**STEP**

Text conversion  
issues



**Text conversion issues**

**Contents**

[1 Introduction 3](#_Toc141247730)

[2 The formats 4](#_Toc141247731)

[2.1 USX 4](#_Toc141247732)

[2.2 Verse-per-line 4](#_Toc141247733)

[2.3 USFM 4](#_Toc141247734)

[2.4 OSIS 4](#_Toc141247735)

[2.5 Others 4](#_Toc141247736)

[3 USX 5](#_Toc141247737)

[4 Validity of input texts 7](#_Toc141247738)

# Introduction

Having spent longer than I care to remember looking at issues to do with text conversion, I may as well make a few notes in case they turn out to be useful to anyone else.

# The formats

## USX

USX is by far the commonest format with which we deal – basically because we are now taking very many of our texts from UBS’s Digital Bible Library (DBL), and all texts there are held in USX form. I am more familiar with USX than with other formats, and therefore devote an entire section to it (section 3).

## Verse-per-line

Sometimes also known as VerseLine or VL. In fact VL is not a standard at all, but rather a family of representations which are only fairly minimally related.

They have in common that they are all based upon plain text, and each verse always occupies a single line in the text file, usually with some representation of the verse reference at the front (eg book + chapter + verse), followed by the actual text.

However, there is no standard format for the scripture reference, and the different variants differ as to what further features they support (eg footnotes, etc).

Each variant therefore requires its own processing.

## USFM

USFM was the predecessor to USX, and we encounter it relatively seldom. The freely available Paratext tool from UBS can be used to convert USFM to USX, and I recommend you do this, because USX, as an XML dialect, is easier to process than is USFM.

## OSIS

OSIS, like USX, is an XML dialect[[1]](#footnote-1), and in theory therefore is relatively easy to process. We make quite heavy use of it within STEP, but only as an intermediate format en route to creating Sword modules – never as an input format.

## Others

I have very occasionally come across other formats – JSON, plain text or even, on occasion, a spreadsheet containing one verse per cell. Each of these would require its own specific processing.

# USX

UBS’s USX is a ‘living’ standard, in the sense that it is still evolving. This means that you may find yourself having to cope with more than one version, and the earlier versions differ from the later ones in quite significant ways.

The following is a list of issues which occur to me. They are not in any particular order.

* Scripture references: Different versions differ as to how scripture references are recorded. In earlier versions, chapter and verse tags carry just a ‘number’ attribute, so that the actual scripture reference for a given verse has to be deduced by tracking the book name from the enclosing book tag, along with the number attribute of the most recent chapter tag and the number attribute of the verse. More recent versions instead use attributes sid or eid (explained in more detail shortly) which contain the full reference.
* Milestone markers: As I recall, chapters and verses may come in either of two flavours – enclosing tags, or milestone tags. In fact I think all of the texts I have seen use the milestone version.[[2]](#footnote-2)
* Verse start- and end- markers: In milestone mode, you need to be able to determine where verses start and end. All versions required the start of verses to be marked with a verse tag, but early versions neither required, nor indeed supported, a verse-end tag: it was simply assumed that a verse ended prior to the next verse start tag. More recent versions require a verse tag at both start and end, both giving the full scripture reference for the verse. They are distinguished in that the verse tag at the start of the verse gives this reference in a ‘sid’ attribute, while the verse tag at the end gives it in an ‘eid’ attribute.
* Elisions: Don’t forget that verses may be elided. sids and eids may therefore be ranges rather than single verses (and where verse tags carry a number attribute rather than an sid or an eid, the number attribute may be, for instance, 1-3).
* Out-of-order verses: You should not assume that a valid text will have all verses in the correct order. Sometimes translators deliberately arrange verses out of order in the belief that the translation is thereby clarified.
* Cross-boundary markup[[3]](#footnote-3): The use of milestone tags is intended to make it possible for formatting or semantic markup to run across verse boundaries – and I have yet to see a text which does not take advantage of this.[[4]](#footnote-4) Cross-boundary markup considerably complicates situations in which we have to treat a verse, or a block of consecutive verses, as an entity in their own right, because it can be difficult or impossible to excise them from their context. As an example, this is a complicating factor where reversification requires blocks of verses to be moved. It is possible to avoid at least some cross-boundary markup by moving verse/eid tags around – they can always be moved towards their corresponding verse/sid so long as to do so does not leave canonical text outside of the verse.
* Tables: Tables are a particular problem, because they invariably span several verses. The placement rules governing where verse tags should be positioned relative to table/row/cell tags are far from obvious, and cannot usually be observed without then introducing problems with cross-boundary markup. The only really viable approach I have found is to turn the entire table into one large elision, and remove the verse tags from within the table. Thus if the table spans vv1-10, I turn vv1-10 into an elision, with vv1-9 empty and v10 containing the entire table, perhaps with some kind of text markers to show the human reader where the verse boundaries originally were. Clearly this is non-ideal – added value features like verse vocabulary and interlinear won’t work properly – but I can’t see a better alternative.
* Cross-references: Regrettably cross-references come in a variety of flavours, and there are definite benefits to reducing them to a standard form.
  + In most cases, you have a ref tag. The content of ref tags gives the target reference in vernacular form, and this is what the user sees and clicks on. And ref tags carry a loc attribute which gives the target reference in USX form, which can be used internally to establish the link.
  + Typically the ref tag is enclosed within char:xt (or char:xot, indicating a cross-reference which should be processed only on a text which contains the OT; or char:xnt, or char:xdc).
  + However char:xt etc doesn’t always contain a ref tag: sometimes it contains just the target reference in vernacular form. Sometimes instead of having an embedded ref tag, char:xt etc have an attribute href-link which gives the target in USX form – in which case, taken with the content of the char:xt it is relatively straightforward to fabricate an equivalent ref tag and insert it into the char:xt – which I recommend.
  + Sometimes, though, char:xt does not have an href-link tag, in which case there is nothing much you can do unless you are able to parse the (vernacular) content of the tag and generate the corresponding USX version yourself in order to fabricate a contained ref tag.
  + And finally, there are all sorts of validation issues, which we look at in section 4.

# Validity of input texts

Regrettably to assume that any text is fully valid is probably to go too far.

With texts which are based upon an XML dialect, it is probably safe to assume that they are at least valid XML (and it will rapidly become apparent if they are not); but even with texts from DBL (where one might imagine that some form of validation has been applied), there appear to be no guarantees beyond XML-validity.

One issue here is the complexity of some of the standards, which leave translators unsure of what facilities are available, unsure of how to use the facilities which *are* available, unsure of the valid ordering and nesting of tags, and – despite the complexity – in some cases, without the facilities they require. As a particular example of this last issue, we have seen a number of texts which make extensive use of bullet point lists not – we think – because the translators actually want bullet points, but because they want indented paragraphs and, finding no obvious way of indicating them, resort to bullet point markup in the hope that some downstream magic will sort things out.

* Cross-reference are seldom validated at source, and there are numerous issues which may need to be addressed.
  + USX supports cross-reference tags which target individual verses and which target contiguous ranges, but not mixed *collections* of verses and / or ranges within a single tag – and downstream representations like OSIS do the same. It is certainly not uncommon to receive texts in which this restriction has not been observed. It is best addressed by converting the cross-reference containing the collection into a series of separate references containing the elements of the collection. However, this may be difficult because it requires you to be in a position to parse the vernacular form of the reference. If this is not feasible, the cross-reference might be converted to a plain vanilla flavour footnote, or else would need to be dropped altogether.
  + Either or both of the USX or vernacular form of the reference may be syntactically invalid.
  + Either or both may be semantically invalid (may name a non-existent book like ZQJ, or may point to a clearly ridiculous verse – v9999 for instance).
  + Either or both, though fully valid, may point to somewhere not available in this text – eg a reference to an OT verse from a text which contains only the NT.
  + The USX and vernacular forms of the reference may not be mutually compatible (ie they may point to different places or cover different numbers of verses).
  + And you also have to bear in mind that the vernacular form may legitimately be incomplete (eg may lack the book name, relying upon the user to deduce this from context) and may contain noise words, such as the ‘vv’ in vv1-10.

\* End of document \*

1. Unlike USX, which is still evolving, OSIS is no longer being maintained and extended. [↑](#footnote-ref-1)
2. My personal opinion is that processing is simplified if you are able to use the enclosing version. In fact for reasons discussed shortly, this is not usually feasible for verses, but I have never found any difficulty in converting chapters to enclosing form, and definite advantages in doing so. [↑](#footnote-ref-2)
3. By this I mean a case where an enclosing tag, such as a paragraph, starts within one milestone-based element and finishes within another. For example, you might have a paragraph which starts before v1 of a chapter, and finishes partway through verse 5. Note that I do not regard markup as cross-boundary if it entirely encloses elements. Thus a paragraph which starts outside of v1 and finishes outside of v5 contains vv1-5 in their entirety, and does not count as cross-boundary. [↑](#footnote-ref-3)
4. In theory markup could also run across chapter boundaries, but I have never seen an example of this – and indeed it is precisely because this does not occur that it is possible to convert chapter milestone tags into enclosing tags. [↑](#footnote-ref-4)