Creating a MultiTerm XML to TBX Mapping File in JSON

Introduction

Because MultiTerm allows user-created data fields, and because TBX is not a single markup language but a family of them, the conversion tool must be configured for the exact kinds of data it will be converting -- their original appearance and the TBX to which they should be translated. We configure this with a file called a 'mapping'.

The mapping is in JSON, a simple textual format for data structures. The full syntax for JSON is diagrammed on http://www.json.org/ , but we will also exemplify the parts we need in this document. In these examples, terms in parentheses represent parts of the mapping that are defined elsewhere, not literal text.

To create your own mapping file, you can follow along with this document, copy and adapt the examples, and replace the parenthetical placeholders with correct JSON as you learn. You may also find it helpful to open one or more sample mappings from the MultiTerm Convert website[1], for additional examples.

The Mapping File Description

At the highest level, the mapping is a JSON array like this one, with five members:

```
[
        "TBX-Example",
        "example.xcs",
        (categorial mapping),
        (queue-draining orders),
        {}
]
```

The first and second elements of the array are strings. The first is the **TBX dialect element**, the name of the TBX dialect that the mapping is intended to produce. The second is the **XCS element**, the filename of the XCS file that defines that dialect. If your mapping is confined to the TBX default data categories, the dialect name can be simply "TBX-Default" and the XCS file can be TBXXCS.xcs (which you can download here[2]). If these categories cannot accommodate your data, you will have to define your own categories, and consequently your own TBX dialect; see the TBX standard for information on this.

The third element, the **categorical mapping**, details the correspondence between data fields defined in MultiTerm and elements of TBX. It is the heart of the mapping, and we will describe it shortly. The fourth element, the **queue-draining orders,** helps to adjust relations among data fields in the same entry, a complex subject we will return to later.

Finally, the fifth element is an empty JSON object. If we need to extend the mapping with additional information in the future, this is where it will be stored.

First Element: TBX Dialect

There are 2 official TBX dialects which can be used here: “TBX-Default” and “TBX-Basic”. However, some mappings may not always map easily to “TBX-Basic”, so for MultiTerm Conversions, it is perhaps best to rely on “TBX-Default”. For a list of all available data categories of “TBX-Default”, please see the TBX Standard[3].

Please consult the TBX Standard for information on how to create a TBX dialect, if you cannot make TBX-Default work. It is preferred that an official dialect is used whenever possible.

[1] http://www.tbxinfo.net/scripts/cgi-bin/mt2tbx.cgi
[2] http://www.ttt.org/oscarstandards/tbx/tbxxcs.xcs
[3] http://www.tbxinfo.net/wp-content/uploads/2016/10/tbx_oscar.pdf

Second Element: XCS

The standard XCS file for “TBX-Default” is “TBXXCS.xcs” and can be found here.[4] The XCS file for TBX-Basic is “TBXBasicXCSV02.xcs” and can be found in the TBX-Basic Package[5].

If you used an unofficial TBX dialect, you must create and provide your own XCS file.

Third Element: Categorical Mapping

Now for the categorical mapping in detail. It is also a JSON object, but not an empty one. It contains up to three name-value pairs, corresponding to the concept, language, and term levels of a termbase:

```
{
        "concept" : (one-level mapping),
        "language" : (one-level mapping),
        "term" : (one-level mapping)
}
```

If you are writing a mapping as you follow along with this document, you should insert this to replace the placeholder above, so that your overall structure looks like this (indented for readability):

```
[
        "TBX-Example",
        "example.xcs",
        {
                "concept" : (one-level mapping),
                "language" : (one-level mapping),
                "term" : (one-level mapping)
        },
        (queue-draining orders),
        {}
]
```

Note that the parentheses are part of the placeholder, and not part of the JSON syntax; they go away when the actual code is inserted. To save space, we will not exemplify every substitution in this document.

The One-Level Mapping

A **one-level mapping**, as its name suggests, describes the mapping from MultiTerm data fields to TBX elements on one structural level. It is also a JSON object, containing a list of name-value pairs -- one pair for every field that can be entered on that level in MultiTerm. For example, in MultiTerm's Standard

[4] http://www.ttt.org/oscarstandards/tbx/tbxxcs.xcs
[5] https://github.com/byutrg/TBX-Basic-Package/archive/master.zip

Bilingual termbase template, the language level only allows two fields: 'Definition' and 'Note'. Therefore, the one-level mapping for "language" would contain only two pairs:

```
{
        "Definition" : (template-set),
        "Note" : (template-set)
}
```

However, many fields are used on some level of your MultiTerm termbase, that is how many name-value pairs your one-level mapping should contain. Just be sure to place a comma after each pair but the last.

The Template-Set

The **template-set** contains one or more templates for creating TBX elements. Most often a field in MultiTerm always corresponds to the same kind of element in TBX. In this case the template-set is very simple. Here is one for definitions:

```
[[[
        "@auxInfo",
        "<descrip type='definition'/>",
        "null",
        "content"
]]]
```

The three square brackets here are no special syntax; they just represent three JSON arrays, nested one inside the other. The innermost array is of a special kind we call a **'teasp'**, so the template-set we just showed you could have been described as

[[(teasp)]]

The TEASP

The teasp is an exact recipe for one new TBX element, which makes it the most important part of the mapping. We'll explain just how it works now, and return afterward to the formal definition of template-sets and the reason for the extra nested arrays.

'Teasp' is in fact an acronym, standing for "Target, Element/Attributes, Substitution, Placement," and this describes the contents of a teasp array:

```
[
        (target),
        (element/attributes),
        (substitution),
        (placement)
]
```

The **target** is a string. It tells where the new element will fit in the TBX entry that contains it. Most often this will be auxInfo, as in our example. In one-level mappings for the term level, you may also define termNote elements, in which case the target should be termNote. There are other kinds of targets, which we will discuss later. The target string must begin with an @ sign. For more specific information on target, see Categorical Mapping: Using TEASP Target and 'Bundling' later in this document.

The **element** to be used, and any **attributes** it should have, are specified by a string. It should contain literal XML, except that JSON requires inserting a backslash before any double quote mark within the string. Our example takes advantage of the fact that XML allows either single or double quotes around attribute values; you may wish to do the same.

The XML may be an empty-element tag, as in our example (signified by the slash just before the closing angle bracket), or a pair of tags with content between.

You should choose an element consistent with the usage of the TBX specification. The most relevant elements are admin (for administrative information), descrip (for descriptive information not covered by another element type), note (for notes), termNote (for properties of a term or term component), ref (for internal cross-references), and xref (for external URIs). If your termbase includes term component information, you will certainly use termComp elements, and may also use termCompList (depending on your exact termbase structure).

If your termbase includes parent fields and subordinate fields, the elements mentioned above are always suitable for parent fields. Some of them may be disallowed for subordinate fields, depending on the element to which the parent field was mapped. Fortunately, our program knows these restrictions, and if your data map out to an illegal combination it will turn the subordinate element into an XML comment and issue a warning.

(You may also read the restrictions in the TBX specification, Annex D.4. If the parent field maps to an admin or note, see the "May contain" list under adminGrp (including adminNote). For a descrip, ref, or xref, see under descripGrp (including descripNote). For a termNote, under termNoteGrp; for a termComp, termCompGrp; and for a termCompList, termCompList itself.)

If data values in the MultiTerm termbase are not the same as those standardized for TBX, the **substitution** tells how to replace them. It is mostly applicable to data categories that take a small, limited set of values, such as grammatical gender. In our example, the definition should be kept just as it is; hence there is nothing to substitute and we simply use the string "null". Other substitutions will be explained momentarily.

Finally, the **placement** is a string that tells where the user data fits into the new TBX element. The usual placement, used in our example, is "content", meaning that the text from MultiTerm (or resulting from a substitution) will be a child of the new element. Other placements will also be discussed momentarily.

At this point you can probably write teasps for at least some of TBX elements that you need to generate. If so, go ahead and do so. If some field from your termbase requires a substitution or some other placement, the next few paragraphs should help you to fill it in.

TEASP: Substitutions

To begin substitutions, here is a sample teasp for a hypothetical Gender field:

```
[
		"@termNote",
		"<termNote type='grammaticalGender' />",
		{
				"fem" : "feminine",
				"masc" : "masculine",
				"neut" : "neuter"
		},
		"content"
]
```

Notice that the substitution part of this teasp is no longer the single string "null" but a whole JSON object. It contains three name/value pairs: On the left side are the data values we expect to receive from MultiTerm, and on the right are the corresponding standardized values to be used in TBX (both as strings). Similar objects can be written for any finite set of values. (Values not found on the left side will be passed through unchanged, useful if MultiTerm happens to contain TBX-standard values.)

Three special substitutions are defined for rare needs. These are like the null substitution in that they are requested with a string, rather than an object. The "camel case" substitution removes spaces from any value and marks the start of a new word with a capital letter instead. The "category tag" substitution attaches the name of the MultiTerm data field to the data. The "lowercase" substitution decapitalizes the data. Future versions of this program may have additional special substitutions if required.

TEASP: Placement

There are four allowed values for placement:

1. target
2. content
3. null
4. type

The next most common placement is "target", used if the value is a cross-reference (either to another entry or to a URL). In these cases, the correct TBX element is a ref or xref (in the MultiTerm XML file), elements that use an attribute named "target" to carry such information. (Do not confuse this attribute or placement with the teasp's target, i.e. the place for the new element in the syntax of TBX. We regret the overlapping names.) You do not need to write the target attribute into your teasp; it will be created by the placement. However, you should write generic content in the element, such as "external graphic" or "see".

At times, you may not need to place content into a TBX element at all, because you can write the element entirely in advance. For this there is a "null" placement. Finally, there is a "type" placement, similar to "target" but used to set the element's type attribute. We expect it to be useful only rarely.

Using Multiple TEASPs

The information above allows you to write a teasp for any TBX data category. However, you may find that a field in MultiTerm corresponds to more than one TBX data category. For example, an "audience" field might mix customer names with internal project labels, or a "grammar" field might record facts about both gender and number. This requires us to revisit the definition of a template-set.

A template-set associates one MultiTerm field with one or more teasps. We have seen a template-set with only one teasp:

[[(teasp)]]

This is a default teasp, meaning that it applies to any data value that might be found in the field. But a template-set can also contain special teasps, which apply only to particular data values. For the "audience" field, we could create a special teasp for the internal project labels, and let the default teasp handle the customer names. The template-set would look like this:

```
[
        [
                [
                        "@auxInfo",
                        "<admin type='customerSubset' />",
                        "null",
                        "content"
                ],
                ["project1", "project2", "project3"]
        ],
        [
                "@auxInfo",
                "<admin type='projectSubset' />",
                "null",
                "content"
        ]
]
```

You will recognize the teasps in the structure. Abstracting them away makes it more tractable:

```
[
        [
                (default teasp: customerSubset),
                ["project1", "project2", "project3"]
        ],
        (special teasp: projectSubset)
]
```

The remaining component just shows which values the special teasp (“projectSubset”) applies to (“project1”, “project2”, and “project3”). These values are a **value-group.** So, let us simplify the example once more:

```
[
		[
				(default teasp: customerSubset),
				(value-group 1: projectSubset)
		],
		(special teasp 1: projectSubset)
]
```

Template-Set Overview

And now we are ready for formal definitions. A template-set is a JSON array with one or more elements. The first element is a 'key list' and any others are special teasps:

```
[
		(key list),
		(teasp),
		(teasp)
]
```

A key list is also an array, with the same number of elements as its template-set. The first element is the default teasp, and any others are value-groups:

```
[
		(teasp),
		(value-group),
		(value-group)
]
```

Finally, a value-group is just an array of strings. The first value-group gives the data values that use the first special teasp, the second value-group gives the ones for the second special teasp, and so on.

Let's now sketch a template-set for the hypothetical "grammar" field we mentioned earlier. For "audience" we made a value-group for project labels, but we let customer names fall to the default teasp. This time, we'll make value-groups for both gender and number. Putting them into the key list, we might have:

```
[
		(teasp: default),
		["fem", "masc", "neut"],
		["mass", "pl", "sg"],
]
```

And for the entire template-set we'd have:

```
[
		[
				(teasp: default),
                ["fem", "masc", "neut"],
                ["mass", "pl", "sg"],
        ],
        (teasp: gender),
        (teasp: number)
]
```

The special teasp for gender was given in an earlier example, and the one for number can be adapted from it. But what shall we do about the default teasp?

If the default teasp is ever used, it's because the "grammar" field contained a value we weren't prepared to deal with. We can use the program to bring this to our attention, by writing a default teasp like this:

```
[
        "@unhandled",
        "<unknown type='grammar' />",
        "null",
        "content"
]
```

The <unknown> element is not valid TBX, so if we run the output file through a validator we will get an error message pinpointing the trouble. It will show the original field name and data value, so we can decide what to do about them. In this teasp we also see the "@unhandled" target. Elements sent to this target will wind up at the end of their entry, where they are easy to spot, wrapped conspicuously in <unhandled> tags. Then we can either edit them out, if they are not necessary in the TBX output, or refine the mapping to cover them.

Template-sets have one more important property: One value can trigger more than one special teasp. For example, suppose our "grammar" field also has values such as "fem.sg" and "masc.pl". (This is unnecessary, since MultiTerm can store both "fem" and "sg" in one entry's grammar field. But users have been known to do it anyway.) We can write one value-group for all values with gender, and another for all values with number:

```
["fem", "fem.sg", "masc", "masc.pl", "neut"],
["mass", "pl", "masc.pl", "sg", "fem.sg"]
```

Then, in each teasp we let the substitution get rid of the irrelevant extra information; in the gender teasp the substitution will be

```
{
        "fem" : "feminine",
        "fem.sg" : "feminine",
        "masc" : "masculine",
        "masc.pl" : "masculine",
        "neut" : "neuter"
}
```

and in the number teasp:

```
{
		"pl" : "plural",
		"masc.pl" : "plural",
		"sg" : "singular",
		"fem.sg" : "singular"
}
```

(we need no substitution to change "mass" since it is already correct for TBX).

When the program encounters one of the combined values, it finds it in both value-groups and so it lets both special teasps generate a TBX element. However, the substitutions ensure that each element only contains one kind of information, thus separating the data categories that were mixed in MultiTerm.

Categorical Mapping: Using TEASP Target and 'Bundling'

We will now end our discussion of the categorical mapping by revisiting the first part of the teasp, namely the target. We have seen the targets "@auxInfo", "@termNote", and "@unhandled" already. Each of these collects elements for a specific slot in the TBX syntax. We will review these and introduce two others like them. You may also define other targets. These do not feed directly into TBX; they are holding areas for elements that need to undergo special additional processing, particularly a kind of processing called 'bundling'. We will introduce this process, which will lead us to the remaining major component of the mapping, the queue-draining orders.

The predefined targets are "@auxInfo", "@termNote", "@termComp", "@termCompList", and "@unhandled". "@_content" serves to place the new element directly as a child of the element that contains it. We do not foresee any practical need for this, but be warned and do not use the name by mistake. "@auxInfo" is the target for almost all information about a term, language, or concept entry. "@termNote" is an exception, used in teasps that create a termNote element, because TBX syntax does not permit mixing these with other auxInfo. "@termComp" similarly collects termComp elements, if an enclosing data field will generate the termCompList for them (which is targeted to "@termCompList"). (If the term component data fields are direct children of the term entry, they should not be targeted to "@termComp", but to a target such as "@syllabification" or "@morphologicalElement", which names the TBX data category to which they belong.) Finally, "@unhandled" as we have seen gives us a way to direct attention to incorrect input for manual editing.

User-defined targets are of two kinds. As we have seen, at the term level they can collect term components, which the converter will format into proper termCompList elements. But they are also used to queue up elements for 'bundling'.

Bundling is a way to restructure the data in a termbase. Often a termbase is defined with fields such as "Definition" and "Definition_Source", each a direct child of an entry. But the source citation isn't really an independent fact about the entry; it is a dependent fact about the definition. It would be better if the source field were subordinate to the definition field, and best practice demands that it be so subordinated in TBX. Bundling is a way to create this relationship of subordination.

To bundle the source into the definition, we must avoid dropping them directly into auxInfo, instead targeting them to a special queue. Almost any word can be the name of a queue, but it is perhaps simplest to name them after the original data field: "@Definition" and "@Definition_Source". (Names that should not be used, other than those listed above, are "@_content", "@id", "@xml:lang", "@type", "@username", and "@date". Targeting elements to these locations may corrupt the TBX output.)

Having created the queues in the teasp, we would then write an order to drain them in the queue-bundling orders.

Queue-bundling orders are contained in a JSON object with any of these three keys:

```
{
        conceptGrp: (list of orders),
        languageGrp: (list of orders),
        termGrp: (list of orders)
}
```

Each key indicates a level of the termbase; each level has its own **list of orders**.

The list of orders is an array. It may be empty:

```
[]
```

or contain one or more orders, separated by commas as usual:

```
[
        (order),
        (order),
        (order)
]
```

An **order** is also an array, containing exactly three strings. The first is the queue where the main element is held; the second is the queue where the subordinate element is held; and the third is the queue where the combination should be placed. Here, these are written without the '@'-sign. So the order for the definitions example would read

```
["Definition", "Definition_Source", "auxInfo"]
```

and, presuming that definitions are at the language level, the whole queue-draining orders section would be

```
{
        "languageGrp" : [
                ["Definition", "Definition_Source", "auxInfo"]
        ]
}
```

TBX often requires grouping elements, such as descripGrp, to wrap around ordinary data categories to associate them with their subordinates. The converter takes care of this detail for you, so your teasp can read the same whether it will have a subordinate or not. (TBX experts: The converter also alters the <note> element if necessary to conform to the behavior given in the spec.)

Orders are executed in the sequence listed, so later orders can drain queues filled by earlier ones if necessary.

No orders should be given to drain auxInfo, _content, or any of the other queues mentioned above as unsuitable names.

Finally, queues that are targeted but not drained are disposed of as follows: At the term level, as we have seen, they are converted into lists of term components. At the language and concept levels, or if they arise from subordinate fields, they will be wrapped in tags named after the queue and pushed into our old friend "@unhandled", again to draw user attention to a flaw in the mapping.

Appendix I: Sample Mapping File 1

```
[
        "TBX-Default”,
        "TBXXCS.xcs",
        {
                "language" : {
                        "Definition" : [[[
                                "@auxInfo",
                                "<descrip type='definition' />",
                                "null",
                                "content"
                        ]]],
                        "Note" : [[[
                                "@auxInfo",
                                "<note />",
                                "null",
                                "content"
                        ]]]
                },
                "concept" : {
                        "Subject" : [[[
                                "@auxInfo",
                                "<descrip type='subjectField' />",
                                "null",
                                "content"
                        ]]],
                        "Status" : [[[
                                "@auxInfo",
                                "<note />",
                                "category tag",
                                "content"
                        ]]],
                        "Source" : [[[
                                "@auxInfo",
                                "<admin type='source' />",
                                "null",
                                "content"
                        ]]],
                        "Note" : [[[
                                "@auxInfo",
                                "<note />",
                                "null",
                                "content"
                        ]]]
                },
                "term" : {
                        "Status" : [
                                [
                                        [
                                                "@unhandled",
                                                "<admin type='Status' />",
                                                "null",
                                "content"
                                        ],
                                        [
                                                "new",
                                                "nonstandardized",
                                                "proposed",
                                                "recommended"
                                        ],
                                        [
                                                "admitted",
                                                "deprecated",
                                                "legal",
                                                "preferred",
                                                "regulated",
                                                "standardized",
                                                "superseded"
                                        ]
                                ],
                                [
                                        "@termNote",
                                        "<termNote type='language-
planningQualifier' />",
                                        {
                                                "nonstandardized" :
"nonstandardizedTerm",
                                                "proposed" : "proposedTerm",
                                                "new" : "newTerm",
                                                "recommended" :
"recommendedTerm"
                                        },
                                        "content"
                                ],
                                [
                                        "@termNote",
                                        "<termNote type='administrativeStatus'
/>",
                                        {
                                                "deprecated" : "deprecatedTerm-
admn-sts",
                                                "regulated" : "regulatedTerm-
admn-sts",
                                                "admitted" : "admittedTerm-admn-
sts",
                                                "standardized" :
"standardizedTerm-admn-sts",
                                                "legal" : "legalTerm-admn-sts",
                                                "superseded" : "supersededTerm-
admn-sts",
                                                "preferred" : "preferredTerm-
admn-sts"
                                        },
                                        "content"
                                ]
                        ],
                        "Note" : [[[
                                "@auxInfo",
                                "<note />",
                                "null",
                                "content"
                        ]]],
                        "Context" : [[[
                                "@auxInfo",
                                "<descrip type='context' />",
                                "null",
                                "content"
                        ]]],
                        "Grammatical Gender" : [[[
                                "@termNote",
                                "<termNote type='grammaticalGender' />",
                                {
                                        "other" : "otherGender"
                                },
                                "content"
                        ]]],
                        "Grammatical Number" : [[[
                                "@termNote",
                                "<termNote type='grammaticalNumber' />",
                                {
                                        "other" : "otherNumber"
                                },
                                "content"
                        ]]],
                        "Usage Register" : [[[
                                "@termNote",
                                "<termNote type='register' />",
                                {
                                        "slang" : "slangRegister",
                                        "in-house" : "in-houseRegister",
                                        "bench-level" : "bench-levelRegister",
                                        "vulgar" : "vulgarRegister",
                                        "technical" : "technicalRegister",
                                        "neutral" : "neutralRegister"
                                },
                                "content"
                        ]]],
                        "Part of Speech" : [[[
                                "@termNote",
                                "<termNote type='partOfSpeech' />",
                                "null",
                                "content"
                        ]]],
                        "Source" : [[[
                                "@auxInfo",
                                "<admin type='source' />",
                                "null",
                                "content"
                        ]]],
                        "Category" : [
                                [
                                        [
                                                "@unhandled",
                                                "<termNote
type='unknownTermType' />",
                                                "null",
                                                "content"
                                        ],
                                        [
                                                "abbreviation",
                                                "acronym",
                                                "equation",
                                                "formula",
                                                "internationalism",
                                                "symbol"
                                        ],
                                        [
                                                "common name",
                                                "full form",
                                                "international scientific term",
                                                "part number",
                                                "short form",
                                                "transcribed form",
                                                "transliterated form"
                                        ],
                                        [
                                                "phraseologism",
                                                "stock keeping unit",
                                                "orthographical variant"
                                        ],
                                        [
                                                "antonym"
                                        ]
                                ],
                                [
                                        "@termNote",
                                        "<termNote type='termType' />",
                                        "null",
                                        "content"
                                ],
                                [
                                        "@termNote",
                                        "<termNote type='termType' />",
                                        "camel case",
                                        "content"
                                ],
                                [
                                        "@termNote",
                                        "<termNote type='termType' />",
                                        {
                                                "stock keeping unit" : "sku",
                                                "phraseologism" :
"phraseologicalUnit",

                                                "orthographical variant" :

"variant"

                                        },
                                        "content"
                                ],
                                [
                                        "@termNote",
                                        "<termNote type='antonymTerm' />",
                                        "null",
                                        "null"
                                ]
                        ]
                }
        },
            {},
            {}
]
```

Appendix II: Sample Mapping File 2 (Real-World Example)

```
[
      "TBX-Default",
      "TBXXCS.xcs",
      {
            "language" : {
                  "Definition" : [[[
                        "@Definition",
                        "<descrip type='definition' />",
                        "null",
                        "content"
                  ]]],
                  "D-Source" : [[[
                      "@D-Source",
                      "<admin type='source' />",
                      "null",
                      "content"
                  ]]],
                  "Note" : [[[
                        "@Note",
                        "<note />",
                        "null",
                        "content"
                  ]]],
                  "N-Source" : [[[
                      "@N-Source",
                "<admin type='source' />",
                "null",
                "content"
                  ]]],
                  "Graphic" : [[[
                      "@Graphic",
                      "<note />",
                      "null",
                      "content"
                ]]],
                  "G-Source" : [[[
                      "@G-Source",
                      "<xref type='xGraphic' >see target</xref>",
                      "null",
                      "target"
                  ]]]
            },
            "concept" : {
                  "Subject" : [[[
                        "@auxInfo",
                        "<descrip type='subjectField' />",
                        "null",
                        "content"
                  ]]],
                  "Status" : [[[
                        "@auxInfo",
            "<note />",
            "category tag",
            "content"
      ]]],
      "Source" : [[[
            "@auxInfo",
            "<admin type='source' />",
            "null",
            "content"
      ]]],
      "Note" : [[[
            "@Note",
            "<note />",
            "null",
            "content"
      ]]],
      "N-Source" : [[[
          "@N-Source",
    "<admin type='source' />",
    "null",
    "content"
      ]]],
      "Graphic" : [[[
          "@Graphic",
          "<note />",
          "null",
          "content"
    ]]],
      "G-Source" : [[[
          "@G-Source",
          "<xref type='xGraphic' >see target</xref>",
          "null",
          "target"
      ]]]
},
"term" : {
      "Status" : [
            [
                  [
                        "@unhandled",
                        "<unhandled type='Status' />",
                        "null",
                    "content"
                  ],
                  [
                        "new",
                        "nonstandardized",
                        "proposed",
                        "recommended"
                  ],
                  [
                        "admitted",
                        "deprecated",
                                    "obsolete",
                                    "not recommended",
                                    "legal",
                                    "preferred",
                                    "regulated",
                                    "standardized",
                                    "superseded"
                              ]
                        ],
                        [
                              "@termNote",
                              "<termNote type='language-planningQualifier'
/>",
                              {
                                    "nonstandardized" :
"nonstandardizedTerm",
                                    "proposed" : "proposedTerm",
                                    "new" : "newTerm",
                                    "recommended" : "recommendedTerm"
                              },
                              "content"
                        ],
                        [
                              "@termNote",
                              "<termNote type='administrativeStatus' />",
                              {
                                    "deprecated" : "deprecatedTerm-admn-sts",
                                    "obsolete" : "deprecatedTerm-admn-sts",
                                    "not recommended" : "deprecatedTerm-admn-
sts",
                                    "regulated" : "regulatedTerm-admn-sts",
                                    "admitted" : "admittedTerm-admn-sts",
                                    "standardized" : "standardizedTerm-admn-
sts",
                                    "legal" : "legalTerm-admn-sts",
                                    "superseded" : "supersededTerm-admn-sts",
                                    "preferred" : "preferredTerm-admn-sts"
                              },
                              "content"
                        ]
                  ],
                  "Note" : [[[
                        "@Note",
                        "<note />",
                        "null",
                        "content"
                 ]]],
                  "N-Source" : [[[
                      "@N-Source",
                "<admin type='source' />",
                "null",
                "content"
                  ]]],
"Context" : [[[
      "@Context",
      "<descrip type='context' />",
      "null",
      "content"
]]],
"C-Source" : [[[
    "@C-Source",
    "<admin type='source' />",
    "null",
    "content"
]]],
"Grammatical Gender" : [[[
      "@termNote",
      "<termNote type='grammaticalGender' />",
      {
            "other" : "otherGender"
      },
      "content"
]]],
"Grammatical Number" : [[[
      "@termNote",
      "<termNote type='grammaticalNumber' />",
      {
            "other" : "otherNumber"
      },
      "content"
]]],
"Usage Register" : [[[
      "@termNote",
      "<termNote type='register' />",
      {
            "slang" : "slangRegister",
            "in-house" : "in-houseRegister",
            "bench-level" : "bench-levelRegister",
            "vulgar" : "vulgarRegister",
            "technical" : "technicalRegister",
            "neutral" : "neutralRegister"
      },
      "content"
]]],
"Part of Speech" : [[[
      "@termNote",
      "<termNote type='partOfSpeech' />",
      "null",
      "content"
]]],
"Related Term" : [[[
    "@auxInfo",
    "<note />",
    "null",
    "content"
]]],
"Source" : [[[
      "@auxInfo",
      "<admin type='source' />",
      "null",
      "content"
]]],
"Type" : [
      [
            [
                  "@unhandled",
                  "<termNote type='unknownTermType' />",
                  "null",
                  "content"
            ],
            [
                  "abbreviation",
                  "acronym",
                  "equation",
                  "formula",
                  "internationalism",
                  "symbol",
                  "variant"
            ],
            [
                  "common name",
                  "full form",
                  "international scientific term",
                  "part number",
                  "short form",
                  "transcribed form",
                  "transliterated form"
            ],
            [
                  "phraseologism",
                  "stock keeping unit",
                  "orthographical variant",

            ],
            [
                  "antonym"
            ]
      ],
      [
            "@termNote",
            "<termNote type='termType' />",
            "null",
            "content"
      ],
      [
            "@termNote",
            "<termNote type='termType' />",
            "camel case",
            "content"
      ],
      [
            "@termNote",
            "<termNote type='termType' />",
            {
                  "stock keeping unit" : "sku",
                  "phraseologism" : "phraseologicalUnit",
                  "orthographical variant" : "variant"
            },
            "content"
      ],
      [
            "@termNote",
            "<termNote type='antonymTerm' />",
            "null",
            "null"
      ]
],
"Author[s]" : [[[
    "@auxInfo",
    "<note />",
    "null",
    "content"
]]],
"Journal Title" : [[[
    "@auxInfo",
    "<note />",
    "null",
    "content"
]]],
"Article Title" : [[[
    "@auxInfo",
    "<note />",
    "null",
    "content"
]]],
"Volume" : [[[
    "@auxInfo",
    "<note />",
    "null",
    "content"
]]],
"Publication Place" : [[[
    "@auxInfo",
    "<note />",
    "null",
    "content"
]]],
"Publisher" : [[[
    "@auxInfo",
    "<note />",
    "null",
    "content"
```

```
            ]]],
            "Publication Year" : [[[
                "@auxInfo",
                "<note />",
                "null",
                "content"
            ]]],
            "Page(s)" : [[[
                "@auxInfo",
                "<note />",
                "null",
                "content"
            ]]],
            "Book Title" : [[[
                "@auxInfo",
                "<note />",
                "null",
                "content"
            ]]],
            "Editor[s]" : [[[
                "@auxInfo",
                "<note />",
                "null",
                "content"
            ]]]
      }
},
{
    "conceptGrp" : [
       [
       "Graphic",
       "G-Source",
       "auxInfo"
       ],
       [
           "Note",
           "N-Source",
           "auxInfo"
     ]
    ],
    "languageGrp" : [
       [
       "Definition",
       "D-Source",
       "auxInfo"
       ],
       [
       "Graphic",
       "G-Source",
       "auxInfo"
       ],
       [
           "Note",
                "N-Source",
                "auxInfo"
          ]
         ],
         "termGrp" : [
            [
                "Context",
                "C-Source",
                "auxInfo"
            ],
            [
                "Note",
                "N-Source",
                "auxInfo"
          ]
         ]
      },
      {}
]
```