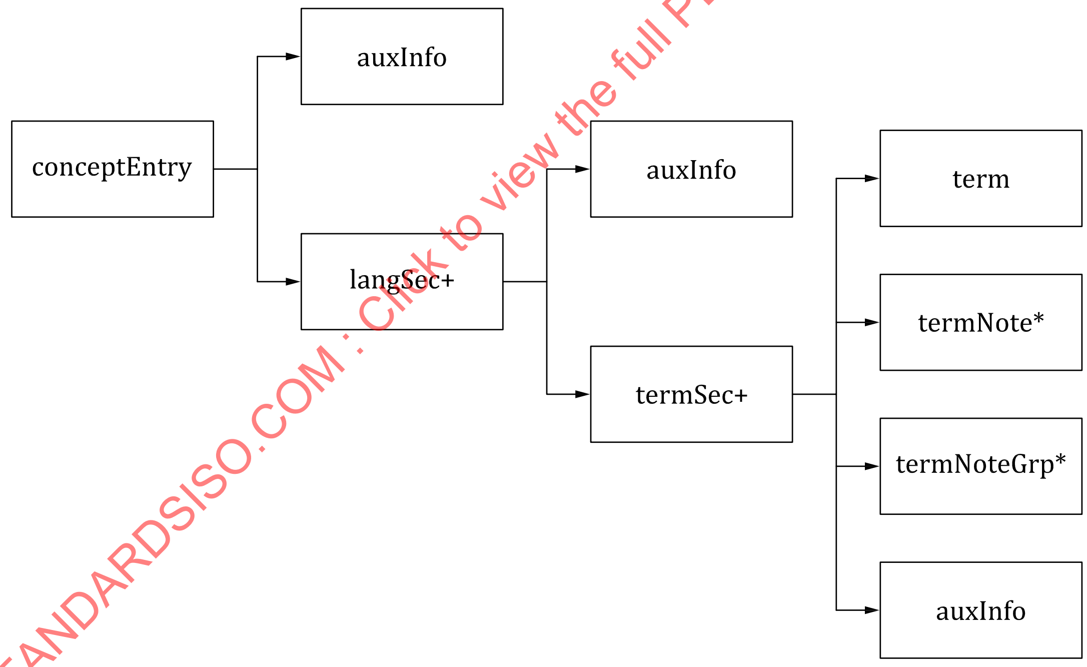
The core-structure elements described in Table 2 (represented as a group by auxInfo in Figure 3 and elaborated in Figure 4) may appear at any of the three levels of an entry: concept level (<conceptEntry>), language level (<langSec>), term level (<termSec>).

Table 2 — Elements that may appear at multiple levels

<date>	A date in a format in accordance with ISO 8601-1 and ISO 8601-2. Permissible date values shall comply with ISO 8601-1 and ISO 8601-2 and use the format yyyy-mm-dd. The date element is used in a <transacGrp> element.
<descrip>	Provides descriptive information about the node in question. The type of descriptive information is indicated by the data category, which is instantiated as the value of the *type* attribute. The data category may be restricted to certain levels of the entry, as indicated in the data category module. It may appear alone, or if additional information needs to be associated with the description, such as administrative information, it may be nested in a <descripGrp> element.
<descripGrp>	Contains one <descrip> element followed by zero or more <descripNote>, <admin>, <adminGrp>, <transacGrp>, <note>, <ref>, and <xref> elements.

Table 2 *(continued)*

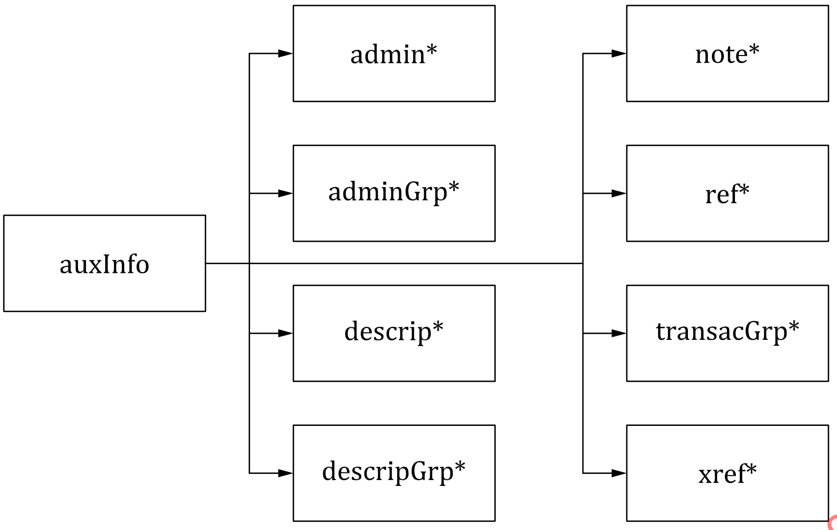
<admin>	Contains information of an administrative nature for the node in question, such as the source of information, or the project or client for which it applies. The type of administrative information is indicated by the data category, which is instantiated as the value of the *type* attribute. It may appear alone, or, if additional information needs to be provided, such as a note or a reference, it may be nested in an <adminGrp> element.
<adminGrp>	Contains one <admin> element followed by zero or more <adminNote>, <note>, <ref>, and <xref> elements.
<adminNote>	Contains some administrative information, such as the source of a note or other text.
<transacGrp>	Contains one <transac> element followed by zero or more <transacNote>, <date>, <note>, <ref>, and <xref> elements. It encloses information about a transaction, such as the date it was carried out, or the person who performed it. A date is specified by a <date> element, and the name of a person or entity responsible for the transaction is specified by a <transacNote> element.
<note>	Contains any kind of note.
<ref>	A cross-reference that points to another element within the <tbx> element.
<xref>	A cross-reference that points to an external object using a URI (a URL or other web address).



Key

+	= at least one required
*	= optional, one or more allowed
No sign	= shall appear exactly once

Figure 3 — Concept entry



Key

*	= optional, one or more allowed
No sign	= shall appear exactly once

Figure 4 — auxInfo

8.3.2 Elements that occur only at the term level

The term level in the entry hierarchy corresponds to the <termSec> in Figure 3. It contains one <term> element followed by zero or more <termNote> and/or <termNoteGrp> elements followed by any of the elements described in 8.3.1 and referred to by the auxInfo entity.

The elements that may occur only at the term level are shown in Table 3.

Table 3 — Elements that occur only at the term level

<term>	Contains one term.
<termNoteGrp>	A nesting element allowing one <termNote> followed by additional administrative or transactional information, <note>, or <xref> elements.
<termNote>	Provides different types of information about a term, depending on the value of the *type* attribute. For instance, it may contain grammatical information such as gender or part of speech, references to other related terms, administrative information such as the source of the term, or what project it is used for, and so forth.

The three levels of the core structure occur in nearly all terminological data collections. An additional level for term components is only used in terminological data collections that describe properties of parts of a term, such as the individual words in multi-word terms or the morphemes making up a single word term. This term component level, enclosed in <termCompSec>, is implemented through a dedicated module.

NOTE See Annex C and the TBX Info website[15] for support information on the term component module.

8.4 Typology of elements

8.4.1 Elements that play a classification or grouping role

Several key core-structure elements play a classification role to group data categories that have a similar purpose, the latter being instantiated as the value of the *type* attribute on the core-structure element. For instance, the <descrip> element primarily contains descriptions of a concept (such as its subject field), the <termNote> element describes properties of the term (such as its part of speech or the context in which it occurs), and the <admin> element contains various kinds of administrative information (such as sources and user names). The data categories that are permitted as type values for these elements are declared in the data category module. For instance, the element <descrip type="definition"> corresponds to the /definition/ data category, and the fact that the <descrip> element, which is declared in the core structure, allows /definition/ as a value of its *type* attribute, is declared in the data category module.

These TBX classification elements comprise the following:

- <admin>
- <adminNote>
- <descrip>
- <descripNote>
- <termNote>
- <transac>
- <transacNote>
- <ref>
- <xref>

There are some nesting elements that perform a grouping role, and they also involve two special uses of the *type* attribute on a core-structure element. The first case involves the element <refObjectSec>, which is a nesting element that contains a set of items (referenced objects). Its *type* attribute characterizes the child elements, i.e. the referenced objects (see the examples in 10.2, 10.3 and 10.6). For example, the *type* attribute value "bibl" indicates that the referenced objects are bibliographic. This is an example of the principle of inheritance, where the child elements inherit the type property of their parent.

The second case involves <termNoteGrp>, <transacGrp> and <descripGrp>, which are also nesting elements. These elements shall not have a *type* attribute. In these cases, the *type* attribute shall be indicated on the principal child element. See the example of a <transacGrp> in 10.7. This example demonstrates that the type of transaction which the <transacGrp> describes is indicated as the content of the child element <transac>. This is an example of the principle of reverse inheritance.

8.4.2 Elements that represent data categories

See Clause 5 for the list of elements from the core structure that represent data categories directly.

8.4.3 Inline markup elements

The inline markup elements are used to specify the language or function of textual strings or to allow TBX document instances to contain various kinds of other markup (such as HTML or text-processing markup) that need to be retained but do not need to be processed during terminology management functions. Inline markup elements are part of the core. TBX supports the following elements:

- <foreign>
- <hi>
- <sc>
- <ec>
- <ph>

The content models for <sc>, <ec>, and <ph> are adopted from ISO 21720 (XLIFF Version 2.0). The other elements are TBX-specific.

The elements <sc>, <ec>, and <ph> are designed to map to and from the corresponding elements in ISO 21720. TBX agents shall adhere to all relevant constraints and processing requirements for the corresponding elements and their attributes as specified in ISO 21720.

The attributes *id* and *startRef* are of data type xsd:NMTOKEN in ISO 21720 and function according to an XLIFF-specific fragment identification mechanism with XLIFF-specific uniqueness requirements. Their corresponding attributes in TBX are of data type xsd:ID and xsd:IDREF respectively and follow the standard XML fragment identification mechanism. The XLIFF attribute *dataRef* of data type xsd: NMTOKEN is replaced with the attribute *target* of data type xsd:IDREF in TBX. The attributes *disp*, *equiv*, *isolated*, *subType*, and *type* shall behave exactly as in XLIFF to facilitate the mapping.

The <foreign> element is used to mark a segment of text that is in a language different from that of the surrounding text. The optional *xml:lang* attribute may be used to identify the language of the text contained in this element, as shown in the following example.

EXAMPLE 1

```
<note>Kimchi is a Korean dish consisting of pickled, fermented
vegetables. The word derives from <foreign xml:lang="ko-KR"> □□
</foreign> in Korean.</note>.
```

The <hi> element delimits a section of text for various processing purposes, such as to mark a mathematical expression, or to mark an entailed term in a definition or another text field. The <hi> element is also allowed within <term> elements to mark portions of terms, such as a character in subscript or superscript, or a component within the term that is italicized or otherwise requires different formatting. The following example shows an entailed term in the definition of the term "canopy".

EXAMPLE 2

```
<descrip type="definition">the expanding, umbrella-like part of
a <hi type="entailedTerm" target="CID1234">parachute</hi></
descrip>.
```

The <sc> element is used to delimit the first of a paired sequence of original codes (for example, the opening part of the HTML <strong> tag). Each <sc> shall have a corresponding <ec> element following it within the same enclosing <descrip> element, unless it is marked as an orphaned code.

The <ec> element is used to represent the last of a paired sequence of original codes (for example, the closing </strong> tag). Each <ec> shall have a corresponding <sc> element preceding it within the same enclosing <descrip> element, unless it is marked as an orphaned code.

The following example shows some text in HTML followed by its TBX representation as a /context/.

EXAMPLE 3

```
<p>Cricket is a game played with a bat and ball on a large field,
known as a <i>ground</i>.</p>.

<descrip type="context">Cricket is a game played with a bat
and ball on a large field, known as a <sc id="i1" type="fmt"
subtype="xlf:i"/>ground<ec startRef="i1" type="fmt"
subtype="xlf:i"/>.</descrip>
```

The original data shall not be mixed with the linear inline content. In case the original data needs to be preserved for later processing or reuse, a *target* attribute should be used to reference the original data included in the TBX back matter. See 10.8 for the data model and usage description.

The following is an example of the original data structure that needs to be included or mapped (using a *target* attribute in the inline tag) in the back matter.

EXAMPLE 4

```
<originalData>
<data id="d1">&lt;i&gt;</data>
data id="d2">&lt;/i&gt;</data>
</originalData>
```

Unless the attribute-value pair *isolated="yes"* is used to mark an orphaned code, in <sc> the use of the *id* attribute is required and in <ec> the use of the *startRef* attribute is required, so that each end code is unambiguously linked to its start code and vice versa.

The *startRef* and *isolated="yes"* mechanisms provide TBX with support to mark up overlapping or otherwise non-well-formed ranges of codes, as in the following incorrect, nevertheless possible HTML construction:

EXAMPLE 5

```
This is <i>some <b>sample HTML</i> markup </b> with improper
nesting.
```

The above markup would be represented as follows using <sc> and <ec>:

EXAMPLE 6

```
This is <sc id="1" type="fmt" subtype="xlf:i"/>some <sc id="2"
type="fmt" subtype="xlf:b"/>sample HTML<ec startRef="1"
type="fmt" subtype="xlf:i"/> markup <ec startRef="2" type="fmt"
subtype="xlf:b"/> with improper nesting.
```

The following is an example of the original data structure that needs to be included or mapped in the back matter:

EXAMPLE 7

```
<originalData>
<data id="d1">&lt;i&gt;</data>
<data id="d2">&lt;b&gt;</data>
<data id="d3">&lt;/i&gt;</data>
<data id="d4">&lt;/b&gt;</data>
</originalData>
```

The values of the *id* attributes shall be unique within their common enclosing `<descrip>` element.

Finally, the `<ph>` element is used to represent a native standalone code or a sequence of such codes (such as an HTML `<br>` tag). It shall not be used to represent codes that occur in logical pairs or orphaned parts of paired codes:

EXAMPLE 8

```
<descrip type="context">Cricket is a game played with a bat and
ball on a large field, known as a ground.<ph id=”d1" type="fmt"
subtype="xlf:lb"/>Each phase of play is called an innings
during which one team bats, attempting to score as many runs as
possible.</descrip>.
```

The following is an example of the original data structure that needs to be included or mapped in the back matter:

EXAMPLE 9

```
<originalData>
<data id="d1">&lt;br/&gt;</data>
</originalData>
```

8.5 Attributes

8.5.1 type

In DCA style, the *type* attribute on a core-structure element specifies a data category. For instance, <descrip type="definition"> is the representation of the /definition/ data category.

8.5.2 xml:lang

The <tbx> root element shall have an *xml:lang* attribute. The value of the *xml:lang* attribute inherits downward through the TBX document instance until overridden by another *xml:lang* attribute. Thus, the language specified in the <tbx> element becomes the working language of the entire TBX document instance. Each <langSec> element shall also have an *xml:lang* attribute to indicate the language that this language section describes. This becomes the object language of the language section. Unless overridden by another *xml:lang* attribute on a child element, the content of all elements in a language section is assumed to be in the language specified by the *xml:lang* attribute of that <langSec> element. This does not apply to the content of elements that are picklist values. Thus, the content of a <descrip type="definition"> element at the concept entry level shall be in the working language of the TBX document instance unless otherwise specified, and a note in a language section shall be in the same language as the language section unless otherwise specified by an explicit *xml:lang* attribute.

The permissible values of the *xml:lang* attribute in TBX are identified in IETF BCP 47[13].

8.5.3 id and target

The *id* and *target* attributes work together to point unambiguously between elements in the same TBX document instance. For example, one entry:

```
<conceptEntry id="database-5574">
...(entry for "hunting dog")
</conceptEntry>
```

could be pointed to by another entry:

```
<conceptEntry id="database-5523">
<descrip type="superordinateConceptGeneric" target="database-5574">hunting
dog</descrip>
...(entry for "retriever" [a type of hunting dog])
</conceptEntry>
```

The content "hunting dog" in the second entry is for display purposes. It provides a visible label for the link to the target entry.

The *target* attribute shall point to an external object for an <xref>. The value of the *target* attribute of <xref> shall be an absolute external URL that uses either the HTTP or HTTPS protocol.

8.5.4 module

An optional *module* attribute is available for DCA style markup to indicate the module that a data category is selected from. This attribute facilitates parallelism with DCT style markup, the latter having the namespace mechanism to indicate the module for data categories. The value of the *module* attribute is the stated name of the module.

8.6 Types of text

The default text types in TBX for elements and attributes are unrestricted strings (plain text) in accordance with XSD:string.

Two additional mixed data entities are declared for TBX elements:

basicText: Apart from plain text, basicText may contain one or more <hi> elements to represent some highly restricted inline markup styles. basicText is primarily used by <term>. However, it is also used for the content of data categories expressing terms such as <descrip type="shortFormFor">, where this restriction is declared as a constraint in the data category module.

noteText: Apart from <hi>, which is also allowed in basicText, noteText allows <foreign>, <sc/>, <ec/>, and <ph/>. noteText is used by core-structure elements that express data categories with potential inline markup requirements, such as /definition/ or /usage note/. These data categories are expressed with the <descrip> or the <termNote> elements in the DCA style, and therefore these two elements allow noteText by default, but this may be further constrained in the data category module.

8.7 Character sets and encoding

TBX document instances shall be in Unicode, in accordance with ISO/IEC 10646, UTF-8, alternatively either UTF-16 or UTF-32.

9 Defining data category modules

9.1 General

This clause details the process of defining a new data category module, if existing modules are not suitable for the creation of a specific dialect. It describes requirements for data category modules that are intended to be used for public dialects. For modules intended for private use, the contents of this clause are recommendations.

The normative definition of a module shall be written in prose (an example of a module definition is provided in 7.3). Additionally, the TBX Module Description (TBXMD) formalism can be used to write a machine processable description of a module (see Annex C).

Defining a module is a four-step process:

1) name the module;

2) select one or more data categories;

3) define data category properties;

4) define data category constraints.

9.2 Naming the module

A module name shall be an NMTOKEN. The module name shall be unique among public modules. If the module is created specifically for a dialect, the module need not have the same name as that of the dialect.

Module names shall not contain the "." character, which is reserved for names of dialects that have been extended.

"Core" is the name reserved for the core structure defined in this document and shall not be used to name any other module.

9.3 Selecting data categories

Industry recognized data categories are available in DatCatInfo[10]. If the desired data category does not exist in Annex B or in DatCatInfo, a new one can be defined for the module.

TBX-compliant modules shall not introduce new data categories that duplicate or overlap the semantic intentions of the data categories declared in the TBX core structure (see Clause 5). If multiple TBX-compliant modules are used in a single dialect, they shall not have any data categories in common, because only non-overlapping modules are permitted to be used in the definition of a TBX dialect.

9.4 Defining data category properties

Each data category shall be associated with a classification element (see 8.4.1).

Additionally, the level or levels at which a data category can occur shall be indicated (conceptEntry, langSec, termSec).

Since data categories associated with the core-structure element <termNote> can only occur at the termSec level, the level for those data categories need not be indicated.

Finally, the PID (persistent identifier) of the data category shall be noted.

9.5 Defining data category constraints

The permissible content for each data category shall be defined: either a general datatype (string, basicText, noteText), or a list of values (picklist, such as "noun", "verb", "adjective", etc.).

If a data category in a module uses a picklist from the referenced data category repository (DatCatInfo or other compliant data category repository), it shall use the entire set or a subset of the picklist values. Special cases are discussed on DatCatInfo[10].

9.6 Using modules

Dialects can be configured to use modules in a kind of telescoping fashion, whereby the first module, TBX-Core, functions like the eyepiece on the telescope. The next broader module adds a few data categories, e.g., to produce a simple glossary, building on TBX-Core to form TBX-Min. A third module might add a few additional data categories to broaden the scope even further. Thus, in this example, TBX-Core includes the core, TBX-Min adds several more data categories, and finally TBX-Basic completes the sequence of dialects. A customized dialect might also add additional modules with data categories in addition to TBX-Basic.

NOTE More information about modules and dialects is available on the TBX Info website[15].

10 Referencing objects

10.1 General

In TBX there are two ways to reference objects (such as documents, graphics, websites, etc.). If the object is internal to the TBX document instance, it shall be embedded in the back matter of the TBX document instance and linked to by a <ref>. If the object is external to the TBX document instance, it shall be linked to by an <xref> element. In practice, most references are external using a URI/IRI.

NOTE Additional guidance on referencing objects is available on the TBX Info website[15].

Most termbases contain entries that need to reference objects that are external to the termbase itself, such as a graphics file, an audio file, or a website. These objects can be in the local environment (such as the computer file system) or external, such as a website.

To refer to objects that are external to the TBX document instance but yet are located in the local computing environment, the <xref> element shall specify the location of the target object using a path that is relative to the directory containing the TBX document instance. For exchange purposes, the locally referenced folder and its contents shall be provided to the recipient along with the TBX document instance.

To refer to a resource that is on the Internet, a persistent URI should be used.

If binary data is embedded in the back matter of a TBX document instance, it shall be encoded in Base64[9].

Various types of referenced objects can be defined for TBX dialects, such as bibliographic references and people, as shown in the examples in the following subclauses.

NOTE Any URLs used in the examples in Clause 10 are fictitious.

10.2 Referencing a file that is embedded in the back matter

A <descrip> element shall be used to reference a file that is embedded in the back matter of a TBX document instance. In this example, the concept entry contains a pointer to a graphic file that is embedded in the back matter in the form of binary data.

```
  <body>
  <conceptEntry id="database-5521">
  ...
    <descrip type="figure" target="fid-cricketbat">cricket bat</descrip>
  ...
  </conceptEntry>
  </body>
  <back>
    <refObjectSec type="binaryData">
      <refObject id="fid-cricketbat">
        <item type="codePage">base64</item>
        <item type="format">jpg</item>
        <item type="data">
/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0
a... </item>
      </refObject>
    </refObjectSec>
  </back>
```

10.3 Referencing a file from the back matter

This example shows how to record the location of the graphic in the back matter, rather than embedding the file in the back matter.

```
  <body>
  <conceptEntry id="database-5519">
  ...
   <descrip type="figure" target="fid-cricketbat">cricket bat</descrip>
  ...
  </conceptEntry>
  </body>
  <back>
    <refObjectSec type="binaryData">
      <refObject id="fid-cricketbat">
         <itemSet>
           <itemGrp>
             <item>bat.jpg</item>
             <xref target="http://www.bestsportsgraphics.com/cricket/bat.jpg"/>
           </itemGrp>
         </itemSet>
      </refObject>
    </refObjectSec>
  </back>
```

10.4 Referencing a file directly in the entry

This example demonstrates how to use an <xref> element to specify the location of an external resource directly in the entry, eliminating the need to include information in the back matter. If the external reference is stored locally, the path is relative to the location of the TBX document instance.

```
  <conceptEntry id="database-5518">...
    <xref type="xGraphic" target="
```

```
http://www.bestsportsgraphics.com/cricket/bat.jpg">cricket bat</xref>
...
</conceptEntry>
```

10.5 Referencing an external source

The following example adds a reference to a website from where the graphic was originally obtained.

```
<conceptEntry id="database-5529"> ...
   <xref type="xGraphic"
target="http://www.bestsportsgraphics.com/cricket/bat.jpg">cricket bat</xref>
   <xref type="xSource" target="http://www.bestsportsgraphics.com">
   Best Sports Graphics</xref>...
</conceptEntry>
```

This type of reference can also be used for definitions, contexts, notes, and so forth. This example shows a <descripGrp> element which groups a definition and its source.

```
<termSec>
    <term>over</term>
       <descripGrp>
          <descrip type="definition">six balls bowled in sequence by the same
          bowler</descrip>
          <xref type="xSource" target="http://www.allaboutcricket.com">
          All About Cricket</xref>
       </descripGrp>
</termSec>
```

10.6 Referencing and documenting a bibliographic source

The following example shows how to reference, in a concept entry, a bibliographic source that is documented in the back matter of a TBX document instance. Data categories for bibliographic references (values of the *type* attribute on the <item> elements) are adopted from the Dublin Core Metadata Element Set[11]. The example given below only includes a subset of the available data categories for demonstration purposes. The XML structure for bibliographic references is adopted from the one specified for “binaryData”.

```
<body>
    <conceptEntry id="database-5510">
       <langSec xml:lang="en-US">
            <termSec>
                 <term>over</term>
                 <descripGrp>
                 <descrip type="definition">six balls bowled by the same bowler
                 </descrip>
                 <admin type="sourceIdentifier" target="bid-cricket">The
                 Complete Guide to Cricket, p. 17</admin>
                 </descripGrp>
                 </termSec>
            </langSec>
       </conceptEntry>
    </body>
    <back>
       <refObjectSec type="bibl">
           <refObject id="bid-cricket">
                 <item type="creator">Farrel, Bart</item>
                 <item type="title">The Complete Guide to Cricket</item>
                 <item type="type">book</item>
                 <item type="identifier">978-3-16-148410-0</item>
                 <item type="publisher">Sports Publishers Inc.</item>
                 <item type="date">2000</item>
           </refObject>
       </refObjectSec>
    </back>
```

10.7 Referencing and documenting information about a person or organization

The following example shows how to reference information about a person and an organization who are responsible for part of a concept entry, which is documented in the back matter of a TBX document instance. Data categories for personal or organizational references (values of the *type* attribute on the <item> elements) are adopted from the vCard Standard[14]. The example only includes a subset of the available data categories for demonstration purposes.

```
<body>
    <conceptEntry id="database-5509">
       <langSec xml:lang="en-US">
       <termSec>
       <term>rabbit</term>
       <descripGrp>
           <descrip type="definition">In the game of cricket, a very poor
           batsman.</descrip>
           <transacGrp>
               <transac type="transactionType">importation</transac>
               <transacNote type="responsibility" target="rpid-harris">John
               Harris</transacNote>
           </transacGrp>
           <transacGrp>
               <transac type="transactionType">origination</transac>
               <transacNote type="responsibility" target="roid-CCA">Canadian
               Cricket Association</transacNote>
           </transacGrp>
       </descripGrp>
       </termSec>
   </langSec>
   </conceptEntry>
</body>
<back>
   <refObjectSec type="respPerson">
       <refObject id="rpid-harris">
           <item type="fn">John Harris</item>
       <item type="title">Senior terminologist</item>
       <item type="role">Approver</item>
       <item type="email">jharris@myserver.com</item>
       <item type="uid">xyz123</item>
   </refObject>
</refObjectSec>
<refObjectSec type="respOrg">
       <refObject id="roid-CCA">
           <item type="org">Canadian Cricket Association</item>
           <item type="tel">123-456-7899</item>
           <item type="email">cca@anotherserver.com</item>
           <item type="adr">75 1st Street, Ottawa, Ontario, Canada</item>
       </refObject>
</refObjectSec>
</back>
```

10.8 Referencing original data from noteText entities

The data model of the noteText entity shall be consistent with the XLIFF 2 inline data model[20]. Since XLIFF 2 does not allow mixing of original data with the inline tags and plain text (string), original data if needed should be referenced from the permissible inline tags to the back matter as follows.

```
<body>
    <conceptEntry id="database-5507">
       <langSec xml:lang="en-US">
         <termSec>
            <term>ground</term>
            <descrip type="context">Cricket is a game played with a
            bat and ball on a large field, known as a <sc id="i1" type="fmt"
            subtype="xlf:i" target="xlf1f1u1d1"/>ground<ec startRef="i1" type="fmt"
            subtype="xlf:i"/ target="xlf1f1u1d2">.</descrip>
         </termSec>
       </langSec>
    </conceptEntry>
</body>
<back>
    <refObjectSec type="originalData">
      <refObject id="xlf1f1u1d1">
       <item type="data">&lt;i&gt;</item>
      </refObject>
      <refObject id="xlf1f1u1d2">
        <item type="data">&lt;/i&gt;</item>
      </refObject>
    </refObjectSec>
</back>
```

Annex A
(normative)

Descriptions of the core-structure elements and attributes

A.1 Introduction

This annex contains a description of the TBX core structure, focusing on DCA style. Three types of XML constructs are described: entities, attribute classes, and elements. The following table describes the parts of the descriptions.

Members	Indicates which elements and attribute classes use (inherit) the attribute or attribute class described.
Attributes	Indicates the attributes and attribute classes that are permissible for the element or attribute class that is currently being defined. The attribute class declaration is followed by the names of the attributes in that class (indicated after the @ symbol, which is the Relax NG representation) in parentheses. If the attribute is unique to the element or attribute class that is currently defined, its declaration is provided directly. The attribute is either optional or required. NOTE Certain values of an element's *type* attribute can constrain the element's content to a picklist. These constraints are specified in the dialect's data category module.
Used by	Indicates which elements or element groups may contain (i.e. be the parent of) the element described.
May contain	Indicates which elements or data types may occur in the element described, i.e., its default content model. For instance, in the case of <descrip>, this row indicates that it may take the child elements <hi>, <sc>, <ec>, and so forth. However, the default content model may be further constrained by the value of the *type* attribute of the element. Any such additional constraints are indicated in the dialect's data category module. Elements are listed in the order they occur. Cardinality (counts of permitted contents) is indicated with the following key: stand alone element name – exactly one () – a group | – OR ? – Zero or One * – Zero or More + – One or More

Appropriate extension points for the value of @type for the classification elements shall be provided in a schema. By default, these extension points shall permit any text value. This default value shall be restricted to no values by schemas specific to the dialect being used if that dialect does not inherit any data categories for a given classification element (such as <admin>) from its stated modules, or expanded to include the data categories of the dialect which pertain to that classification element. Due to the value of the *type* attributes being used as generic identifiers (tags) in DCT style, a schema must provide appropriate extension points for these types of elements in all places where the classification elements are available for DCA. Data categories expressed in DCT style other than those described in Clause 8 (which are part of the Core) are imported from the namespace of the module from which they were inherited. Therefore, the recommended DCT extension points correspond to elements and attributes from a namespace external to that of the Core, and are associated with DCA classification elements (such as <descrip> and <termNote>).

This annex provides the details necessary to facilitate the creation of a valid Core schema. Any constraints given in Clause 8 apply to this annex, even if not explicitly restated in this annex. It is therefore required that Clause 8 be thoroughly studied before creating a schema from the information in this annex.

A.2 Entities

entity.auxInfo	The auxInfo entity corresponds to a set of elements that can occur at any one of three levels: the concept level (<conceptEntry>), the language level (<langSec>), and the term level (<termSec>). It includes the elements <descrip>, <descripGrp>, <admin>, <adminGrp>, <transacGrp>, <note>, <ref>, and <xref> (see 8.3.1). (admin | adminGrp | descrip | descripGrp | note | ref | transacGrp | xref)*
entity.basicText	basicText is used for terms and term-like elements. It allows string plus <hi> for some limited embedded markup (see 8.6). (string | hi)*
entity.noteText	noteText allows more inline markup than basicText. It is often suitable for data categories that normally contain text, such as definitions, notes, and contextual examples (see 8.6). (string | sc | ec | foreign | hi | ph)*
entity.noteLinkInfo	This entity groups elements for administrative information, notes, and links. It includes the elements <admin>, <adminGrp>, <transacGrp>, <note>, <ref>, and <xref>. It is used by <itemGrp>. (admin | adminGrp | transacGrp | note | ref | xref)*

A.3 Attribute classes

id	The id attribute class indicates a unique identifier of an element. Element identifiers can include entry identifiers (e.g. eid-...), concept identifiers (e.g. cid-...), term identifiers (e.g. tid-...), among others. This attribute is optional.
Members	IDLangTgtDtyp [admin adminNote descrip descripNote ref termNote transac transacNote] idlang [change note revisionDesc sourceDesc title titleStmt] idtype [item itemSet] adminGrp back body date descripGrp encodingDesc fileDesc foreign itemGrp tbxHeader p publicationStmt refObject refObjectSec term termNoteGrp text termSec transacGrp xref
Attributes	@id - (optional)
Datatype	xsd:ID

idlang	The idlang attribute class combines the *id* attribute and the *xml:lang* attribute. Both are optional for elements that use this attribute class.
Members	change note revisionDesc sourceDesc title titleStmt
Attributes	id (@id) lang (@xml:lang)
Datatype	string

IDLangTgtDtyp	The IDLangTgtDtyp attribute class groups together the attributes *id*, *xml:lang*, *target* and *datatype.*
Members	admin adminNote descrip descripNote ref termNote transac transacNote
Attributes	lang (@xml:lang) id (@id) target-IDREF (@target) @datatype - (optional)
Datatype	string

idtype	The idtype attribute class combines the *id* attribute and the *type* attribute. Both are optional for elements that use this attribute class.
Members	item itemSet
Attributes	id (@id) @type - (optional) Indicates data category through the value of this *type* attribute
Datatype	string

lang	The lang attribute class corresponds to the attribute *xml:lang*.
Members	IDLangTgtDtyp [admin adminNote descrip descripNote ref termNote transac transacNote] idlang [change note revisionDesc sourceDesc title titleStmt] foreign hi p
Attributes	@xml:lang - (optional) Indicates the language of the element content. The allowed values are found in IETF BCP 47. This attribute is optional for elements that use this attribute class. For a description, see the information at the <tbx> element.
Datatype	xsd:language

target-IDREF	The target-IDREF attribute class corresponds to the *target* attribute, which is a pointer to another element. The link is established when the value of the *target* attribute of the source element is the same as the value of the *xml:id* attribute of the targeted element. When this attribute is used with <descrip>, the target is often another term, such as in the tag <descrip type="abbreviatedFormFor" target="tid-1234">. It is also available for the data category /transferComment/ to indicate the term being commented about. The pointer is a unique identifier. The content of elements that take this attribute is typically the display value of the targeted element, such as the targeted term. The elements that take this attribute cannot be empty; display content is mandatory.
Members	IDLangTgtDtyp [admin adminNote descrip descripNote ref termNote transac transacNote] hi
Attributes	@target - (optional)
Datatype	xsd:IDREF

A.4 Elements

NOTE There is an optional attribute @module on classification elements (DCA style). The attributes @disp, @equiv, and @isolated can be used in mapping inline markup to ISO 21720 (XLIFF)[20].

<admin>	Contains information of an administrative nature for the node (parent element) in question, such as the source of information, or the project or client for which it applies. The type of administrative information is indicated by the value of the *type* attribute. It can appear alone, or, if additional information needs to be provided, such as a note or a reference, it can be nested in an <adminGrp> element. NOTE This element can occur at any of the three levels of the concept entry. Several types, however, logically should occur at restricted levels. For instance, /conceptOrigin/ would typically occur at the conceptEntry level. These level constraints are declared in the dialect's data category module.
Attributes	IDLangTgtDtyp @type - (required) Indicates the data category type of the <admin> element, such as /product subset/ or /originating person/.
Used by	adminGrp descripGrp termNoteGrp entity.auxInfo entity.noteLinkInfo
May contain	entity.noteText

<adminGrp>	An element that contains a group of elements that contain administrative information.
Attributes	id (@id)
Used by	descripGrp termNoteGrp entity.auxInfo entity.noteLinkInfo
May contain	admin (adminNote | note | ref | xref)*

<adminNote>	A note about administrative information.
Attributes	IDLangTgtDtyp @type - (required)
Used by	adminGrp
May contain	string

<back>	An element that contains information supplementary to that described in concept entries and shared across the terminological data collection.
Attributes	id (@id)
Used by	text
May contain	refObjectSec*

<body>	An element that contains the concept entries of the terminological data collection.
Attributes	id (@id)
Used by	text
May contain	conceptEntry+

<change>	Information about one particular change event in the revision history of a TBX document instance.
Attributes	idlang
Used by	revisionDesc
May contain	p+

<conceptEntry>	An element that contains information about a concept. It shall contain at least one language section.
Attributes	@id – (required)
Used by	body
May contain	entity.auxInfo langSec+

<date>	A date in a format in accordance with ISO 8601-1 and ISO 8601-2. Permissible date values shall use the format yyyy-mm-dd. Preferably as xsd datatype. The date element is used in a <transacGrp> element.
Attributes	id (@id)
Used by	transacGrp
May contain	A date in a format in accordance with ISO 8601-1 and ISO 8601-2.

<descrip>	An element that contains descriptive information about a concept or sometimes about a term. The type of information that the element contains, and any restrictions on the permissible values of the element, are determined by the value of the *type* attribute. These constraints are specified in the dialect's data category module.
Attributes	IDLangTgtDtyp @type - (required) Indicates the data category specified by the <descrip> element, such as /definition/ or /associated-concept/.
Used by	descripGrp entity.auxInfo
May contain	entity.noteText

<descripGrp>	An element that contains one <descrip> element as well as additional child elements for associated administrative information.
Attributes	id (@id)
Used by	entity.auxInfo
May contain	descrip (admin | adminGrp | descripNote | note | ref | transacGrp | xref)*

<descripNote>	Element used to indicate the type of definition or context with which it is associated.
Attributes	IDLangTgtDtyp @type - (required) Indicates the data category specified by the <descripNote> element, such as /contextType/.
Used by	descripGrp
May contain	string

<ec>	The <ec> tag encloses the ending tag of a tag pair.
Attributes	@startRef - required if <ec> is in the same note text as its corresponding <sc> @isolated – required with value of “yes” if <ec> is not in the same note text as its corresponding <sc>. Otherwise, @isolated is optional, but must be “no” if used. @disp – (optional) @equiv – (optional) @id - (optional) @type - (optional) permitted values: fmt, ui, quote, link, image, other@subtype - (optional) @target - (optional)
Used by	entity.noteText
May contain	string

<encodingDesc>	Includes a <p> element, for including any information about encoding.
Attributes	id (@id)
Used by	tbxHeader
May contain	p+

<fileDesc>	An element containing child elements that describe the TBX document instance.
Attributes	id (@id)
Used by	tbxHeader
May contain	publicationStmt? titleStmt? sourceDesc+

<foreign>	Used to mark a segment of text that is in a language different from that of the surrounding text.
Attributes	idlang
Used by	entity.noteText
May contain	entity.noteText

<hi>	Highlights a segment of text and optionally points to another element.
Attributes	target-IDREF @type - (optional) values: entailedTerm, hotkey, italics, bold, superscript, subscript, math
Used by	entity.basicText entity.noteText
May contain	string

<item>	One descriptor of an object in the back matter of a TBX document instance.
Attributes	idtype
Used by	itemGrp itemSet refObject
May contain	entity.noteText

<itemGrp>	Contains one <item> element as well as additional child elements for associated administrative information.
Attributes	id (@id)
Used by	itemSet refObject
May contain	item entity.noteLinkInfo

<itemSet>	An element that contains a set of <item> or <itemGrp> elements.
Attributes	idtype
Used by	refObject
May contain	(item | itemGrp)+

<langSec>	An element that contains all the information in a concept entry pertaining to one language, including all the <termSec> elements (terms and associated information) for that language.
Attributes	@xml:lang - (required) Indicates the language of the language section. See also the description on the <tbx> element.
Used by	conceptEntry
May contain	entity.auxInfo termSec+

<note>	Any kind of note.
Attributes	idlang
Used by	adminGrp descripGrp titleStmt transacGrp entity.auxInfo entity.noteLinkInfo
May contain	entity.noteText

<p>	Paragraph text that is contained by the <tbxHeader>, for including various descriptions of the TBX document instance.
Attributes	idlang @type - (optional).
Used by	change encodingDesc publicationStmt sourceDesc
May contain	entity.noteText

<ph>	Placeholder - Used to encapsulate markup codes that do not occur in logical pairs with other codes. Examples include the XHTML <br/> and <hr/> tags. This element should not be used to encapsulate codes that occur in pairs (such as the XHTML <em> and </em> tags), which should instead be encapsulated with <sc> and <ec>.
Attributes	@type - (optional)
Used by	entity.noteText
May contain	string

<publicationStmt>	Any information pertaining to the publication of the TBX document instance.
Attributes	id (@id)
Used by	fileDesc
May contain	p+

<ref>	A cross-reference that points to another element within the <tbx> element of the TBX document instance.
Attributes	IDLangTgtDtyp @type - (required)
Used by	adminGrp descripGrp transacGrp entity.auxInfo entity.noteLinkInfo
May contain	string

<refObject>	An element in the back matter of a TBX document instance that contains a series of items that describe one referenced object.
Attributes	id (@id)
Used by	refObjectSec
May contain	(item | itemGrp | itemSet)+

<refObjectSec>	A grouping element that contains a list of objects that are referenced in the back matter of a TBX document instance.
Attributes	id (@id) @type - (required)
Used by	back
May contain	refObject+

<revisionDesc>	Information about any revisions that have been carried out on the content of the TBX document instance or on the source database from which it is derived.
Attributes	idlang
Used by	tbxHeader
May contain	change+

<sc>	The <sc> tag encloses the beginning tag of an original code tag pair.
Attributes	@id - (required) @isolated – Required with value “yes” if <sc> is not in the same note text as its corresponding <ec>. Otherwise @isolated is optional but must be “no” if used. @type - (optional) @subtype - (optional) @target - (optional)
Used by	entity.noteText
May contain	string

<sourceDesc>	Any information about the source of the TBX document instance.
Attributes	idlang
Used by	fileDesc
May contain	p+

<tbx>	The root element of a TBX document instance.
Attributes	@xml:lang - (required) Indicates the default language of the entire <tbx> element content. The permissible values of the *xml:lang* attribute are specified by IETF BCP 47[13]. @type - (required) Indicates the name of the TBX dialect, in the format: TBX-[name]. @style - Indicates the xml style (DCA or DCT). For instance: <tbx type="TBX-basic" style="dca" xml:lang="en">
Used by	n/a
May contain	tbxHeader text

<tbxHeader>	A grouping element that contains child elements which describe the TBX document instance.
Attributes	id (@id)
Used by	tbx
May contain	fileDesc encodingDesc? revisionDesc?

<term>	The term that is being described in a <termSec> element. This element may contain a <hi> element to allow a limited amount of inline markup. This is intended to handle markup requirements in special cases such as for terms that represent scientific concepts. However, it is recommended to use inline markup only when necessary to represent the term in its base form. The <hi> element shall not be used for presentational styles chosen for aesthetic purposes.
Attributes	id (@id)
Used by	termSec
May contain	entity.basicText

<termNote>	An element that contains descriptive information about a term. The type of information that the element contains, and any restrictions on the permissible values of the element, are determined by the value of the *type* attribute. These constraints are specified in the dialect's data category module.
Attributes	IDLangTgtDtyp @type - (required) Indicates the data category specified by the <termNote> element, such as /part of speech/ or /term type/.
Used by	termNoteGrp termSec
May contain	entity.noteText

<termNoteGrp>	A grouping element that contains one <termNote> element, and auxiliary information such as administrative information, transaction information, notes and cross-references. It does not allow any <descrip> elements. It may only appear at the term (termSec) level and below.
Attributes	id (@id)
Used by	termSec
May contain	termNote entity.noteLinkInfo

<termSec>	A grouping element that contains child elements describing a term.
Attributes	id (@id)
Used by	langSec
May contain	term (termNote | termNoteGrp)* entity.auxInfo