Draft Blueprint for the Linport BTS Profile

Jason K. Housley, Alan K. Melby, Arle Lommel
Provo, UT USA, September 2011, Linport-101-draft v8e

Abstract

BTS is a Linport (www.linport.org) profile. A Linport profile is a subset of the full Language Interoperability Portfolio (Linport) blueprint. All Linport blueprints are works in progress that will eventually be submitted to a standards body after wide-ranging open discussion within the world of multilingual document production. BTS stands for Bilingual Translation-project with Specifications. The BTS data model is neutral (a) as to structural styles used in various implementations and (b) as to whether a particular implementation is human or machine focused. The BTS profile is designed to offer a light-weight, translation-oriented Linport package for the elements of a single bilingual project.

1. Nature

The long-term objective of the BTS project is to define a freely available, vendor-neutral way to combine the elements of a simple translation project into a package that be used throughout the world of translation/localization with a wide variety of tools. The BTS project is s sub-project of the Linport project (see www.linport.org). The scope of the BTS project does not include refining the XLIFF standard. This task is left to the Oasis technical committee over XLIFF and to other projects cooperating with Oasis.

Some terms:

- **Linport** = **L**anguage **In**teroperability **Port**folio ("Linport" used ambiguously for the project, for the blueprint being created by the project and for instances of the data model evolving in the project)
- **blueprint** = a work in progress, not yet a draft standard, that will eventually be submitted to a standards body to help build a standard, that includes a data model and possible implementations of it
- **profile** = a subset for a particular audience and purpose
- **BTS** = Bilingual Translation-project with Specifications (a Linport profile)
- **Specifications** = structured translation specifications compatible with www.ttt.org/specs (a set of specifications is central to a translation request)
- **portfolio** = a collection of elements relevant to the authoring, translation/localization, and publication of content, sometimes called a package or an instance (there are a number of existing proprietary package formats that the Linport project hopes to bring together)
- **template** = an Xdossier-style description of a Linport data model
- **Xdossier** = a data object based on a directory structure containing files in several formats, with an emphasis on key-value pairs, and designed for browsing with web browsers. Xdossier is mappable to XML (http://dragoman.org/xdossier/)

The BTS Linport Profile is intended to accommodate a single translation project in a transitive manner: This means that a BTS is strictly bilingual and the same BTS instance is sent, modified, and returned by the parties involved in the translation. For example, a translation project manager may create a BTS instance and send it to a translator. The translator translates the source text, inserts the translation, and returns the modified BTS instance. BTS instances each have a unique ID. When a BTS instance is modified, the version must change so that users do not have to do a comparison of all files. This is open to discussion in connection with workflows used in the translation/localization industry. (Using the BTS in an intransitive manner would involve multiple BTS instances being sent and returned for a single project, i.e., intransitive usage means that changes to an instance result in an entirely new instance with a new ID: one BTS is sent and a separate one received.) The BTS blueprint is currently transitive and strictly bilingual and corresponds to a specific translation project rather than a group of related projects, which would require a full Linport. One set of translation specifications (i.e. a detailed translation request) must apply to all source and target document elements in a BTS instance.

2. Data Model

The BTS data model is a subset of the full Linport Data Model that underlies the full Linport Template. In order for a BTS to be valid, it must include at a minimum three terminal elements:

1. `metadata` element
2. `specifications` element
3. `source` document element.

The BTS also includes two additional terminal elements that correspond to the actual contents of the BTS instance: a `manifest` element and a human-oriented `index`. The `manifest` is intended for machine processing of the BTS contents whereas the `index` is intended for human access to the same information through a Web browser. The index is redundant, that is, it could be deleted and re-created automatically from the rest of a BTS instance. The manifest can be used to check the integrity of a BTS instance.

The BTS data model is intended to allow multiple implementations differing in primary audience (human or machine) and purpose. In section 3, multiple structural styles will be discussed. There is an active discussion within the Linport community regarding structural styles. The structural style chosen for an implementation the BTS data model may influence whether a BTS instance of that implementation is more oriented toward human consumption, using non-translation-specific tools such as a browser, or more oriented toward machine processing using translation-specific tools and perhaps connecting with content management systems.

The *non-terminal* elements that contain the required terminal elements are also required. At the root level the BTS consists of four elements (also found in the full Linport Template), some of which are non-terminal:

1. `portfolio information` (section 2.1)
2. `sets information` (section 2.2)
3. `sets` (section 2.3)

4. human-oriented `index` of the BTS contents (section 2.4)

2.1 Portfolio Information.

The `portfolio information` element must contain `manifest` and `metadata` elements. The `portfolio information` may also contain a `stylesheet` element consisting of CSS files and other document elements that are used in styling the instance for human consumption

2.1.1 *Manifest*

The manifest is a list of URLs for elements either contained or referenced in the BTS, excluding the manifest itself. In the case where an element is contained in the BTS instance, the URL in the manifest will be relative to the root of the instance; otherwise URLs will be links to remote items. *Absolute local file URIs are not supported*. Optionally, the manifest may include a one-way hash of each terminal element where possible.[1]

2.1.2 *Metadata*

Four metadata elements from the full Linport Template are required in a BTS:

1. `Profile`: The type of Linport Profile for the instance. This may be a URI or a name such as “BTS”.
2. `Implementation`: The choice of structural style or styles used to implement the data model in concrete instances. (e.g., Xdossier, Machine-Processing oriented, etc; see section 3.3)

[1] Common hashing algorithms such as md5 are preferred to promote interoperability. http://www.ietf.org/rfc/rfc1321.txt

3. `UUID`: The Unique ID for the instance.
4. `Version` (The exact format of "version" is under discussion.)

Optionally, the metadata may contain the following data elements:

1. `Project ID`: A grouping identifier for linking multiple instances together. BTS instances generated from a full Linport will generally share a Project ID.
2. `Contact`: The contact information associated with the instances to be used for the sender's contact information. `Contact` may include: `organization`, `name`, and `contact info` data elements along with any optional `notes` about contacting the person or persons specified. This will probably be expanded based on elements of the vCard data model (www.imc.org/pdi/vcardoverview.html).

2.2 Sets

The `sets` element always contains a single `set` element with a user-defined meaningful name (in a full Linport, as opposed to BTS, the `sets` element may contain multiple `set` elements). All `document` elements that comprise a `set` must conform to the same specifications, and the BTS accordingly allows for only one set of specifications.

`Document` elements must follow the Linport naming convention (where "lang" comes from ISO 639):

[name].[lang].[extension]

For example, a plain text document element may be named: "mytext.en.txt".

The `set` element found under sets must contain at least one `source document` element. If the source text is found at a remote location, a `document` element named "non-local-source.txt" without the language tag could be included to avoid an empty element. The contents of the non-local-source document element should be the URL for the source identical to that found in the manifest. Alternatively, the link could be placed in the manifest and the `set` element could be omitted from the BTS instance. This depends on the nature of the manifest, that is, whether it should be derivable from the other elements of a BTS instance or used to check the integrity of an instance. This is under discussion within the Linport project.

Document elements found in a set may include XLIFF.[2] For instance, a translation project might involve only one document in the set, namely an XLIFF file that includes both source and target text data.

2.3 Sets Information

The `sets information` root element contains a *single* `set` element with a user-defined or automatically generated name corresponding to the `set` element found in `sets`. The `set` element found under `sets information` contains a single `set information` element, as described in section 2.2.

2.3.1 Set Information

The `set information` element contains information about the corresponding `set element` found in `sets`. The `set information` element has one required terminal element, namely the translation `specifications`. In addition to the `specifications`, `set information` may contain a `reference` element, which is further subdivided into `human reference` and `machine reference` elements. The `human reference` element contains document elements intended for human readers, while the `machine reference` element contains machine-readable document elements. For example, a TMX file would be a machine-readable document element found under the `machine reference` element.

[2] http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html

2.4 Human Oriented Index

The `root index` element is intended as a human-readable roadmap of the actual BTS instance. The `index` is not a general table of possible entries but a guide for the actual payload of a particular BTS instance. The `index` should provide a mechanism (whether hypertext links or another method) for accessing all of the contents of the current BTS instance. The default structural style for the human-oriented `index` is an html file.

2.5 Data Model as an Outline

The following outline summarizes the preceding prose description of the BTS data model. It is anticipated that the BTS data model will also be described using UML (www.uml.org).
Note: optional elements marked with " [?] "

A. Root
   a. Portfolio Information (`portinfo`)
      i. Manifest
         1. Specifications URL
         2. Source URL
         3. Other Document URL [?]
         4. Etc. (other terminal elements of the BTS instance)
      ii. Metadata
         1. Profile
         2. Implementation
         3. UUID
         4. version (must change when instance is modified without changing UUID)
         5. Contact [?]
            a. Organization [?]
            b. Name [?]
            c. Contact Info [?]
      iii. Stylesheet [?] (CSS for index.html)
   b. Sets (note: In a full Linport instance there can be multiple sets, but not in a BTS)
      i. Set (*mysetname*)
         1. Source
         2. Target? (Note source and target may be a single XLIFF file)
   c. Sets Information (setsinfo)
      i. Set (*mysetname*)
         1. Set Information (setinfo)
            a. Specifications
            b. Reference [?]
               i. Human [?]
                  1. Reference Document (e.g. styleguide.pdf)
               ii. Machine
                  1. Reference Document (e.g. memory.tmx)
   d. Human Oriented Index (index.html; Required but can be derived from the BTS contents)

3. Portfolio Element Structural Styles

The BTS data model can be expressed using a variety of structural styles. A structural style is a method of representing data. All structural styles used in a BTS are *isomorphic,* meaning that there is no loss of data when converting between structural styles. However, different structural styles vary in terms of human readability, machine processability, and method of automatic validation. The following subsections describe the structural styles used for terminal and non-terminal elements.

3.1 Non-Terminal Elements

The non-terminal elements are those Linport elements that always contain at least one element. They are intended to function as a means of grouping related elements. The default structural style for a non-terminal element is a directory.

3.2 Terminal Elements

The terminal elements provide the content of the BTS and can be in one of several supported structural styles from key-value pairs to XML. Terminal elements structured as key value pairs can reside either in a single file with a set delimiter or as a directory of files where the filenames comprise the key and the file contents the value, as in Xdossier.[3] For example, a manifest.xml file with the appropriate data elements in XML and a manifest directory with key-value files corresponding to the data elements would both be valid `manifest` elements.

3.3 Multiple Isomorphic Instances

It is anticipated that there will be multiple isomorphic BTS instances that differ only in structural style. One may be more human-consumption oriented and one may be more machine-processing oriented. Hopefully these multiple equivalent BTS instances, which are all compliant with the BTS data model, will facilitate the structural style discussion.

4. Translation Specifications

The translation specifications are a core part of the BTS instance. The specifications help to promote high quality translation by providing essential project information. A set of structured translation specifications should be part of every translation request. Sometimes an initial request for translation/localization services will be incomplete. The detailed project specifications will be worked out cooperatively between the requester and the provider. Specifications can be broken down into four main areas of a translation project: Linguistic, Production, Environment, and Relationships. For detailed information about translation specifications see: http://www.ttt.org/specs/. The framework of translation specifications used in the BTS project have evolved over a decade with input from many stakeholders in the translation/localization industry. An existing translation quality-assurance standard (F2575) from ASTM International (www.astm.org) includes an earlier version of the framework of specifications used in BTS.

5. Larger Context

The BTS profile is part of the Linport project, which is merger of the MED project from the European Commission Directorate General for Translation that started several years ago and the Container Project that originated in March 2011 at the LISA standards summit. See www.linport.org for more information about the Linport project. Cooperation between the Linport project and other similar projects is invited.

[3] http://dragoman.org/xdossier/