

XLingPaper's use of T_EX Technologies

H. Andrew Black, Hugh J. Paterson III

Abstract

We discuss the use of T_EX technologies by XLingPaper, an authoring tool for producing academically oriented publications with features required for linguistic publishing. We present the T_EX modules used and the rationale for the history of XLingPaper development.

1 Introduction

Within the publishing industry there are several notable products for producing complex documents in beautiful formats. T_EX [19] [20] is one of the well known publishing technologies used to meet these needs. Since 2000, XML-based technologies such as XSL-FO¹ or the T_EXML² project [23] have been used to integrate content and compose complex documents such as textbooks and maintenance manuals. Requirements for composing these large, inter-linked documents birthed the development of tools like XMLmind³ and Xpublisher.⁴ These can be used to compose the content within predefined XML structures. XLingPaper [5] [6] [7] seeks to provide a constrained environment in which authors of complex works dealing with language descriptions and linguistic analyses can focus on content structure independently from the styling requirements of documents. In this way the underlying design principle of XLingPaper maximizes the SGML design practice of separating content from presentation. With XLingPaper, authors can keep content structure independent from style information and thereby provide maximal transfer-ability between publishing styles. The software does this while providing authors a clear structured interface for authoring content.

The XLingPaper software has a growing number of users who have successfully typeset complex documents including:

- master theses [38] [21] [26],
- doctoral dissertations [13] [29],
- textbooks [24],
- linguistic grammars [9],
- journal articles [8], and
- bilingual software documentation [1] [2].

XLingpaper⁵ is a plug-in to the XMLmind XML Editor. XLingPaper benefits from the XMLmind

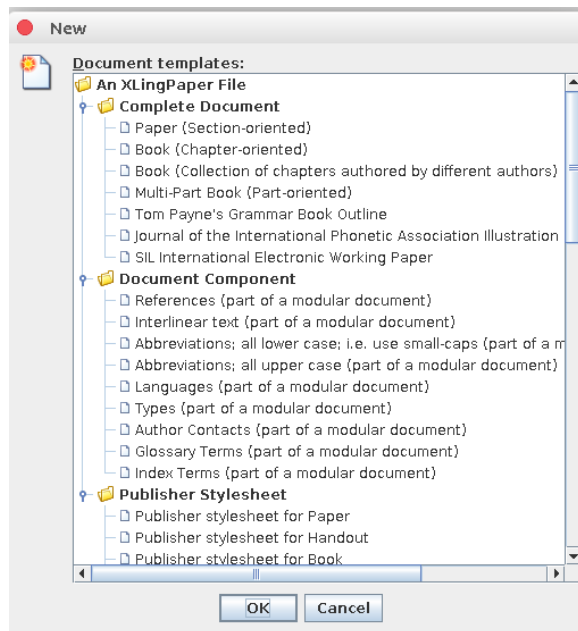


Figure 1: XLingPaper predefined document types via DTD

XML Editor's Java-based implementation which allows it to be used on Mac OS X, Windows, and Linux. XLingPaper, via a DTD, defines several document classes (articles, books, chapters, etc., as illustrated in Figure 1), in each case providing document layout sections (paragraphs, examples, endnotes, etc.). By working within the user-interface of the XMLmind XML Editor, as shown in Figure 2, formatting errors are reduced because users are constrained on where in the document flow they can introduce block and line level document elements. That is, first, authors cannot input X_LAT_EX code directly into the document and second, the introduction of layout sections within the document flow is constrained via the DTD.

2 What is XLingPaper?

As previously mentioned, XLingPaper is an XML- and Java-based computer plug-in for the XMLmind XML Editor. It is designed to reduce friction in the process of writing, reading, and publishing linguistic papers, grammars, and books by removing common time-syncs related to inconsistent formatting. A full list of benefits to all parties in the publishing work flow is available [7]. For many PDF is the quintessential file format for final publication of publishing outputs. XLingPaper supports PDF production but as illustrated in Figure 3, XLingPaper can produce documents with at least five outputs, all from the same source document:

¹ www.w3.org/TR/xsl11

² getfo.org/texml

³ www.xmlmind.com/xmlmind

⁴ www.xpublisher.com/products

⁵ software.sil.org/xlingpaper

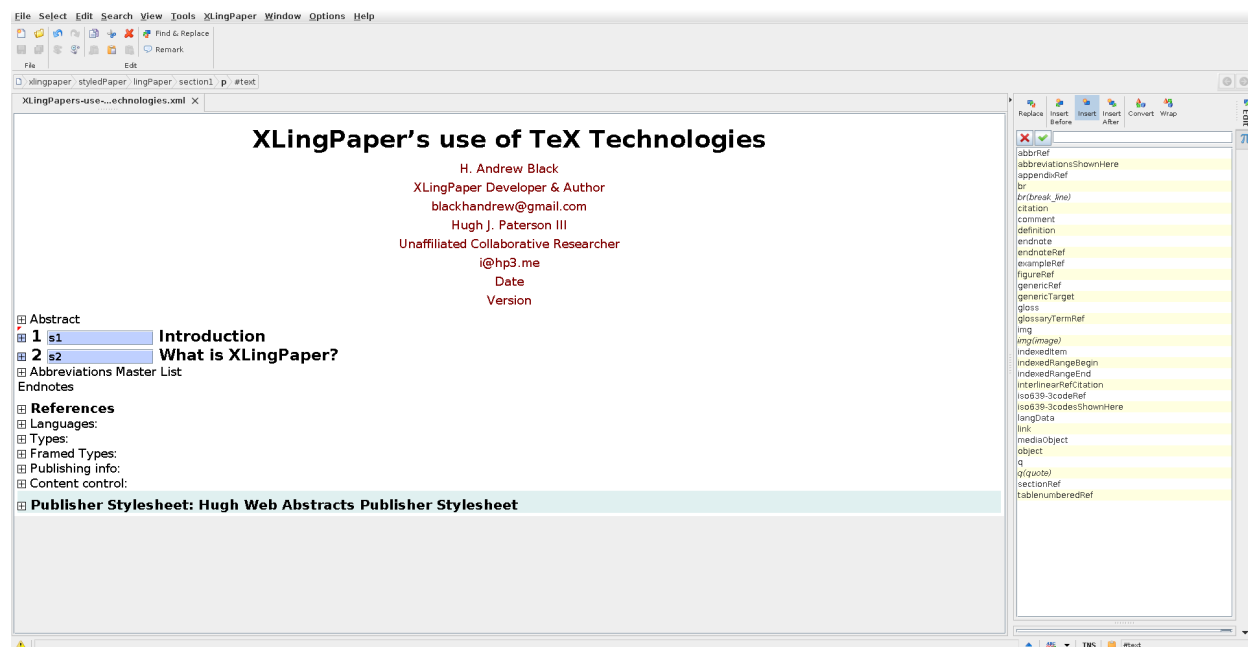


Figure 2: XLingPaper’s user interface. Left side: document content editing. Right side: block and line level units available for use at the cursor location.

- PDF (version 1.5),
- Web pages (HTML 4),
- Microsoft Word (.doc),
- Open Office Writer Document (.odt), and
- ePUB.

XLingPaper automatically numbers tables, examples, figures, and sections. It keeps track of internal references to these entities along with citation references and gloss abbreviations. This keeps numbering and reference links dependable and automated. It also automatically generates indexes, a table or list of abbreviations used, and a section for references cited (using a custom references implementation).

Unlike most editing programs which are based on either the WYSIWYG paradigm or as text editors used to code or produce Markdown, XLingPaper (via the XMLmind XML Editor) is a structured editor. Rather than visually structuring the document to look the way it is to be formatted, the author “marks up” the items in the document according to their kind. One of the many benefits that using a DTD provides is that there is a “grammar” of what a well-formed linguistic document looks like. This makes moving, replacing, switching, or reordering sections, chapters, and examples less error prone because it prevents users from inadvertently creating ill-formed documents.

The following sections of this paper discuss the \TeX technologies used by XLingPaper.

3 XLingPaper and \TeX

The following sections provide more detail on the linguistic publishing context, design requirements and packages used by XLingPaper.

3.1 \TeX and Linguistic document production

\TeX has long been embraced by linguists. Peter [27] writes of a personal communication with Don Knuth where Knuth suggests that linguists were some of the earliest adopters outside of mathematicians. Thiele [33] in an interview given in 2007 states that she was typesetting linguistic journals via \TeX in 1983—a date prior to the release of Knuth’s book on using the \TeX typesetting system [19]. Thiele [32] gives an early overview of \TeX use in linguistics with mention of significant repositories outside of CTAN. A slightly more recent (2004) update by Peter [27] provides some additional tips and tools for typesetting common information structures in linguistic publishing. The \TeX community produced many packages which would shape the visual face of publishing in linguistics, including TIPAP⁶ by Rei [30] which is an excellent typeface for phonetic transcriptions and

⁶ ctan.org/pkg/tipa

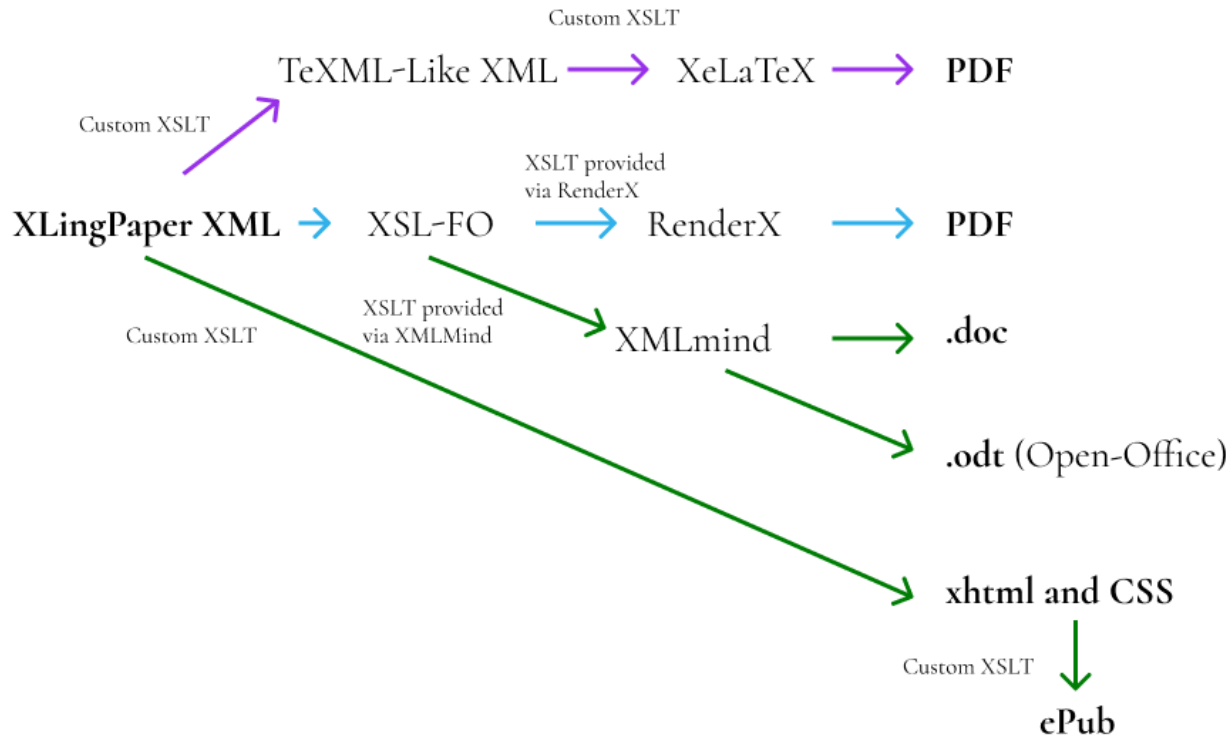


Figure 3: XLPingPaper’s data processing pipeline to multiple formats

`pst-asr`⁷ by Frampton [14] for autosegmental representations. As the Peter and Thiele articles list out and review (through 2004), there are special packages for all areas of linguistics from syntax trees `qtrees`⁸ to specialized packages for presenting examples and interlinear glossed texts such as `expex`⁹. In addition to packages which provide specific functionality, there are packages which are essentially collections of macros such as `covington`¹⁰ and `gb4e`¹¹ which serve a variety of page layout functions targeted at publishing in linguistic topics.

The CTAN repository currently lists fifty-four different T_EX packages for linguistic typesetting,¹² though some of these packages also include capabilities targeted as multi-lingual or multi-script publications or are specific styles for publications at linguistic programs at institutions of higher education (and there may be more which are not tagged but should be). Several of the packages tagged “linguistic” pre-date Unicode [34] but still see significant use. Sometimes it is the case that secondary packages are de-

veloped in an attempt to “fix” publishing outputs in different ways to bring Unicode features along with the features of the original package. For example, TIPA is not Unicode compatible, but the packages `unitipa`¹³ and `tipauni`¹⁴ seek to address different implications of not publishing with Unicode and using TIPA. Understanding the long history of publishing and the interdependency that packages have presents additional complexities to new T_EX users.

Mentioning these complexities is important to the XLPingPaper discussion for two reasons. First, it exemplifies some of the complexities that the user interface seeks to simplify as it presents authors not just a visual environment for document composition, but also a cohesive output solution. Second, while the diagrams in linguistic books and journals since the 1980’s exemplify many beautiful, sharp, crisp, illustrations created directly in T_EX, many trainers of T_EX tools, but not all,¹⁵ have steered authors towards a more generic set of packages which do not include specific diagram creating macros.¹⁶

⁷ ctan.org/pkg/pst-asr

⁸ ctan.org/pkg/qtrees

⁹ ctan.org/pkg/expex

¹⁰ ctan.org/pkg/covington

¹¹ ctan.org/pkg/gb4e

¹² ctan.org/topic/linguistic

¹³ ctan.org/pkg/unitipa

¹⁴ ctan.org/pkg/tipauni

¹⁵ For counter examples see [22] [31] [16] and [28].

¹⁶ Among others, see the Linguistics Dissertation guide for the University of Hawai’i at Mānoa [17], University of Pennsylvania [11], and Language Science Press Guidelines [25].

Rather, they suggest that authors use secondary illustration tools to generate illustrations and then include them as vector PDFs or images. In fact this second method is the document production path that the XLingPaper philosophy follows.

Unicode was introduced in 1991 and by the early 2000's it was being heralded in academic linguistics as a best practice along with document and data storage in XML formats [3] [4] [35]. Due to the heavy reliance on Unicode in modern language documentation and linguistic work, XLingPaper specifically uses X_QLaTeX and compatible packages to produce PDF outputs. This brings continuity to the text input process for users across their workflows. It also makes importing and using language or phonetically transcribed examples simpler by removing the need to use macros to create characters.

3.2 Design desiderata for XLingPaper outputs via T_EX

From the outset, XLingPaper was designed to be free. The XMLmind XML Editor had a Personal Use License that met this requirement for the vast majority of the target audience of XLingPaper. The few who did not meet the terms of that license most likely would be able to afford to purchase (or have their organization purchase) a professional version of the XMLmind XML Editor. The actual XLingPaper plug-in has always been free.

XLingPaper development started in 2001. In 2006, it added XSL-FO for PDF production. Prior to 2009, XLingPaper used RenderX¹⁷ to produce PDF documents. However, in 2009 plans were made to add X_QLaTeX-based output to XLingPaper because, while there was a free version of RenderX, the output contained a watermark. By implementing the ability to export to PDF via X_QLaTeX, there would be no water marks in PDF documents. The X_QLaTeX method of PDF production is now the default method to produce PDF documents.

When the X_QLaTeX method of PDF production was introduced, XLingPaper had a way to format output per a user-created publisher style sheet. This meant the developer (Andrew Black) needed to be able to map from an XLingPaper publisher style sheet to X_QLaTeX. Mapping style sheet information contained in the XML was the second criterion. It was known that LaTeX was the ideal T_EX implementation to target. However, pure LaTeX came with predefined output formatting for front matter, chapters, sections, back matter, etc. Pure LaTeX, then, would not allow direct control of formatting of all

of these per an XLingPaper user-defined publisher style sheet. This meant overriding these standard features of LaTeX with a custom implementation of the T_EX commands needed to control formatting. However, memoir¹⁸ [36] [37], a recent discovery for the programmer of XLingPaper, accomplishes many of the same tasks and could be considered to replace some of the custom code if it were shown to be easy to implement and that the size of the total XLingPaper code base would be reduced.

The third criterion concerned some of the target audience for XLingPaper. Many of the expected users of XLingPaper live and work in places around the world where Internet connections are characterized by high costs, low bandwidth capacity, and general unavailability. Therefore, the download required to install XLingPaper needed to be as small as possible. The size constraint impacts XLingPaper because its distribution must be independent of larger mainstream T_EX distribution solutions such as T_EXLive which have a large footprint. Therefore the developer determined which LaTeX packages and binaries were needed and created a custom installation package for just those items (a list of these packages is included in section 6.1).

The twenty LaTeX packages that are part of the custom XLingPaper distribution are still rather large for someone for whom Internet bandwidth is an expensive and inconsistent commodity. To reduce bandwidth requirements two assumptions were made which have more or less proven to obtain. The first assumption that the developer made was that the twenty packages and binaries would not need to change over time; in contrast, the second assumption was that XLingPaper would acquire new features and need bug fixes. These assumptions resulted in an architecture where page layout information expressed in XML is translated via custom T_EX commands to either T_EX directly or to commands understood by LaTeX packages distributed with XLingPaper. This abstraction layer was then executed when the X_QLaTeX file was processed. This middle layer has granted XLingPaper flexibility in adding new code and capabilities while keeping the “heavy” LaTeX packages stable. The net result is a “heavy” first install package, but light-weight upgrade packages. In the thirteen year history of development, there have been a few occasions where upgrades have required the download of new “heavy” packages. One such case was when the ability to use framed units was added. These elements depend on

¹⁷ www.renderx.com

¹⁸ ctan.org/pkg/memoir

the `mdframed`¹⁹ package [10]. The architecture separating stable packages from custom code, however, has generally worked out well and kept update sizes low.

3.3 PDF production

When an author has XLPaper produce PDF output via $\text{Xe}_{\text{L}}^{\text{A}}\text{T}_{\text{E}}\text{X}$, XLPaper produces a $\text{T}_{\text{E}}\text{X}$ ML-like XML file. This is then converted into $\text{L}_{\text{A}}^{\text{A}}\text{T}_{\text{E}}\text{X}$ format via a set of XSLT transforms and processed via $\text{Xe}_{\text{L}}^{\text{A}}\text{T}_{\text{E}}\text{X}$ to produce the PDF. Figure 3 contains a diagram of this process.

3.4 $\text{T}_{\text{E}}\text{X}$ ML

In the process of planning for the transition of the default PDF renderer from RenderX to $\text{Xe}_{\text{L}}^{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{T}_{\text{E}}\text{X}$ ML was discovered. However, initial analysis understood it to have two infelicities:

1. $\text{T}_{\text{E}}\text{X}$ ML had Python as a dependency and the developer did not want to force XLPaper users to have to install a version of Python specifically for $\text{T}_{\text{E}}\text{X}$ ML when that version might conflict with other installed versions of Python. Furthermore, this approach would make the installation package much larger due to the inclusion of Python.
2. XLPaper users require a high degree of control for white space, something it was not clear was possible via $\text{T}_{\text{E}}\text{X}$ ML.

He did implement some Java code to deal with mapping certain characters used in $\text{T}_{\text{E}}\text{X}$ commands to their $\text{T}_{\text{E}}\text{X}$ equivalents. These include characters such as `[`, `]`, `<`, and `>`. He used Java because the XMLmind XML Editor is written in Java and XLPaper already used Java code to improve the user experience in the XMLmind XML Editor.

3.5 Ling- $\text{T}_{\text{E}}\text{X}$

When the developer began to implement the $\text{Xe}_{\text{L}}^{\text{A}}\text{T}_{\text{E}}\text{X}$ based output, he discovered the Ling- $\text{T}_{\text{E}}\text{X}$ group²⁰ which also ran the Ling- $\text{T}_{\text{E}}\text{X}$ mailing list from 1995–2018.²¹ Today many of the mailing list participants can be found interacting on the $\text{T}_{\text{E}}\text{X}$ stackexchange.²² The Ling- $\text{T}_{\text{E}}\text{X}$ website mostly advertised the In 2009 several other related $\text{T}_{\text{E}}\text{X}$ webpages discussing linguistic typesetting also existed.²³ Linguistics as an academic discipline has embraced $\text{T}_{\text{E}}\text{X}$ technology for a long time.

¹⁹ ctan.org/pkg/mdframed

²⁰ web.archive.org/web/20150702123633/http://heim.ifi.uio.no/~dag/ling-tex/

²¹ ling-tex.ifi.uio.no/narkive.com

²² tex.stackexchange.com

²³ www.essex.ac.uk/linguistics/external/clmt/latex4ling

<code>attachfile2</code>	<code>lineno</code>
<code>booktabs</code>	<code>longtable</code>
<code>calc</code>	<code>lscape</code>
<code>color</code>	<code>mdframed</code>
<code>colortbl</code>	<code>multitrow</code>
<code>etoolbox</code>	<code>normalem</code>
<code>fancyhdr</code>	<code>polyglossia</code>
<code>fontspec</code>	<code>setspace</code>
<code>footmisc</code>	<code>tabularx</code>
<code>hyperref</code>	<code>xltxtra</code>

From what he could tell, the packages that help with interlinear texts did not allow for the larger number of capabilities XLPaper already dealt with. These include no limits on the number of lines within an interlinear grouping, no limits on the number of free (and/or) literal lines, including a source reference within the interlinear, and especially tagging interlinear items with an ISO 639-3 code for the language used in the interlinear. So he rolled his own. Figure 4 contains an example output with some of the special capabilities XLPaper offers.

4 Typesetting tasks XLPaper users often encounter

Linguistic documents have several formatting needs that other kinds of documents do not. This section discusses some of them.

4.1 Numbered example layouts

Linguistic documents usually have many numbered examples. The prose often refers to examples near the material or to previous examples. XLPaper automatically keeps track of the numbers. Besides table-like layouts, linguists also need lists of words along with their glosses (as shown in Figure 4), interlinear clauses (as shown in Figure 3), and even having headings in portions of the example.

4.2 Automatically wrapping interlinear texts

Many linguists want to include interlinear glossed text in their document. XLPaper allows these to be wrapped automatically which makes the author's job much easier. Figure 5 shows one such text portion.

4.3 Gloss abbreviations

Linguists standardly use glosses for indicating the meaning of pieces of words (morphemes). XLPaper allows the author to define a set of abbreviations and their definitions. When producing the output, XLPaper creates hyperlinks between the abbreviation and its definition.

Una frase cuantificadora puede acompañar al sustantivo (véanse [Los Cuantificadores](#) y [Los Números Cardinales](#)). Cuando se presenta esta frase, siempre va delante del núcleo de la frase nominal, como en los ejemplos en (2).

- (2) a. [tcf- NÁa majñuu nákhū iduu iya'
Zila] náā māhjūùⁿ nákù īdūū ījā?
LOC entre TOT.cuatro ojo.3SG agua
'De entre los cuatro manantiales'⁵ [Smajiin:6]
- b. [tpl- Gí'doo witsu rakhóó mikhúdú
Tlac] EST.tener.3SG cinco nariz.3SG (EST).picud@
'Tiene cinco esquinas picudas' [FC:5.1]

El cuantificador puede presentarse en construcciones donde no hay sustantivo expreso, como se explica en [Los Cuantificadores](#). Un ejemplo se incluye aquí.

Figure 4: Interlinear example

Bantu D30 canonical infinitive verb pattern is exemplified in the Mbo data in (11):

- (11) a. [ex[ko-sis-o]ex] [ex[[- - -]]ex] move forward
b. [ex[kɔ-kij-a]ex] [ex[[- - -]]ex] act
c. [ex[ko-ɓund-o]ex] [ex[[- - -]]ex] break
d. [ex[kɔ-ɓut-a]ex] [ex[[- - -]]ex] become long
e. [ex[ko-ɓep-o]ex] [ex[[- - -]]ex] wink
f. [ex[kɔ-kɛk-a]ex] [ex[[- - -]]ex] decorate
g. [ex[ko-sok-o]ex] [ex[[- - -]]ex] cackle
h. [ex[kɔ-mvɔd-a]ex] [ex[[- - -]]ex] suck
i. [ex[kɔ-bab-a]ex] [ex[[- - -]]ex] carry

Figure 5: List of words

Rikha²

FC:1

Rikha rígi' najmaā náā yúoo' rā'khā ká',³ rā'khā suan'⁴
 flor.de.calabaza INAN:PROX IMPF.producirse LOC guía.3SG calabaza.especie calabaza.especie

khamí náā yúoo' rā'khā' májin'.⁵
 y LOC guía.3SG chilacayote

'La flor de calabaza se da en la guía de la calabaza de Castilla, de la "calabaza espina" y del chilacayote.'

FC:2

Rí rikhoo rā'khā suan', nagí'dūu namídi rí
 SBD:INAN flor.de.calabaza.3SG calabaza.especie IMPF.empezar.3SG.FM ± IMPF.florear SBD:INAN

gūn' agóstó.
 luna agosto*

'La flor de la "calabaza espina" empieza a abrir en el mes de agosto.'

FC:3

Mba'ju, mujmū' rí'jiūu.
 (EST).grande:PL (EST).amarill@ flor.3SG

'Sus flores son grandes y amarillas.'

Figure 6: Wrapped interlinear text

5 Outputs L^AT_EX allow that others do not

While X_LLingPaper has a large array of linguistically-oriented formatting capabilities, there are some that only the X_LL^AT_EX output can produce. This is, of course, due to the formatting power of T_EX and X_LL^AT_EX.

5.1 Automatically wrapping interlinears

One of the most popular features of X_LLingPaper is its ability to automatically wrap long interlinear examples and lines in interlinear texts. It does so by formatting each aligned word in an hbox and then having X_LL^AT_EX put them together in a hanging indent paragraph. This is based on the work of Kew & McConnel 1990 [18].

5.2 Font rendering

X_LL^AT_EX renders fonts extremely well. It can even handle special features requiring Graphite²⁴ processing. For other outputs, some fonts (such as Charis SIL) may not line up vertically as expected due to them having different ascender and descender values. One has to add custom commands to deal with these. In the case of Graphite, they may not be able to be done at all. The RenderX way of producing PDF cannot handle stacked diacritics, but the X_LL^AT_EX way does it very well.

²⁴ graphite.sil.org

5.3 Hyphenation for non-English languages

Since we use the `polyglossia` package, one can write an X_LLingPaper document in a non-English language and X_LL^AT_EX will hyphenate according to that language's hyphenation rules.

5.4 Author contact information

X_LLingPaper allows one to define a set of contact information for authors. Only the X_LL^AT_EX output is able to format them correctly.

5.5 Vertical fill

For title page material, only the X_LL^AT_EX output allows using vertical fill between items. The other outputs require using overt, fixed spacing values.

5.6 Blank page

When one wants a totally blank even-numbered page between a final odd-numbered page and the next odd-numbered page which begins, say, a chapter or appendix, only the X_LL^AT_EX approach is able to do this.

6 Features other outputs have that the L^AT_EX output does not

X_LL^AT_EX does not allow for custom table cell padding and spacing. Having said that, Andy cannot remember any X_LLingPaper user ever asking for a way to do this for the X_LL^AT_EX output. It just looks great.

Background color is not available for section titles.

Section 11.17.1.1 “Known limitations of using X_YL^AT_EX” in the X_YLingPaper user documentation lists known problems.

6.1 List of L^AT_EX packages used

X_YLingPaper currently uses the following X_YL^AT_EX packages (in alphabetical order):

<code>attachfile2</code>	<code>lineno</code>
<code>booktabs</code>	<code>longtable</code>
<code>calc</code>	<code>lscape</code>
<code>color</code>	<code>mdframed</code>
<code>colortbl</code>	<code>multirow</code>
<code>etoolbox</code>	<code>normalem</code>
<code>fancyhdr</code>	<code>polyglossia</code>
<code>fontspec</code>	<code>setspace</code>
<code>footmisc</code>	<code>tabularx</code>
<code>hyperref</code>	<code>xltxtra</code>

6.2 Custom T_EX commands

X_YLingPaper has a number of custom commands that enable it to handle various tasks in a way that is consistent with our desired outcomes. The following lists some of them in a schematic way:

Command for	Purpose
Table of contents	Store and retrieve page numbers; format the contents.
Lists	Numbered and bulleted lists with control over indents, etc.
Examples	Example number and example content, where the content can be a line, a list of lines, a set of words, a list of a set of words, interlinear, a list of interlinears, etc.
Indexes	Handle keeping track of X _Y LingPaper’s indexing capability, including page numbers.
Interlinears	Handle lines in an interlinear text or example, including dealing with an ISO 639-3 code in an interlinear example.
Block quotes	Handle special cases needed for block quotes.
Table headers	Attempt to calculate a column’s width via its contents.

7 Conclusion

While the X_YLingPaper approach to writing linguistic documents has great value in and of itself, the fact that it can produce great looking output via X_YL^AT_EX makes it very worthwhile learning to use. We feel that being able to produce PDF via X_YL^AT_EX has made X_YLingPaper a fantastic tool for linguists.

References

- [1] Beadle, Jennie, and Matthew Lee. 2020a. *Paratext 9 Manual – in English*. SIL International. lingtran.net.
- [2] Beadle, Jennie, and Matthew Lee. 2020b. *Paratext 9 Manual – in French*. SIL International. outilingua.net.
- [3] Bird, Steven, and Gary Simons. 2002. Seven Dimensions of Portability for Language Documentation and Description. In ISCA SALT MIL SIG: “Speech and Language Technology for Minority Languages,” 23–30. Las Palmas, Canary Islands, Spain: ELRA. www.lrec-conf.org/proceedings/lrec2002/pdf/ws15.pdf#page=29.
- [4] Bird, Steven, and Gary Simons. 2003. Seven Dimensions of Portability for Language Documentation and Description. *Language* 79 (3):557–82. www.jstor.org/stable/4489465.
- [5] Black, Cheryl A., and H. Andrew Black. 2012. *Grammars for the People, by the People, Made Easier Using PAWS and XlingPaper*. In *Electronic Grammaticography*, edited by Sebastian Nordoff, 103–28. LD&C Special Publication 4. Honolulu, Hawai‘i: University of Hawai‘i Press. hdl.handle.net/10125/4532.
- [6] Black, H. Andrew. 2009. Writing Linguistic Papers in the Third Wave. *SIL Forum for Language Fieldwork* 2009 (004): 11 pages. www.sil.org/resources/publications/entry/7790.
- [7] Black, H. Andrew. 2017. *Why Learn to Use X_YLingPaper*. Dallas, Texas: SIL International. software.sil.org/downloads/r/xlingpaper/resources/documentation/WhyUseX_YLingPaper.pdf.
- [8] Brownie, John. 2013. Adverbs in the Mussau-Emira Verb Phrase. *Language & Linguistics in Melanesia* 31(1):1–11. www.langlxmelanesia.com/issues.
- [9] Buck, Marjorie J. 2018. *Gramática del amuzgo Xochistlahuaca, Guerrero*. (Serie de gramáticas de lenguas indígenas de México No16.) Tlalpan, Ciudad de México, México: Instituto Lingüístico de Verano, A.C. [SIL International in

- Mexico]. www.sil.org/resources/archives/75518.
- [10] Daniel, Marco, and Elke Schubert. 2013. *The mdframed Package: Auto-split Frame environment* version 1.9b.
- [11] Dimitriadis, Alexis. 2016. *TeX/LaTeX Information*. Web page. www.ling.upenn.edu/advice/latex.html.
- [12] Donnelly, Kevin. 2013. Representing Linguistic Pitch in X_YL^AT_EX. *TUGboat* 34 (2): 223–27. tug.org/TUGboat/tb34-2/tb107donnelly.pdf.
- [13] Ebarb, Kristopher J. 2014. *Tone and variation in Idakho and other Luhya varieties*. University of Indiana Ph.D. dissertation. pqdtopen.proquest.com/doc/1625743679.html?FMT=ABS.
- [14] Frampton, John. 2006. *Pst-Asr: Tex Macros for Typesetting Autosegmental Representations*. Version:1.1. CTAN. www.bakoma-tex.com/doc/generic/pst-asr/pst-asr-doc.pdf.
- [15] Frampton, John. 2012. *Expex for Linguists: Example Formatting, Glosses, and Reference*. Version: 4.1. mathserver.neu.edu/~ling/tex/expex/base/doc/expex-doc.pdf.
- [16] Freitag, Constantin and Antonio Machicao y Priemer. 2019. *LaTeX-Einführung Für Linguisten*. Berlin, Germany: Humboldt-Universität zu Berlin. doi.org/10.13140/RG.2.2.29299.27682.
- [17] Holton, Gary. 2021. *Writing Your Dissertation with L^AT_EX*. Typescript. Hawai'i. Github.com. [gmholton.github.io/files/DissertationWriting.pdf](https://github.com/gmholton/files/DissertationWriting.pdf).
- [18] Kew, Jonathan and Stephen McConnell. 1990. *Formatting Interlinear Text*. Occasional Publications in Academic Computing, Number 17. Dallas, Texas: Summer Institute of Linguistics.
- [19] Knuth, Donald Ervin. 1984. *The T_EXbook*. A. Computers & typesetting. Reading, Massachusetts: American Mathematical Society; Addison-Wesley.
- [20] Knuth, Donald Ervin. 1986. *T_EX: The Program*. B. Computers & typesetting. Reading, Massachusetts: Addison-Wesley.
- [21] Lamicela, Andrew Charles. 2020. *Distinguishing Passive from MP2-marked Middle in Koine Greek*. University of North Dakota M.A. thesis. commons.und.edu/theses/3277.
- [22] Liter, Adam. 2017. *L^AT_EX Workshop (for Linguists)*. adamliter.org/content/LaTeX/latex-workshop-for-linguists.pdf.
- [23] Lovell, Douglas. 1999. T_EXML: Typesetting XML with T_EX. *TUGboat*. 20 (3): 176–183. tug.org/TUGboat/tb20-3/tb64love.pdf.
- [24] Marlett, Stephen A. 2019. *Phonology From the Ground Up: The Basics*. Dallas, Texas: SIL International. www.sil.org/resources/archives/79207.
- [25] Nordhoff, Sebastian, and Stefan Müller. 2020. *Language Science Press Guidelines*. Berlin, Germany: Language Science Press. langsci.github.io/guidelines/latexguidelines/LangSci-guidelines.pdf.
- [26] Paterson III, Hugh J. 2021. *Language Archive Records: Interoperability of Referencing Practices and Metadata Models*. University of North Dakota M.A. thesis. commons.und.edu/theses/3937.
- [27] Peter, Steve. 2004. TEX and Linguistics. *TUGboat* 25 (1): 58–62. tug.org/TUGboat/tb25-1/peter.pdf.
- [28] Machicao y Priemer, Antonio, and Constantin Freitag. 2019. *LaTeX-Einführung Für Linguisten*. Presentation at the MGK Workshop – SFB 1412, Berlin. doi:www.linguistik.hu-berlin.de/de/staff/amp/latex20sfb/07-141-math2-trees-handout.pdf.
- [29] Rasmussen, Kent. 2018. *A Comparative Tone Analysis of Several Bantu D30 Languages (DR Congo)*. University of Texas Arlington Ph.D. dissertation. hdl.handle.net/10106/27483.
- [30] Rei, Fukui. 1996. TIPA: A System for Processing Phonetic Symbols in L^AT_EX. *TUGboat* 17 (2): 102–14. www.tug.org/TUGboat/tb17-2/tb51rei.pdf.
- [31] Smith, Zac, Todd Snider, and Mia Wiegand. 2016. *LaTeX and Linguistics - How to Make Your Research Pretty*. Presentation at the Cornell Linguistics Circle, Cornell, New York. conf.ling.cornell.edu/miawiegand/Latex_Slides.pdf.
- [32] Thiele, Christina. 1995. TEX and Linguistics. *TUGboat* 16 (1): 42–44. tug.org/TUGboat/tb16-1/tb46ling.pdf.
- [33] Thiele, Christina. 2007. *Christina Thiele Interview by Dave Walden for the T_EX Users Group*. Transcript. tug.org/interviews/thiele.html.
- [34] Unicode Consortium, ed. 1991. *The Unicode Standard: Worldwide Character Encoding*. Version 1.0. Reading, Massachusetts: Addison-Wesley. www.unicode.org/versions/Unicode1.0.0.

- [35] Ward, Monica. 2002. Reusable XML Technologies and the Development of Language Learning Materials. *ReCALL* 14 (2): 285–94. doi.org/10.1017/S0958344002000629.
- [36] Wilson, Peter. 2007. The Memoir Class. *TUGboat* 28 (2): 243–46. www.tug.org/TUGboat/tb28-2/tb89wilson.pdf.
- [37] Wilson, Peter. 2021. *The Memoir Class for Configurable Typesetting: User Guide*. version 3.70. Normandy Park, WA: The Herries Press. texdoc.org/serve/memoir/0.
- [38] Wood, Joyce Kathleen. 2012. *Valence-Increasing Strategies in Urim Syntax*. Graduate Institute of Applied Linguistics M.A. thesis. www.diu.edu/documents/theses/Wood_Joyce-thesis.pdf.
- [39] Beier, Christine, and Lev Michael. 2022. *Managing Lexicography Data: A Practical, Principled Approach Using FLEx (FieldWorks Language Explorer)*. In *The Open Handbook of Linguistic Data Management*, edited by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 301–14. Open Handbooks In Linguistics. Cambridge, Massachusetts: The MIT Press. doi.org/10.7551/mitpress/12200.003.0029.
- [40] Conzett, Philipp, and Koenraad De Smedt. 2022. *Guidance for Citing Linguistic Data*. In *The Open Handbook of Linguistic Data Management*, edited by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 143–55. Open Handbooks In Linguistics. Cambridge, Massachusetts: The MIT Press. doi.org/10.7551/mitpress/12200.003.0015.
- [41] Lehmann, Christian. 2004. *Interlinear morphemic glossing*. In *Morphologie: Ein internationales Handbuch zur Flexion und Wortbildung Morphology: an international handbook on inflection and word-formation*, edited by Geert E Booij, Christian Lehmann, Joachim Mugdan, and Stavros Skopeteas, 2:1834–57. Handbücher zur Sprach- und Kommunikationswissenschaft Handbooks of Linguistics and communication science 17. Berlin; New York: Walter de Gruyter. doi.org/10.1515/9783110172782.2.20.1834.
- [42] Schenner, Mathias, and Sebastian Nordhoff. 2016. *Extracting Interlinear Glossed Text from LATEX Documents*. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, edited by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, et al., 4044–48. Portorož, Slovenia: European Language Resources Association (ELRA). www.aclweb.org/anthology/L16-1638.pdf.

- ◇ H. Andrew Black
blackhandrew (at) gmail dot com
- ◇ Hugh J. Paterson III
i (at) hp3 dot me
<http://hp3.me>