

TUG 2012 — program and information

The main conference will take place in the Kennedy Room, lobby level.

Breakfasts (7–9 am) and lunches will take place in Parkers Lounge.

Sunday, July 15, 4–6 pm: *opening reception and registration*, Kennedy Room.

Monday, July 16: *concurrent L^AT_EX workshop*, Sue DeMeritt & Cheryl Ponchin, mezzanine level (room tba).

Monday July 16

8:00 am	<i>registration</i>	
9:00 am	Steve Peter, TUG	<i>Opening</i>
9:15 am	Amy Hendrickson, T _E Xnology Inc.	<i>The wonders of \csname</i>
9:55 am	<i>break</i>	
10:15 am	Frank Mittelbach, L ^A T _E X3 Project	<i>E-T_EX: Guidelines for future T_EX extensions, revisited</i>
11:15 am	Steve Peter	<i>Metafont as a design tool</i>
11:35 am	Will Robertson, University of Adelaide	<i>The lineage and progeny of fontspec and unicode-math</i>
12:15 pm	<i>lunch</i>	
1:20 pm	<i>group photo</i>	
1:30 pm	Bruno Delprat & Stepan Orevkov, INALCO & Université Paul Sabatier	<i>MayaPS: Typing Maya hieroglyphics with (L^A)T_EX</i>
2:15 pm	Sherif Mansour & Hossam Fahmy, Cairo University	<i>Experience with Arabic and LuaT_EX</i>
2:50 pm	<i>break</i>	
3:05 pm	Norbert Preining, JAIST	<i>Typesetting with Kanji — Japanese typography</i>
3:45 pm	Federico Garcia, Alia Musica Pittsburgh	<i>T_EX and music</i>
4:25 pm	David Walden, E. Sandwich, MA	<i>My Boston: Some printing and publishing history</i>
5:05 pm	q&a (after Herbert Schulz, Naperville, IL hours) Steve Peter, et al.	<i>Workshop: Introduction to TeXShop</i> <i>Workshop: Installing LuaT_EX</i>

Tuesday July 17

9:00 am	Troy Henderson, University of Mobile	<i>User-friendly web utilities for generating L^AT_EX output and MetaPost graphics</i>
9:40 am	Richard Koch, University of Oregon	<i>The MacT_EX install package</i>
10: am	Bill Cheswick	<i>An iT_EX update</i>
10:20 am	<i>break</i>	
10:40 am	Peter Flynn, University College Cork	<i>A university thesis class: Automation and its pitfalls</i>
11:20 am	David Latchman, T _E Xnical Designs	<i>Preparing your thesis in L^AT_EX</i>
12:00 pm	<i>lunch</i>	
1:00 pm	Boris Veytsman, George Mason Univ.	<i>T_EX and friends on a Pad</i>
1:40 pm	Bart Childs, College Station, TX	<i>L^AT_EX source from word processors</i>
2:20 pm	<i>break</i>	
2:40 pm	Federico Garcia	<i>Documentation in T_EXnicolor</i>
3:20 pm	Jim Hefferon & Michael Doob, Saint Michael's Coll. & Univ. of Manitoba	<i>Reaching for the stars with Asymptote</i>
4:00 pm	David Walden, moderator	<i>Roundtable discussion: T_EX consulting</i> Flynn, Hendrickson, Latchman, Thiele, Peter, Veytsman at Oceanaire (theoceanaire.com/Locations/Boston)
6–10 pm	<i>banquet</i>	

Wednesday July 18

9:00 am	Pavneet Arora, Bolton, Canada	<i>Sleep de(p)rived typesetting</i>
9:40 am	Bob Neveln & Bob Alps, Widener University & Evanston, IL	<i>Adapting ProofCheck to the author's needs</i>
10:20 am	<i>break</i>	
10:40 am	Christina Thiele, Carleton Prod. Ctr.	<i>Almost 30 years of using T_EX</i>
11:20 am	Will Robertson & Frank Mittelbach	<i>L^AT_EX3: From local to global — a brief history and recent developments</i>
12:00 pm	<i>lunch</i>	
1:00 pm	Norbert Preining	<i>T_EX Live 2012: Recent developments</i>
1:40 pm	Boris Veytsman & Leyla Akhmadeeva, GMU & Bashkir State Med. Univ.	<i>Towards evidence-based typography: First results</i>
2:20 pm	<i>break</i>	
2:40 pm	Didier Verna, EPITA	<i>Star T_EX, the Next Generation</i>
3:20 pm	TUG meeting; q&a	
≈ 4:00 pm	<i>end</i>	

Bart Childs

L^AT_EX source from word processors

Hennings' CTAN survey is a good starting point when considering projects implied by the title of this article. I find it a fair view of most related packages. He suggests having one of two goals: converting the *document structure* or converting the *appearance*. My goal is neither of these: it is producing L^AT_EX source that is clean and therefore maintainable. This is in keeping with Knuth's original goals in producing T_EX: graphic excellence and a document convenient for archiving. Structure and appearance are important but neither of these in the word processor format are as important as clean L^AT_EX to me. My current system is a hybrid. I use the common word processor exchange format — Rich Text Format, the OpenOffice Writer package and its `Writer2LaTeX` application, and macros for the Emacs editor written in Elisp. The test cases for this system are books on rotordynamics, a CS/1 programming text, a memoir on a friend's life including significant text fragments in the Czech language, and a novel that includes three love triangles. Even the worst case of significant mathematics formatting done in Word Perfect is tractable (I did not say easy). The most surprising problems are due to the limited skill of the users of their word processors.

Bruno Delprat & Stepan Orevkov

MayaPS: Typing Maya hieroglyphics with (L)T_EX

We present a system for the hieroglyphical composition of ancient Maya texts, to be used for their palaeography, the production of dictionaries, epigraphical articles and textbooks. It is designed on the base of T_EX and PostScript using the Dvips interface, and includes a set of Maya fonts.

The ancient Mayan writing system is very particular: base writing signs attach each other from all four sides (left, right, top, bottom), being also rotated and rescaled and could not be produced with usual T_EX's tools. For example, we can type: `\maya{1i.AM2 u.TUN/CHU uj.UJ.ki death.KIMI/1a}` to obtain the example shown in the preprint.

Peter Flynn

A university thesis class: Automation and its pitfalls

Despite the large collection of thesis classes available, there are always features that an institution needs which can better be met by writing Yet Another Thesis Class. There are also variations in the quality and availability of documentation, assumptions about preloaded packages, and the ease (or otherwise) with which the author can modify the layout.

In the case of UCC, the official requirements were very simple, avoiding the tendency to overspecify details found in some university formats.

The class was also required to be generally applicable to any discipline, so only a minimum of packages was needed (although this turned out to be more than anticipated).

The major component was an attempt to automate as much of the front matter as possible, based on options to tokenize the discipline and class of degree. This was done to avoid accidental omissions, variations, and misspellings in the titling (or even deliberate rewording); and to ensure that the relevant compulsory components appeared in the right place without the author having to do anything.

The result has been piloted with 20–30 PhD candidates for a year, and needs only a few final changes before release. Two other institutions in the state have already expressed an interest in basing their own thesis classes on this one.

Federico Garcia

Documentation in T_EXnicolor

My package `colordoc` builds on Frank Mittelbach's `docstrip` system of documentation, adding some utilities to use color in the code: matching delimiters (`{` and `}`) are colored the same, just as matching `\if`–`\fi` pairs. Commands are made red, bold, and italics, when they are being `\def`ined, just as variables when they are being declared (`\newcount`, `\newif`, etc.). These tools have certainly saved me a lot of time and trouble when editing or trying to understand a code. In the presentation I also describe the interesting general lines of the workings of both `doc` and `colordoc`.

Federico Garcia

T_EX and music

For some years I've been working on writing a professional music typesetting system with T_EX (and METAFONT). In 2005 I even had a grant from the T_EX Development Fund that allowed me to do a first model of the system. Since then the system has evolved, and particularly between 2010 and now I have actually developed a promising model.

This presentation does a little bit of history of the main idea of the T_EXmuse system, which as will be seen is entirely inspired by the T_EX 'spirit'. This touches on the potentially disastrous problems of WYSIWYM for music typesetting; on what similarly oriented systems have done (and not); and on whether even in spite of these problems the project is worth pursuing. The answer is in the affirmative, mainly because of the more disastrous problems of the alternative, i.e. commercial software. (In a nutshell, those problems are that those programs are *not* inspired by T_EX's spirit! In concrete, there are some achievements of T_EXmuse that would change the life of any composer.) I also demonstrate some pretty cool programming tricks

that I have found, both in \TeX and in METAFONT, that in my view speak to the beauty of the systems. The talk does not require technical knowledge of music or music typesetting.

Jim Hefferon & Michael Doob

Reaching for the stars with Asymptote

Asymptote is a stand-alone program that excels at generating line art, and takes its inspiration from Metafont and MetaPost. It is relatively recent, but already quite capable. I'll introduce some of its features for a person who has never seen it at work.

Troy Henderson

User-friendly web utilities for generating \LaTeX output and MetaPost graphics

There are several facets of the creation of \LaTeX documents and MetaPost graphics that deter users from initially trying both \LaTeX and MetaPost. These include the basic structure of the source files, the compilation of the source files, and the conversion of the output to a desired format. Furthermore, many \TeX users wish to create 2D and 3D graphs of functions for inclusion into their documents. Many of these graphs require considerable amounts of source code to create professional quality graphics, and this is yet another deterrent for those who might otherwise consider using MetaPost. This presentation will introduce several free web utilities that aim to eliminate each of these obstacles and describe the usage and methods of these utilities.

Amy Hendrickson

The wonders of $\backslash\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}\dots\backslash\text{e}\text{n}\text{d}\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$

A surprisingly useful tool, $\backslash\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}\dots\backslash\text{e}\text{n}\text{d}\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$, offers many opportunities for interesting and useful macros, whenever it is convenient to dynamically generate a series of definitions. When each definition contains a counter in its name, we can then call the definition using a loop that advances a counter, and then calling the definition using the loop counter inside the $\backslash\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}\dots\backslash\text{e}\text{n}\text{d}\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$.

A trivial use is for endnotes. More interestingly, $\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$ definitions can be used to send a set of definitions to the `.aux` file, where each new definition contains the current page number in its name, with a 'security level' number being defined. This allows the dynamic redefinition of the security level for a particular page, *within the .aux file* depending on whether the new number is higher or lower than the previous number. This can be used to determine the highest security level on any particular page. When the `.aux` file is then input, we can access the $\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$ definition in a running head of the \LaTeX document, calling $\backslash\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}\dots\backslash\text{e}\text{n}\text{d}\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$ control sequence and using the current page number within

$\backslash\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}\dots\backslash\text{e}\text{n}\text{d}\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$, to activate the definition of the highest security level on that page, and use the information to print the security level on top of the page.

Another interesting use is for on-line report generation, where a $\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$ definition can be used, for instance, to generate hyperlinks for financial analyses of stocks, in a report that compares hundreds of stocks, and then be able to automatically build a hyperlinked table of contents, using tabs built with `TikZ`.

Code and examples will be shown for each of these methods and uses of dynamically generated macros using $\backslash\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}\dots\backslash\text{e}\text{n}\text{d}\text{c}\text{s}\text{n}\text{a}\text{m}\text{e}$, and we'll look at some other ways this tool may be used as well.

Richard Koch

The Mac \TeX install package

Mac \TeX installs everything needed to run \TeX on a Macintosh with a single button click. I'll discuss the history of this package — Wendy's conspiratorial lunch and Jonathan Kew's all night coding session — modifications over the years, and important changes in the 2012 version. I'll discuss the importance of installing an unmodified, vanilla version of \TeX Live, explain how we add extra pieces so everything is automatically configured and ready to go, and end with possible future changes. The talk may contain completely unmotivated and irrelevant additional comments.

David Latchman

Preparing your thesis in \LaTeX

The submission of a thesis or dissertation is the culmination of many a graduate student's career. Given the time and effort toward research and attaining their degrees, this can often be a stressful time for many students. \LaTeX offers the advantage of separating form from content and as the typical university thesis class can take care of a university's formatting requirements thus makes a student's life easier — well, at least it is supposed to.

Unfortunately, some formatting 'blends' into the content, thereby adding to the stress of an already unpleasant task. But there is some light at the end of the tunnel. With some preparation, typesetting a thesis in \LaTeX can be relatively pain free. But it's not simply a matter of just knowing what packages but *how* to use them and what is needed to use them effectively. Topics covered will include the typesetting of equations — both mathematical and chemical — as well as the proper formatting of tables and bibliographies.

Sherif Mansour & Hossam Fahmy

Experience with Arabic and Lua \TeX

This is an experience report of an attempt to include the AlQalam font for Arabic script within

LuaTeX. We describe the problems we faced trying to figure out how to use a new right-to-left font within LuaTeX. We also describe how to call the many different shapes that are defined via parameters in the original font. We also present some ideas to modify the line breaking algorithm of TeX to allow the use of different shapes for the same character in order to justify the line. This is still work in progress.

Frank Mittelbach

E-TeX: Guidelines for future TeX extensions, revisited

Shortly after Don Knuth announced TeX 3.0 I gave a paper analyzing TeX's abilities as a typesetting engine. The abstract back then said:

Now it is time, after ten years' experience, to step back and consider whether or not TeX 3.0 is an adequate answer to the typesetting requirements of the nineties.

Output produced by TeX has higher standards than output generated automatically by most other typesetting systems. Therefore, in this paper we will focus on the quality standards set by typographers for hand-typeset documents and ask to what extent they are achieved by TeX. Limitations of TeX's algorithms are analyzed; and missing features as well as new concepts are outlined.

Now — two decades later — it is time to take another look and see what has been achieved since then, and perhaps more importantly, what can be achieved now with computer power having multiplied by a huge factor and last not least by the arrival of a number of successors to TeX which have lifted some of the limitations identified back then.

Bob Neveln & Bob Alps

Adapting ProofCheck to the author's needs

ProofCheck is a system for writing and checking mathematical proofs. Theorems and proofs are contained in a plain TeX or LaTeX document. Parsing and proof checking are accomplished through Python programs which read the source file. Although the use of these programs has never been restricted to any particular logical or mathematical language, the work required to actually implement an author's choices in these matters, especially in the logic, and to make the necessary modifications of the supporting files has been sufficiently laborious as to pose an obstacle to the use of ProofCheck. This paper describes updates to the system whose purpose is to alleviate these labors to the extent possible so as to facilitate the use of ProofCheck in a logical and linguistic setting of the author's choice.

Steve Peter

Metafont as a design tool

Well-written Metafont sources provide a font designer with a nearly unparalleled tool to explore variations on a typographic theme. Paired with TeX in an advanced environment, the designer can explore serif structure, bracketing, weight variations and more in the context in which the font will be used: real textual matter. I'm going to ignore the production problems inherent to Metafont (not to mention the various possible solutions) to concentrate on the design aspects of this amazing tool.

Norbert Preining

Typesetting with Kanji — Japanese typography

Japanese typography is very particular and demanding in several respects: four different writing systems: Kanji, Hiragana, Katakana, Roman letters mixed together; vertical and horizontal typesetting; traditional grid layout versus mixture of writing systems. This all led to a spin-off TeX implementation called "Publishing TeX" (pTeX) that can deal with these specifics.

Until 2011 there was an independent distribution of TeX for Japanese users, first based on teTeX, later on TeX Live (`ptetex`, `ptexlive`). TeX Live 2011 and 2012 introduced all of the necessary tools and features and we hope that with TeX Live 2012 the need for a special setup for Japanese users is past.

In this talk we give an overview of the specialities of Japanese typography, presenting the difficulties met in modern texts. Continuing, we present solution provided by TeX Live to some of these problems, and discuss further development.

Norbert Preining

TeX Live 2012: Recent developments

TeX Live will be released in early summer 2012 and brings a couple changes that have been in the works for a long time: a "multi-updmap" that reads several `updmap.cfg` files, and multi-repository support for the TeX Live Manager `tlmgr`.

`updmap` is a program that generates the necessary configuration files for `dvips`, `dvipdfm(x)`, `pdftex`, and `pxdvi` to display PostScript Type 1 fonts. It reads a configuration file that lists several map files, and combines all the font definitions from these map files. Until now local font maps had to be integrated into this `updmap.cfg` file, and so could easily be overwritten or otherwise be lost.

The new implementation has a long history. The original Perl version was written by Fabrice Popineau for Windows, later extended by Reinhard Kotucha and Karl Berry and used starting last year on all platforms supported by TeX Live. The

code has now been extended to deal with multiple configuration files in a transparent way.

This allows a clear separation of `updmap.cfg` file parts. One `updmap.cfg` file now can (but does not have to) only provide information about the `texmf` tree it resides in. In other words, fonts installed into, for example, the `TEXMFLOCAL` tree can be activated by an entry in the `updmap.cfg` file *in this tree*.

We will discuss this new functionality and provide usage examples and advise on transition from the old system.

The other big change in \TeX Live this year is the extension of the \TeX Live Manager with the capacity of reading multiple repositories. In recent years, a few alternative \TeX Live repositories have come into existence with a wide range of usage patterns: distribution of local packages (Japanese \TeX related packages in `tlptexlive`, Korean \TeX User Group repository), \TeX Live infrastructure testing (in `tlcritical`), provision of development and nonfree packages (in `tlcontrib`), etc.

Until now a user had to go through all desired repositories one by one passing the necessary parameters for each in turn. The new `tlmgr` supports use of several sources at the same time. The selection of packages appearing in multiple repositories is done by “pinning” packages to a repository.

We will present this new functionality, give usage examples, and a guided tour through setting up and using this new feature.

We will close with an overview on other changes in \TeX Live 2012.

Will Robertson & Frank Mittelbach

L^AT_EX3: From local to global — a brief history and recent developments

The original source code for $L^A T_E X3$ dates to the early 1990s. Key aspects of its development occurred during that decade, but it was not until the late 2000s that the project began delivering code that was widely used by mainstream $L^A T_E X$ users. What happened in this time? This talk will discuss how $L^A T_E X3$ development evolved over the decades and how it reached a state of being used to produce real users’ documents whether or not they are actually aware of it. $L^A T_E X3$ can be thought to consist of separate ‘layers’, and the programming layer known as `expl3` is starting to be used to solve problems in and write packages for $L^A T_E X2_\epsilon$. Our plans are not restricted to such ‘under-the-hood’ measures, however, and we have discussed layers of $L^A T_E X3$ that will have more visibility at the user interface. Our talk will discuss these separate layers and where our plans lead in the future, and will conclude with a demonstration of what’s new in the current code.

Will Robertson

The lineage and progeny of fontspec and unicode-math

My first $L^A T_E X$ package, `fontspec`, was written in 2004 before I knew how to program in $L^A T_E X$ and in truth before I knew how to program at all. This trial-by-fire introduced me to the lovely world of \TeX programming and after some time I ended up writing a smattering of other works. (All the while actually starting to learn what this whole ‘programming’ thing was all about, including how to please and displease people who were just trying to get work done, thank you very much.) Some time later I foolishly tried ‘planning’ an ambitious new package, `unicode-math`, that took significantly longer to release. In the course of writing that package I learned really just how little I actually knew, and as a side-effect somehow ended up helping to write code for the $L^A T_E X3$ project. In this talk I will talk about the motivation for writing these two packages, discuss recent developments with them, and finally touch on how $L^A T_E X3$ influenced their development.

Herbert Schulz

Workshop: Introduction to TeXShop

A workshop introducing some of the more obscure and less used features of `TeXShop` for users who wish to become more proficient in its use to produce $L^A T_E X$ documents.

Christina Thiele

Almost 30 years of using T_EX

It’s not just \TeX that’s gotten older and more seasoned . . . Reflections on changes in `TeX` and friends as used in a small typesetting company: software and hardware, of course, but also procedures and skills, resources that went from zero to virtually infinite, all of it interwoven with life and personal change. It’s not earth-shaking news, but we’ve come far enough that looking back yields some interesting comparisons.

Didier Verna

Star T_EX, the Next Generation

In 2010, I asked Donald Knuth why he chose to design and implement \TeX as a macro-expansion system (as opposed to more traditional procedure calls). His answer was that: 1) he wanted something relatively simple for his secretary who was not a computer scientist; 2) the very limited computing resources at that time practically mandated the use of something much lighter than a true programming language.

The first part of the answer left me with a slight feeling of skepticism. It remains to be seen that \TeX is simple to use, and when or where it is, its underlying implementation has hardly anything to do with it.

The second part of the answer, on the other hand, was both very convincing and arguably now obsolete as well. Time has passed and the situation today is very different from what it was 30 years ago. The available computing power has grown exponentially, and so has our overall skills in language design and implementation.

Several ideas on how to modernize T_EX already exist. Some have been actually implemented. In this talk, I will present mine. Interestingly enough, it seems to me that modernizing T_EX can start with grounding it in an old yet very modern programming language: Common Lisp. I will present the key features that make this language particularly well suited to the task, emphasizing on points such as extensibility, scriptability and multi-paradigm programming. The presentation will include reflections about the software engineering aspects (internals), as well as about the surface layer of T_EX itself. Most notably, I will explore the possibilities of providing a more consistent syntax to the T_EX API, while maintaining backward compatibility with the existing code base.

Boris Veytsman

T_EX and friends on a Pad

T_EX on an Eee Pad is quite workable.

Boris Veytsman & Leyla Akhmadeeva

Towards evidence-based typography: First results

At the previous TUG meeting we reported experi-

mental design for checking whether the typographic features of the text (fonts, page layout, justification, etc.) influence the way readers comprehend and remember the contents. Our study is intended primarily for the designers of textbooks, where the comprehension of the text is very important.

In this work we report the first results of our study. It seems that despite the beliefs of typographers, the text comprehension and the speed of reading is not much influenced by typography. These findings confirm the generalized ecological hypothesis by Legge and Bigelow. It seems the human brains are flexible enough to allow us to read even badly designed pages.

We also discuss the role of T_EX as a useful tool to create various controlled page designs for typographic study.

David Walden

My Boston: Some printing and publishing history

Boston, where I have lived for nearly 50 years, has an important publishing and printing history. Therefore, for TUG 2012 I have used various library and other Boston resources to learn more about the printing and publishing history of the city. This presentation sketches what I have learned about several eras of Boston printing and publishing: 1) Colonial period, 1630–1775; 2) Revolutionary War (1775–1783) and transition; 3) Literary culture of the mid-19th century; 4) Later 1800s to mid-1900s; 5) Personal observations, 1964–present.


MayaPS: Typing Maya Hieroglyphics with T_EX/L^AT_EX

Bruno Delprat and Stepan Orevkov

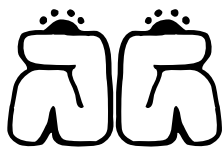
Abstract

We present a system for the hieroglyphical composition of ancient Maya texts, to be used for their palaeography, the production of dictionaries, epigraphical articles and textbooks. It is designed on the base of T_EX and PostScript using the Dvips interface, and includes a set of Maya fonts.


The ancient Maya's writing system is very particular: base writing signs attach each other from all four sides (left, right, top, bottom), being also rotated and rescaled and could not be produced with usual T_EX's tools. For example, we type:


```
\maya{li.AM2 u.TUN/CHU uj.UJ.ki death.KIMI/1a}
to obtain  (Dresden codex)
```

1 Introduction



The package MayaPS is designed for editing the palaeography of ancient Maya hieroglyphical texts using T_EX or L^AT_EX and Dvips. The PhD dissertation [1] and a previously published Spanish language symposium communication [2] are typeset using it. MayaPS is available on <http://picard.ups-tlse.fr/~orevkov>.

To get the above maya word  *xib* (male), we typed: `\mayaSize{2cm}\maya{422.422}`.

and to get:  *katun* (calendar cycle of 20 years), we input: `\maya{(023.153.023):220} \emph{katun} (calen...`

2 Structure of the Ancient Maya Script

2.1 General Principles

The ancient Maya logo-syllabic writing has been in use in Central America's Southern Mexico, Guatemala, Belize, Honduras and Costa Rica for more than 1300 years, from the 3rd century AD to the mid 16th century, when Spaniards forbid its use and burned Maya books on religious and political grounds.









This impressive civilization left rich inscriptions on monuments, ceramics and divinatory almanacs. They constitute nevertheless a small volume of avail-

able texts: three surviving manuscripts (the Dresden, Madrid and Paris codexes) and about a thousand short inscriptions. Maya texts are now largely deciphered, with a variable degree of reliability.

The writing system comprises more than 500 base signs called glyphs. Since the end of the 19th century, Western scholars set up catalogues of Maya hieroglyphics with different encoding numbers, the most popular being the Thompson Catalogue [11].



Figure 1: Dresden codex page 30b(2) & text palaeography with translation below

			
400/010.030	+176/204.031	117.260	133/111.023
<i>tse</i> -ah	<i>lakin</i>	<i>chac-rib</i>	<i>kabil</i>
Was stand- ing	East	red man	sweet
			
423/515	530.112	515/504.013	026.401
<i>cehel-uah</i> ;	<i>Chac</i>	<i>hanal</i>	<i>u-bool</i>
deer tamal	god Chac	meal	its tribute



Ancient Maya words are composed by attaching together primitive (non decomposable) glyphs, in a quite similar way as Chinese characters are composed. Composed primitive glyphs are re-scaled so that they harmoniously fill a rectangle of a fixed size (called *cartouche* by mayanists). The cartouches are

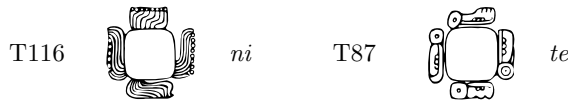
placed in a regular way on a page. Maya manuscripts texts are organized in blocks of 2 to 16 cartouches which constitute as many sentences, often followed by associated numbers, dates and eventually a picture. According to the number of cartouche space available on the almanac page to write a short or long sentence, the scribe would squeeze in or spread out writing signs among the cartouches to avoid empty boxes and obtain a nice looking page layout.



2.2 Glyph types and orientations

In ancient Maya writing system, there are two types of primitive glyphs called *central elements* and *affixes*. Usually the shape of central elements is closer to square whereas affixes are narrower.

Central elements always appear in the same orientation but affixes turn so that they stick to other glyphs by their long side, following a general orientation rule.


Complete glyphic cartouches are made of 1 to 5 basic signs or glyphs. Thompson [11] has shown that affixes, like  *ni*, present rotation patterns and symmetries, around a central element, such as  *KIN*, which orientation is fixed.




Through the analysis of the Maya codices, we determined that affix patterns follow a determined rule. For example,  *te* (tree) is an affix and it usually attaches to a central element like this: . So, there are five standard orientations for each affix: when it is single and when it attaches from the left, from the right etc.

In the Dresden Codex we find the following corresponding cartouche compositions:













Affixes are generally syllabic value signs which can combine together or with a central element to write a Maya word. Central elements are generally of logographic nature: corresponding to a morpheme or a word, and read globally for example  *KIN* (sun, day).

A complete glyphic cartouche often corresponds to a lexical entry with preceding and following grammatical affixes as in : *KIN-ni* (sun, day), but it can also in some cases correspond to two words if



they are short, or more rarely otherwise to a part of an expression spelt over two cartouches.

2.3 Glyph composition into cartouches

For the composition of glyphs the following standard notation is used in the historical and linguistic literature on ancient Maya. Namely, if  and  are two glyphs (primitive or not), then *A.B* and *A:B* encode the glyphs  and , for example: 204.031  and 204:031 . To control the order of composition, one can use parentheses in the same way as in mathematical formulas. For example, both *A.B:C* and *A.(B:C)* stand for  but *(A.B):C* stands for , for example: 026.172/023  and (154.123)/306  (in glyph codes, ‘/’ means the same as ‘.’).


3 Description of MayaPS

3.1 Main Features

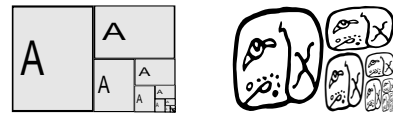
The text cartouches  and  that we used above are composed of primitive glyphs:



MayaPS does not care for any grammatical function or linguistic meaning of primitive glyphs. They are just graphical elements which are elementary bricks of Maya typesetting, like letters for European languages.

As it should be clear already, each primitive glyph is referred to by its code (called further the *glyph code*). So, the glyph codes used above are “422”, “005”, “023”, “153”, and “220”. In general, a glyph code is any sequence of digits 0...9 and letters a...z, A...Z. The encoding system is rather flexible. For example, after the command `\mayaDefine{A9z}{442}` you can type `maya{A9z}` to get .

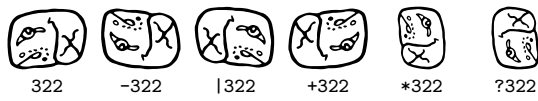
Any formula of this kind is admitted by MayaPS, even something like this:



The picture on the right hand side (a fantasy writing cartouche, as there is no such glyph in ancient Maya writing) is printed by the command







```
\maya{322.322:(322.322:(322.322:(322.322:(322.322:(322.322))))}}
```


The type of each glyph (affix/central) and the five default orientations for each affix are written in the font file. Orientation can be changed with *modifiers* -|+*? whose meaning is:

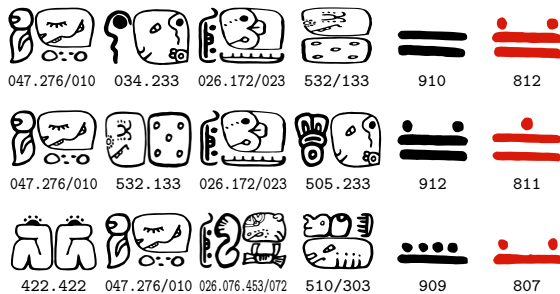


The modifiers can be composed together yielding:



Let us discuss again the glyph . We see that the primitive glyph 023 occurs here in two different ways:  and . Moreover, in  it occurs twice horizontally. MayaPS automatically chooses the orientation of each primitive glyph of the affix type according to “orientation rules” formulated by the first author after a careful analysis of ancient manuscripts. Of course, these rules have exceptions. It is very easy to handle them. For example, if you type `\maya{422.222/024}`, you obtain  (the default orientation), but if you type `\maya{-422.410}`, you obtain .

A more representative example is the palaeography of page 22c of the Dresden Codex, due to the first author:



To obtain it, we typed:

```
\mayaC{ % \mayaC = glyphs with input codes
047.276/010 034.233 026.172/023 532/133 910 812
047.276/010 532.133 026.172/023 505.233 912 811
422.422 047.276/010 026.076.453/072 510/303 909
```



MayaPS permits the support of multiple fonts. In this paper we use mostly the font `codex` created using the tools mentioned in §4.1, but another style glyph set `gates` has been implemented based on the same glyph codes, for example:








MayaPS provides a tool to add or replace glyphs in existing fonts and a tool for making new fonts.



3.2 Substitutions (ligatures)







To each Maya font is associated a list of substitutions. As we told in Introduction, a new substitution $s_1 \rightarrow s_2$ can be defined by the command `\mayaDefines s_1s_2` where s_1 and s_2 are any strings (chains of characters). Substitutions are applied to the arguments of glyph drawing macros. They are applied non-recursively. Some substitutions can be predefined in a Maya font. Three types of substitutions are predefined in the font ‘`codex`’:

(1). *Ligatures*. One or several affixes or central elements can be melted inside a central element or, more frequently, inside a head figurative element instead of being simply attached to it, forming a *ligature* as a single bound form. For example, when you type `\maya{070:349}`, you obtain  353 rather than  070:349 because of the predefined substitution (ligature) $070:349 \rightarrow 353$. Here we typed `\maya{070:(349)}` to print the non ligatured form. The ligature was not applied because ‘070:349’ is not a substring of ‘070:(349)’.

As another frequent ligature example you have: 373  *Cacau* D7c (2), that decomposes into simpler glyphs: $369 < 023 / 023 >$. The operator `< >` indicates that both affixes 023  are placed in the centre of 369 .

Within Maya texts, both forms: melted as a ligature (single glyph code), and separately drawn (2 - 3 glyph codes) are equivalent and may constitute orthographical variants, as for:  and . Our catalogue includes around 100 ligature glyphs.

(2). *Thompson codes*. The basic glyph codes in the font ‘`codex`’ are based on glyph numbers of the Evreinov catalogue [3], from which font drawings were derived. However, many specialists are more familiar with glyph codes in Thompson’s catalogue [11]. Due to predefined substitutions $T1 \rightarrow 026$ , $T2 \rightarrow 410$  etc., those codes can be used also for text input.

(3). *Phonetic values*. Reconstituted phonetic values in the Maya language can also be used to ease text input. Phonetic values of affixes are conventionally written in small case letters: $a \rightarrow 050$ , $aj \rightarrow 044$ , $ak \rightarrow 506$  etc.; for central elements block letters are used: $AT \rightarrow 200$ , $BA \rightarrow 213$ , $BA2 \rightarrow 212$ .

Like for Thompson codes, predefined substitution tables permit the use of multiple character input methods, as for Chinese characters computer input with: *pinyin* (PRC's official romanization), *cangjie* (decomposition into graphic keys) or *dianbaoma* (Chinese telegraph codes).

4 Maya fonts

4.1 Font Creation Mechanism

A MayaPS font is an ASCII text file with the extension `.mpf`. Its structure is rather flexible. It is described in detail in [8]. It has several sections of PostScript code (a header and glyph definitions)[6] separated by lines starting with `%@`. TeX macros use these marks to select needed sections for including them into `'mayaps.tmp'` (see §7.1). Substitution rules (see §4.2) have the form `%L@ s1 s2`.

There is a tool (involving a special vectorizer 'cotrace') for creating MayaPS fonts out of monochrome bitmaps. The fonts supplied with MayaPS are made with it.

When MayaPS fonts created with this tool are used, they generate Type 1 fonts [5] in the resulting `ps` file. As Type 1 Maya fonts are used, the resulting `pdf` document after conversion is considerably smaller than the intermediate `ps` file. Only definitions for font signs used in the text are included, and just once.

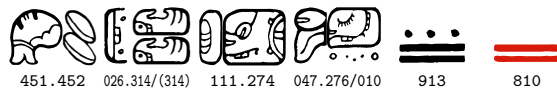
The `\mayaAddGlyph` macro allows to include a new glyph from an `eps` file but we do not recommend to use it too much because it increases very fast the resulting `ps` and, especially, `pdf` file.

4.2 Current Available Fonts

3 extensive Maya script fonts and a partial Olmec script font have been produced so far by the first author.

The font 'codex' was designed mostly from drawings of the Evreinov glyph catalogue [3] and is the one used up to now in this article.

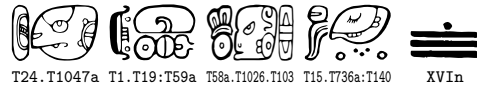
The font 'gates' is derived from the lead cast font designed by William Gates in the 30's for his book [4], has been implemented based on the same glyph codes as for font 'codex'. For example, the beginning of the above quote from the Dresden Codex printed in the font 'gates' looks as:



(we typed `\gates` and then, just copied the above same codes as for a text in the 'codex' style).

The font 'thompson' is made out from the Thompson catalogue [11] drawings, where original Txxx codes from the catalogue are implemented, showing that glyph codes in different fonts can be independent.

A correspondence ligature table, included in the font, permits glyph input using Evreinov derived codes in a text displayed in 'thompson' font, as is shown below:



Similar correspondence ligature tables are included in font 'codex' and 'gates' for glyph input using Thompson codes and phonetic reconstituted values, for example:



The font 'olmeca' is derived directly from drawings of *La Mojarra* stella [7] of the 2nd century A.D. and from the *Tuxtla* tablet. It represents a partial set of Olmec glyphs, which were composed in vertical texts without rotation of affixes.



5 Principles we tried to follow

5.1 Length optimization

Suppose, you have already a `ps` file (produced by TeX/Dvips [9]) in an alphabetic language and you include ancient Maya glyphs into it. Then MayaPS adds to `ps` only:

- MayaPS header (7 Kb);
- definitions of primitive glyphs (0.5 – 3 Kb per glyph for 'codex');
- about 60 bytes for each occurrence of each composed glyph.

MayaPS includes the definitions of only those primitive glyphs which are effectively used in the text. Each definition is included only once even if the primitive glyph is used many times. This property holds after conversion `ps` to `pdf`, because Type 1 fonts are used for primitive glyph (as usually, the size of `pdf` is between the sizes of `ps.gz` and `ps`).


5.2 Simplicity of installation and no need of support

To use MayaPS, it is enough to copy few files into any directory (folder) ‘visible’ by \TeX , for example, the directory where the `tex` file is. In particular, no extra font in the usual sense is needed (a typical beginner’s problem is how to make \TeX to ‘see’ a new font).

The algorithm to draw a composed glyph is implemented in the PostScript language [6], and Dvips [9] is used for calling it from a \TeX file via `\special` macro (this is why MayaPS does not work with pdf \TeX). So, \TeX , Dvips, and PostScript are needed. Nothing else is used in MayaPS. The only exception is the tool for creating new MayaPS fonts (`mpf` files) where C-programs are used, but the font file format is described and it is easy to make an `mpf` file out of a Type 1 font (a detailed instruction is given in reference [8]).

6 A few words about the implementation

6.1 Interaction \TeX /PS

A glyph code (example: 111.+176/111 for ) is passed to `ps` file by the Dvips command:

```
\special{"M(111.+176/111) w h d E}
```

where $w \times h$ is the cartouche size and d is the font descriptor (an integer number). Dvips literally includes the argument of `\special{" }` into `ps` file and the task of drawing the composed glyph is delegated to a PostScript interpreter. The glyph drawing subroutine `E` is defined in the header included to `ps` file by Dvips command:

```
\special\{header:mayaps.tmp}
(see [9]; §5 for more detail)
```

Before issuing the command `\special{"M...E}`, all primitive glyph names are extracted from the glyph code and it is checked if their definitions are already included into the header `mayaps.tmp`. The token list `\output` is appended so that at the end of each page, the definitions of all newly appeared glyphs are copied from `mpf` files to `mayaps.tmp`.

6.2 Substitution mechanism

In earlier versions of MayaPS, the substitution mechanism was implemented by creating for each substitution $s_1 \rightarrow s_2$ a macro whose name (calling sequence) contains s_1 and whose expansion is s_2 .

Then, for each substring of each composed glyph, it was checked by the command:

```
\ifx\curname ... \endcurname\relax
```

if the corresponding macro is defined. However, this command leaves the tested calling sequence in \TeX ’s

the memory forever. As a result, \TeX ’s capacity (60000 calling sequences) had been exceeded when the thesis [1] over-passed 300 pages.

The new substitution mechanism creates the tree of initial subwords of left hand sides of all substitutions. Now the number of used calling sequences does not exceed the size of this tree.

Acknowledgments. The idea to use PostScript language rather than \TeX for drawing composed glyphs belongs to Ilya Zakharevich. The \TeX part of MayaPS is inspired by `epsf.tex` (by Tom Rokicki) and even some code is taken from there. Our glyph numbering system is adapted from the Evreinov catalogue [3], as are most `codex` font drawings. Glyph drawings for the `gates` font are taken without modification from William Gates’ [4] Dresden Codex palaeography.

Another attempt to adapt \TeX / \LaTeX for ancient Native-American languages that concerned Olmec writing was done in [10], using a very different approach from ours.

References

- [1] B. Delprat. *Le codex de Dresde: Paléographie et traduction comparée d’un almanach maya du 15e siècle*. Thèse de doctorat, Institut National des Langues et Civilisations Orientales, Paris, n.p. (thèse en cours).
- [2] B. Delprat and S. Orevkov. maya \TeX – un sistema de composición tipográfica de textos jeroglíficos mayas para la computadora. In *XXI Simposio de investigaciones arqueológicas en Guatemala*, Guatemala de la Asunción, 23-27 July 2007.
- [3] Yu. G. Kosarev E. V. Evreinov and V. A. Ustinov. *Primenenie elektronikh vychislitel’nykh mashin v issledovanii pis’mennosti drevnikh maya [The Use of Electronic Computing Machines Applied to Research on Ancient Maya Writing]*. Akademia nauk SSSR [Academy of Sciences of the USSR], Novosibirsk, 1969. 4 vol.
- [4] William E. Gates. The dresden codex reproduced from the tracings of the original colorings and finished by hand. In *Maya Society Publication*, number 2. The Maya Society at the Johns Hopkins University, Baltimore, 1932.
- [5] Adobe System Inc. *Adobe Type 1 Font Format*. File `T1Format.pdf` available on <http://www.adobe.com>.
- [6] Adobe System Inc. *PostScript Language Reference Manual*. Files `plrm.pdf` and `plrm2.pdf` available on <http://www.adobe.com>.

- [7] Martha J. Macri and Laura M. Stark. *A Sign Catalog of the La Mojarra Script*. Pre-Columbian Art Research Institute, San Francisco, 1993.
- [8] Stepan Orevkov. *MayaPS: Typing Maya with TeX/LaTeX. Reference manual*. available on <http://picard.ups-tlse.fr/~orevkov>.
- [9] T. Rokicki. *Dvips: A DVI-to-PostScript Translator*. File `dvips.pdf` included in the most of T_EX distributions, available on <http://www.ctan.org>.
- [10] A. Syropoulos. Typesetting native american languages. *Journal of Electronic Publishing*, 8(1), 2002. <http://www.press.umich.edu/jep>.
- [11] J. E. S. Thompson. *A Catalogue of Maya Hieroglyphs*. Univ. Oklahoma Press, 1962.

◇ Bruno Delprat
 INALCO & SeDyL-CNRS, 7, rue
 Guy Môquet
 94801 Villejuif cedex
 France
brunodelprat@club-internet.fr
<http://celia.cnrs.fr/Fr/Labo/Delprat.htm>

◇ Stepan Orevkov
 Institut de mathématiques de
 Toulouse, Université Paul
 Sabatier
 31062 Toulouse
 France
orevkov@math.ups-tlse.fr
<http://www.math.univ-toulouse.fr/~orevkov/mayaps.html>

A university thesis class: automation and its pitfalls

Peter Flynn

Abstract

Despite the large collection of thesis classes available, there are always features that an institution needs which can better be met by writing Yet Another Thesis Class. There are also variations in the quality and availability of documentation, assumptions about preloaded packages, and the ease (or otherwise) with which the author can modify the layout.

In the case of UCC, the official requirements were very simple, avoiding the tendency to overspecify details found in some university formats. The class was also required to be generally applicable to any discipline, so only a minimum of packages was needed (although this turned out to be more than anticipated).

The major component was an attempt to automate as much of the front matter as possible, based on options to tokenize the discipline and class of degree. This was done to avoid accidental omissions, variations, and misspellings in the titling (or even deliberate rewording); and to ensure that the relevant compulsory components appeared in the right place without the author having to do anything.

The result has been piloted with 20–30 PhD candidates for a year, and needs only a few final changes before release. Two other institutions in the state have already expressed an interest in basing their own thesis classes on this one.

A summary of some of the points covered here was published in a recent *TUGboat* [2].

1 Yet Another Thesis Class

A recent retrieval from CTAN¹ shows that there are currently 42 thesis or thesis-related packages available for L^AT_EX (see Figure 1). These are almost all institution-specific, and implement a wide variety of rules and restrictions which fall into five broad groups:

- title page metadata;
- sequencing of preliminary pages;
- wording of formal declarations;
- formatting and layout;
- markup abbreviation and shortcuts.

The level of detail required by each institution varies so widely that using a thesis class from else-

¹ Scripted with the assistance of *dog*, *tidy*, and *l_xprintf*, thanks to the robustness of the directory and link structure implemented by the CTAN team.

adfathesis Australian Defence Force Academy thesis format.	thuthesis Thesis template for Tsinghua University.
afthesis Air Force Institute of Technology thesis class.	uafthesis Document class for theses at University of Alaska Fairbanks.
beamer2thesis Thesis presentations using beamer.	ucdavisthesis A thesis/dissertation class for University of California Davis.
classicthesis A 'classically styled' thesis package.	ucthesis University of California thesis format.
ebsthesis Typesetting theses for economics.	ucthesis209 L ^A T _E X 2.09 document style for UC Theses.
elteikthesis Thesis class for ELTE University Informatics wing.	uhthesis University of Houston thesis document style.
fbithesis Computer Science thesis class for University of Dortmund.	uiucthesis UIUC thesis class.
gatech-thesis Georgia Institute of Technology thesis class.	umich-thesis University of Michigan Thesis L ^A T _E X class.
hepthesis A class for academic reports, especially PhD theses.	umthesis Dissertations at the University of Michigan.
jasthesis A 'standard' thesis class.	unamthesis Style for Universidad Nacional Autónoma de México theses.
jkthesis Document class for formatting a thesis.	unswthesis UNSW theses.
msu-thesis Class for Michigan State University Master's and PhD theses.	uothesis Class for dissertations and theses at the University of Oregon.
muthesis Classes for University of Manchester Dept of Computer Science.	uowthesis Document class for dissertations at the University of Wollongong.
pitthesis Document class for University of Pittsburgh theses.	uscthesis USC thesis style for L ^A T _E X 2.09.
psu-thesis Package for writing a thesis at Penn State University.	utorontothesis A thesis class definition for University of Toronto.
ryethesis Class for Ryerson University Graduate School requirements.	utthesis Thesis package for the University of Texas at Austin.
sapthesis Typeset theses for Sapienza University, Rome.	ut-thesis University of Toronto thesis style.
seuthesis L ^A T _E X template for theses at Southeastern University.	uwthesis University of Washington thesis class.
suthesis Typeset a Stanford University thesis.	uwthesis209 L ^A T _E X 2.09 style for University of Washington theses.
thesis Typeset thesis.	withesis University of Wisconsin-Madison Thesis L ^A T _E X Class.
thesis-titlepage-fhac Little style to create a standard titlepage for diploma thesis.	york-thesis A thesis class file for York University, Toronto.

Figure 1: Thesis and thesis-related packages available from CTAN as of May 2012

where usually means some re-configuration and re-programming, which may be beyond the skills of the author. In some cases there is extensive documentation and an example thesis document; in others there is very light formatting and specification, and authors are left to modify the document as they see fit.

In the present case, there were other requirements which led to the decision to write a local thesis class rather than modify an existing one:

- the class had to be usable by all disciplines, not just those in which L^AT_EX has traditionally been used the most;
- it had to automate (where possible) those areas where the author would not in any case have a choice;
- it had to allow for the writing of a thesis in the Irish language.

Some guidance was available in the form of the Thesis Requirements published by the university. I am indebted to the staff of the Registrar's Office, the Graduate Studies Office, and the Boole Library; to the individual colleges, departments, faculties, and schools; and to students and other users for all their comments and suggestions.

2 Building the thesis class

We took an early decision to base the class on the standard `report` class because it appeared to be the one most familiar to existing users for writing theses.² This also meant we could adopt (or in some cases, prohibit) existing class options.

Among the suggestions we received in feedback from users (when we discussed developing a thesis class), was to keep it simple and make it obvious. We interpreted this to mean that we should as far as possible keep the existing meanings for existing commands and environments, and not introduce new ones which were not easily memorable.

Other suggestions included adding optional arguments to certain commands for features which were very frequently used, as repeatedly having to specify something manually was seen as part of the tedium of using L^AT_EX by users who lacked the programming skills needed to write their own macros.

2.1 Title page metadata

This is often technically one of the simplest parts of a document class design, but it has an impact out of proportion to its position. It is, after all, just

² Users later reported the most compelling reason was that the `report` class supports the `abstract` environment, which the `book` class does not.

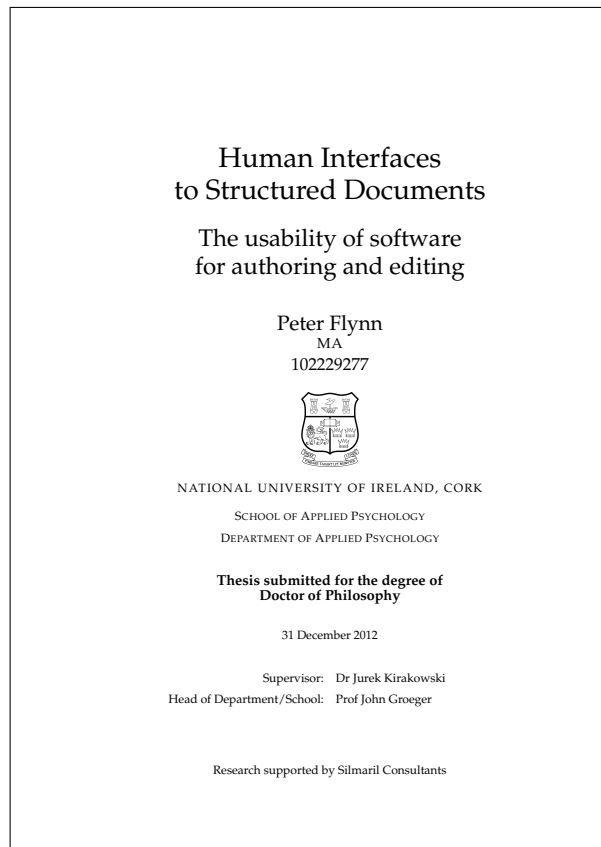


Figure 2: Title page layout for the UCC thesis

one page in a document that will run to hundreds of pages, but it is the first page people see, and the first place that critical eyes will look for errors. It is also hard to convince new postgraduate students that they do not have much choice in how this page is laid out.

The current layout (see Figure 2) is vertically- and horizontally-centred, and contains conventional metadata:

- title and subtitle;
- author and [optional] registration number and qualifications;
- institutional identity (crest³ and name);
- divisional identity (college, school, etc);
- class of degree;
- date;
- names of supervisors and head of discipline.

The only less conventional addition was the acknowledgement of a sponsor — this is commonplace in the Acknowledgements, but its appearance on the title

³ The crest is a character in a METAFONT font which was drawn for us by Jeremy Gibbons in 1989, and is only accessible for download locally.

```

\documentclass[apsych,phd,12pt]{uccthesis}
\usepackage{palatino}
\begin{document}
\title{Human Interfaces
  \\\to Structured Documents}
\subtitle{The usability of software
  \\\for authoring and editing}
\author[102229277]{Peter Flynn}
\qualifications{MA}
\department{Applied Psychology}
\professor{Prof John Groeger}
\supervisor{Dr Jurek Kirakowski}
\sponsor{Silmaril Consultants}
\date{31 December 2012}
\maketitle
...
\end{document}

```

Figure 3: Commands for the title page metadata

page is becoming a condition of some funding agencies. There is provision for the numbering of multi-volume theses with a `\volume` command, but I am informed by the Graduate Studies Office that multi-volume theses are a great rarity nowadays.

Because of complexities in the university structure, we had decided to enforce the discipline and degree options in the `\documentclass` command, rather than allow authors to name their affiliation in an uncoordinated manner. This enabled us to use the information before anything else, to set the string names for the faculty (college) and department (school) and the bibliographic reference format prevalent in the discipline. The remaining metadata, therefore, is given with conventional commands as shown in Figure 3.

The `\title`, `\author`, and `\date` are standard; the remainder were added with the commonly-used method of defining an internal command default which is then redefined when the author uses the equivalent external command, for example:

```

\def\@subtitle{\relax}
\newcommand{\subtitle}[1]{%
  \gdef\@subtitle{#1}}

```

We can then test the internal commands for equality to `\relax` during the processing of `\maketitle` to see whether or not the metadata commands were used, so that their absence can be accommodated in the spacing or replaced by warning messages, for example:

```

\if\relax\@subtitle
  \else{\large\@subtitle\par}\fi

```

```

\if\relax\@professor You need to give the
  name of your head of discipline
  \else\@professor\fi

```

All the additional commands are given defaults or warnings in this way, so that a beginner accustomed only to `\title`, `\author`, and `\date` (or a user who has not Read The Fine Manual) will not be faced with L^AT_EX errors.

2.2 Sequencing of preliminary pages

The university rules require only that the Table of Contents comes immediately after the title page. Because the underlying report is invoked with the `oneside` option, there are no blank pages. The `\tableofcontents` command is therefore contained in the `\maketitle` command, so that it cannot be omitted or moved.

The List of Figures and List of Tables are also in the `\maketitle` command, but as the class must also be usable in disciplines where there may be no tables or figures at all, the need for these lists is determined by two Boolean switches:

```

\iflof\listoffigures\fi
\iflot\listoftables\fi

```

These are set to false at the end of the document if there were no figures or tables, using global counters defined in the table and figure code, and written to the `.aux` file, where they will take effect on the subsequent run:

```

\ifnum\c@totfigure=0 \write\@mainaux{%
  \string\global\string\loffalse}\fi
\ifnum\c@tottable=0 \write\@mainaux{%
  \string\global\string\lotfalse}\fi

```

However, there are also class options `noLot` and `noLoF` which will prevent the LoT or LoF being used even when tables and figures are present — when there are only one or two tables or figures in use, a formal list may not be wanted.

The compulsory formal Declaration that this work is the student's own is also produced automatically as part of the `\maketitle` command, after the ToC (and LoF and LoT, if present), set centred on a page to itself.

No other declaration is required by the university, and it is not usual for copies of any of the forms signed by the supervisor or Extern to be included, but these could clearly be implemented at the same point by the same method if needed.

A considerable number of students require preliminary (unnumbered) sections, before the thesis proper starts with the first part or chapter. These are needed to hold explanatory material such as an Introduction, a list of materials, or tutorial matter

on a special topic. While this could be done with a `\section*` command, a new `\prelim` command was created to ensure a page-break beforehand, and to create an entry in the ToC which would otherwise be absent.

The `abstract` environment was also changed to use this `\prelim` so that it too would occupy a page to itself (the Abstract is limited by the rules to 300 words).

Two new environments were created, `dedication` and `acknowledgements`; the first sets the content centred on a page by itself; the second just uses the `\prelim` command to title the Acknowledgements. The rules do not specify an order for the Abstract, Dedication, Acknowledgements, or any other prelims.

No decision has been taken about the position of glossaries: these are not mentioned in the rules, and while the `glossaries` package is recommended, there is no compelling evidence one way or another for its placement either here or at the end of the document.

2.3 Formatting and layout

The current rules [5] are very undemanding in this regard:

The text must be either, printed, typewritten or otherwise reproduced on good quality size A4 paper, with a left-hand margin 4 cm. Double or one and a half spacing is recommended. Copies must be bound or otherwise securely fastened and numbered consecutively, page numbers to be located centrally at the bottom of the page.

No mention is made of the other margins, or of the typeface or size, or of the format of bibliographic references, so some unilateral decisions were made (the handling of bibliographic formats is dealt with in § 3).

- The top and bottom margins are set to 3cm and the right-hand (foreedge) to 2.5cm;
- The typeface defaults to Computer Modern;
- The body size defaults to 11pt;
- The `setspace` package is used to set the default to 1½ line-spacing — some leading would in any case have been needed for 11pt type on lines this length;
- Page numbering uses roman numerals from the title page to the beginning of the first part or chapter, at which point it restarts in arabic numbers.

After some discussion locally and on `comp.text.tex`, we decided to make the default document setting `\raggedright`. This was partly because it helps

avoid H&J problems, especially in the natural sciences where very long words are more frequent; partly because a thesis is not a professionally-typeset publication like a book, and does not appear to benefit greatly from justification; and partly because ragged-right setting improves readability on a page width with relatively long lines.

However, `\raggedright` also turns off paragraph indentation, so the `parskip` package is used to add space between paragraphs. This layout is in fact expected by students whose experience to date has been restricted to wordprocessors, where it is conventional to use an empty paragraph between paragraphs to simulate paragraph-spacing. Two extra options, `justified` and `indented` can be used to restore the state to book-style.

Running headers and footers were implemented with the `fancyhdr` package, to provide navigational detail from `\leftmark` and `\rightmark` *without* the capitalization used in the default classes (this proved remarkably difficult to defeat), and *with* the use of a `\hbox` in the definitions to allow long titles to be line-wrapped. In draft mode, the footer also provides the `\jobname` and a timestamp.

Some minor changes included more space above and below captions; the enforcement of the rule for caption positions (Tables: above, Figures: below) by restyling floats with the `float` package;⁴ the allocation of more space to the page number in the ToC, LoF, and LoT; and the reassignment of wider values to the page-fractions for floats, along the lines suggested in the T_EX FAQ [1].

Two minor changes are made to the default layout of block quotations and description lists (below), but all the remaining parts, chapters, sectioning, lists, and other structural elements behave and appear as usual in a L^AT_EX document. The `\usepackage` command can be used in the normal way: a list of the packages already in use is contained in the class documentation.

2.4 Markup abbreviation

No attempt was made to abbreviate any of the commands: although students do this frequently for their own pattern of usage, I have not been able to see any common methods. There are a few conventions in some disciplines for short-name macros for commonly-occurring constructs, and authors are free to use them. However, as mentioned earlier, some frequently-occurring constructs were found to benefit from a small amount of automation, the most important of which is the block quotation.

⁴ This has the (possibly) useful side-effect of allowing the [H] positional specifier.

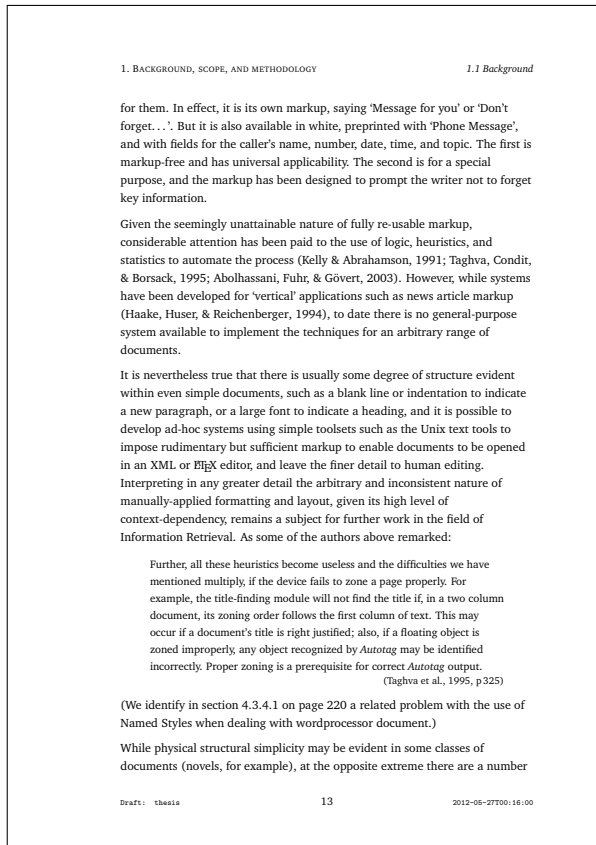


Figure 4: Default page layout for the UCC thesis

2.5 Quotations

The L^AT_EX default for quotations is unusual, in that it does not make the type size smaller, nor does it defeat indentation on the first line, both of which are established conventions.⁵ In academic work, a block quotation also usually requires a citation.

To implement this, the `quotation` environment was modified to change the size and start with a `\noindent`; but also to take an argument, the BIB_TE_X key of the cited passage. This enables the quotation to be set with a right-aligned citation immediately below. Although such a citation would be compulsory, it is currently defined as optional in order for its presence to be tested, and to allow for uncited use where the context already makes the origin obvious. However, if an optional argument such as this is to contain page or chapter references, which are themselves optional to the `\cite` command, additional armour is required:

```
\begin{quotation}[{[p.36]{smith92}}]
```

⁵ I have never been able to find a use for the `quote` environment, so the use of `quotation` is recommended to authors.

In response to user requests for an Epigraph at the start of chapters, an `epigraph` environment was created, in the same way as the modified `quotation` above. This has two arguments, however: a compulsory one for the BIB_TE_X key, but if the citation is to be informal and non-rigorous, that argument can be empty, and the optional argument can be used, for example:

```
\begin{epigraph}[Popular saying]{}]
```

In both cases this is cumbersome and needs regularising, so a future version will probably use the `xargs` package to handle the additional metadata.

2.6 Description lists

A frequent annoyance for users is the inability of the label argument to an `\item` in the `description` list environment to be broken at line-end when it is very long. The default formatting is also unusual compared with modern practice of setting the label value on a line by itself (as with the default formatting of HTML's `<dt>` element type). A number of alternative formats from the L^AT_EX Companion [4] have been tried, but no final decision has been taken on this yet.

3 Adherence to university structure

The university is currently in transition from the traditional hierarchical Faculties and Departments to a more fluid structure of Colleges and Schools. As a result there a complex and overlapping transitional organisation in which disciplines are being merged, split, and renamed. In order to ensure that the correct names are used, the author's affiliation and class of degree must be given as class options, rather than as free text in the argument of a command. This also avoids misspellings, and the unfortunate tendency of some students to represent the discipline in terms other than the official ones. With the imminent arrival of electronic submission, where the name of the discipline or school will be part of the PDF metadata, regularity is becoming more important.

In creating these options, it became clear that other data could also be keyed to it, in particular the bibliographic reference format required for each discipline. As a result, selecting an affiliation option now both sets the correct naming *and* presets the `.bst` file (and any associated `.sty` file) for the discipline.

As there are currently 87 options for affiliation and 92 for class of degree, this method would have been unworkable in terms of manual maintenance. Fortunately, the class was developed using an XML-based methodology which generates the `.dtx` and

A university thesis class: automation and its pitfalls

.ins files, so the relevant string names and tokens could simply be transformed from an annual XML extract from the databases maintained by the university administration. In any case, if and when the transitional phase of restructuring is completed, the class interface will be updated to use a key/value syntax rather than 179 separate options!

4 Testing, feedback, and adoption

Informal testing was initially carried out by use in my own thesis, but was extended to drafts provided by students who came looking for help with L^AT_EX formatting. Discussions were held with the Registrar's Office and the Graduate Studies Office to ensure that the layout implemented conformed to the rules.

In January 2010, an early version of the XML-generated class was made available locally for download [3], and over the course of 18 months about 45 students used it for their theses, reporting bugs as they were discovered.

The feedback was largely positive, and the automation of the title page and prelims was seen as a major benefit. The most useful feedback came as bug reports, and led to a spate of updates over the next six months as various solutions to errors were tested and implemented.

Some unresolved issues remain:

- separate options for the disciplines and classes of degree need to be replaced by key/value pairs;
- glossaries appear to be much more common than was previously envisaged;
- the formatting problems of description lists remain unresolved;
- the debate continues over the default unindented, paragraph-spaced, ragged-right setting *vs* justified and indented setting;
- the identification of the 'right' bibliographic reference format for each discipline is problematic. An enquiry among colleagues representing each discipline provided only a 20% response, so there is a lot more data to collect (the default has been set to Harvard). A few disciplines have two common formats — Physics, for example, allows either IEEE^{TR} or AIP — so a mechanism is required to allow that to be specified.

At the moment the assumption is that students will be using BIB_TE_X, not *biblate_x*, as not all the formats required are yet available in the latter.

The class has by no means been tested to destruction: it appears to work with all the common packages, including *hyperref*, but the more use it gets

and the more bugs are reported, the more likely it is to work.

A number of academics who use L^AT_EX themselves have started recommending the class to their students, and some informal changes have been made to create variant formats within the class for essays, term papers, and minor dissertations. An approach has also been received from a group of students in another institution in the state (which has no resident L^AT_EX expertise or support), asking for help in writing a thesis class, and I am aware of at least three students in a third institution who have adopted the class and simply changed the identity by editing the .cls file.

Writing a class file is a non-trivial activity, and I was fortunate to have access to a development and maintenance methodology which made editing and creation very much easier than writing a .cls file by hand. Having a clearly-defined goal makes development easier, as does having a set of patterns to work to, and I am indebted to the many authors of the classes listed in Figure 1 for their work. The intention is that from version 1.00 the package will be available on CTAN (minus the crest, which is reserved to the institution).

References

- [1] T_EX Frequently Asked Questions. <http://www.tex.ac.uk/faq/?label=floats>, Mar 2012.
- [2] Peter Flynn. Typographers' Inn. *TUGboat*, 33(1):8–10, 2012.
- [3] Peter Flynn. UCC Thesis Document Class. <http://research.ucc.ie/latex/#uccthesis>, Apr 2012.
- [4] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The L^AT_EX Companion*. Addison-Wesley/Pearson Education, Boston, MA, 2nd edition, 2004.
- [5] University College Cork. Thesis Submission: Layout of Thesis. <http://www.ucc.ie/en/graduatestudies/current/thesis/Layout/>, Nov 2009.

◇ Peter Flynn
Electronic Publishing Unit,
University College Cork
Cork, Ireland
Phone: +353 21 490 2609
pflynn (at) ucc dot ie
<http://research.ucc.ie/profiles/H505/pflynn>

The Wonders of `\csname... \endcsname`

Amy Hendrickson

57 Longwood Avenue

Brookline, MA 02446

USA

amyh@texnology.com

<http://www.texnology.com>

Abstract

A surprisingly useful tool, `\csname, \endcsname`, offers many opportunities for interesting and useful macros, especially when it is convenient to dynamically generate a series of definitions.

Trivially a series of `csname` definitions may be used to produce endnotes, but there are more interesting and complex constructions as well.

A second example shows how `\csname` may be used for on-line report generation. In this instance, we dynamically generate hyperlinked tabs for a custom risk analyses of particular stocks chosen on-line by the client. We can use these named tabs to build a hyperlinked TOC on the fly.

The final example shows how definitions made with `\csname` can be used to send a set of definitions to an auxiliary file, where each new definition contains the current page number in its name, and a number as its definition.

This allows the dynamic redefinition of the command for a particular page, *within the auxiliary file* depending on whether the value of the new definition is higher than the value of the previous definition for the same page.

When the auxiliary file is brought into the base `.tex` file the next time `LATEX` is run on the document, it will include a series of unique macros, one for each page in the document, defining the highest number given for that page. Since the definition is made with `\csname... \endcsname` we can have the page number contained in the name of the definition. This allows us to call the definition in the running head of the `LATEX` document, using the current page number in the `\csname... \endcsname`. We'll see a practical use for this construct.

Code will be shown for each of these methods to dynamically generate macros using `\csname`.

1 The Basics

The TeX primitive commands `\csname, \endcsname` allow some useful macro constructs that wouldn't otherwise be possible. Here are some of its useful characteristics.

1. We can use `\csname` to find out if a command has been defined, since an undefined command is equal to `\relax`. `\csname` allows us to test to see if this is the case, and make choices based on the result: `\expandafter\ifx \csname anycommand \endcsname\relax <do this>\else<do that>\fi`.
2. Unlike commands made with `\def`, commands may be defined with `\csname... \endcsname` that include non-letters in the name of the definition (with the exception of `%`).

For example, this is a valid definition that uses symbols and a number in its name:

```
\expandafter\def\csname $#2\endcsname{Hi!}.
```

It may be called using `\csname, \endcsname, \csname $#2\endcsname` to produce: Hi!.

3. We can drop macro arguments in the form of a parameter number into `\csname... \endcsname`. In this example from `LATEX` code, `\setcounter` tests the first argument to see if it is a counter that has been defined with `\newcounter`; if false, give error message; if true, globally set the counter to be equal to the second argument.
- ```
\def\setcounter#1#2{\@ifundefined{c@#1}%
 {\@nocounterr{#1}}%
 {\global\csname c@#1\endcsname#2\relax}}
```

This has the effect of hiding some complexity from the user, who only sees the name of the

counter and the number to which the counter is set, i.e., `\setcounter{page}{21}`.

- Here's where things get interesting. We can expand commands within a definition name made with `\csname... \endcsname`. This opens up many complex possibilities. For a set of possibilities, we can include a counter in the name of a new definition: `\expandafter\def \csname apple\the\applenumber\endcsname{}`

In this article we'll see a number of ways we can use `\csname... \endcsname` with counters.

### 1.1 Dynamic Macro Building

We can use a counter within `\csname... \endcsname` to make a series of macros, a new one every time the counter is advanced. We do this by including a definition, made with `\csname... \endcsname` with a counter in its name, within the body of another definition. The outer definition advances a counter every time it is used, producing a new and unique macro every time it is called.

Using our previous example: `\expandafter \def\csname apple\the\applenumber\endcsname` we can make a command that will make more commands in this way:

```
\newcount\applenumber
\def\applenumber#1{\global\advance\applenumber by 1
\expandafter\def\csname apple\the\applenumber
\endcsname{#1}}
```

Now we can access the newly made inner `csname` macro by using a loop, which advances a counter in each iteration, and calls the `csname` macro using a counter in the body of its name.

Here we call the newly made `csname` macros with a loop that advances a counter. This example tests to see if the command is defined, and if true, gives it a number with `\the\loopnumber` and calls the command; else, ends the loop.

```
\newcount\loopnumber
\loopnumber=1
\loop\expandafter\ifx
\csname apple\the\loopnumber\endcsname\relax
\else \the\loopnumber.
\csname apple\the\loopnumber\endcsname
\global\advance\loopnumber by 1
\repeat
```

Used:

```
\applenumber{Macintosh}\applenumber{Gala}
```

Results:

- Macintosh
- Gala

This tool has surprisingly many uses. For our first real world example: making endnotes.

## 2 Endnotes Example

In this example we want to change the definition of footnote so that it produces endnotes rather than footnotes. We do this by making an endnote definition that makes a new macro every time it is used.

We start by making a new counter to be used by our endnotes, `\endnumber`. In the `\endnote` macro we advance the `\endnumber` counter, then raise and print the number in the text for our endnote number.

Next we make a construction with `\csname` that builds a new definition, using the current state of the `\endnumber` counter. This new definition will save the text of the endnote.

```
\newcount\endnumber

\def\endnote#1{\global\advance\endnumber by 1
\the\endnumber}%
%%
%% Here we make the new definition using
%% \the\endnumber in the definition name so that
%% each new definition is unique:
%%
\long\expandafter
\def\csname endnote\the\endnumber\endcsname{%
\small\leftskip=12pt\relax\parindent=-12pt
\indent\hbox to 12pt{\the\loopnumber.\hfill}}
%%
%% Here we save the text of the endnote:
#1
\strut\vskip2pt}}
```

Now we set footnote to be equal to endnote, so every time `\footnote` is used, the command actually called is `\endnote: \let\footnote\endnote`

To print the endnotes, we make a loop that advances a counter with every iteration. That counter is used within the name of the definition made with `\csname... \endcsname`. The loop continues until it comes to an undefined endnote, thus cycling through every defined endnote.

```
\newcount\loopnumber
\def\printendnotes{\global\loopnumber=1
%%
%% Test to see if any end notes have been
%% defined; If so, provide the title and
%% start loop; if not, do nothing.
%%
\expandafter\ifx
\csname endnote\the\loopnumber\endcsname\relax
\else
\subsection*{Endnotes}\everypar{}
\vskip6pt
\small\leftskip=12pt
```

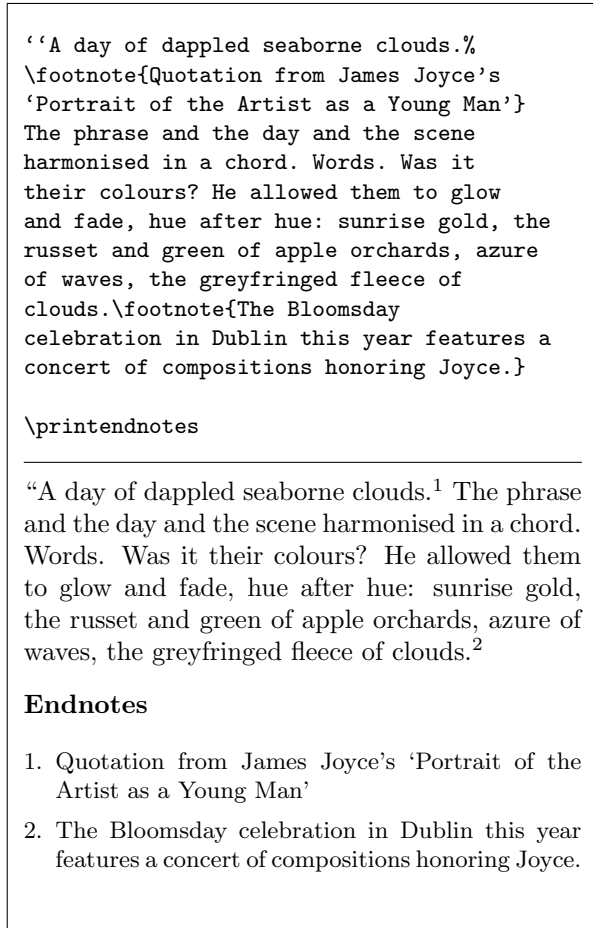


Figure 1: Testing the Endnote Commands

```

%% Loop continues until it finds an
%% undefined endnote
%%
\loop\expandafter\ifx
\csname endnote\the\loopnum\endcsname\relax
 \else
%% Print endnote
\csname endnote\the\loopnum\endcsname
\vskip2pt
%%
%% Reset: redefine current endnote to \relax
%% preventing this definition from being
%% used the next time \printendnotes is called.
%%
\global\expandafter
\let\csname endnote\the\loopnum\endcsname\relax
%%
\global\advance\loopnum by 1
\repeat
\fi
%% \fi ends test at beginning of this macro
%% to see if any endnotes have been defined.
}

```

### 3 Example: On-line Report Generation

A somewhat similar construction may be used to make hyperlinked tabs for on-line report generation.

This set of macros is used to automate the naming of hypertargets so that we can hyperlink to them on the first page of the report, using a `csname` construction and a loop, and using Tikz for making the hyperlinked tab.

The name and number of companies analyzed is determined by the client who submits a request online. Each company's analysis will start on a titled new page. Part of the definition for the title of the report includes this command: `\maketab{#1}`

`\maketab` takes a stock symbol as its argument, and generates a hypertarget so that we can link to it from the beginning of the report, in the equivalent of the table of contents page, using the same `\codenum` counter. Then it makes a new definition with `\csname` and the `\codename` counter in its name, with the stock symbol as its definition, and sends it to the `.aux` file.

```

\def\maketab#1{\global\advance\codenum by 1
\hypertarget{link\the\codenum}{ }
\immediate\write\@auxout{\string\expandafter%
\string\gdef\string\csname\space
tab\the\codenum\string\endcsname{#1}}

```

Once we have this in place we can use our loop construction for the first, and possibly continuing, pages to build the hyperlinked tabs. `\gettabs` uses a loop to call the individual tabs, as long as there is one defined. This can continue over a number of pages if necessary.

```

\begin{multicols}{5}
\loopnum=1\gettabs
\end{multicols}

```

As you can see, `\gettabs` is where the work is done. Here is how it is defined.

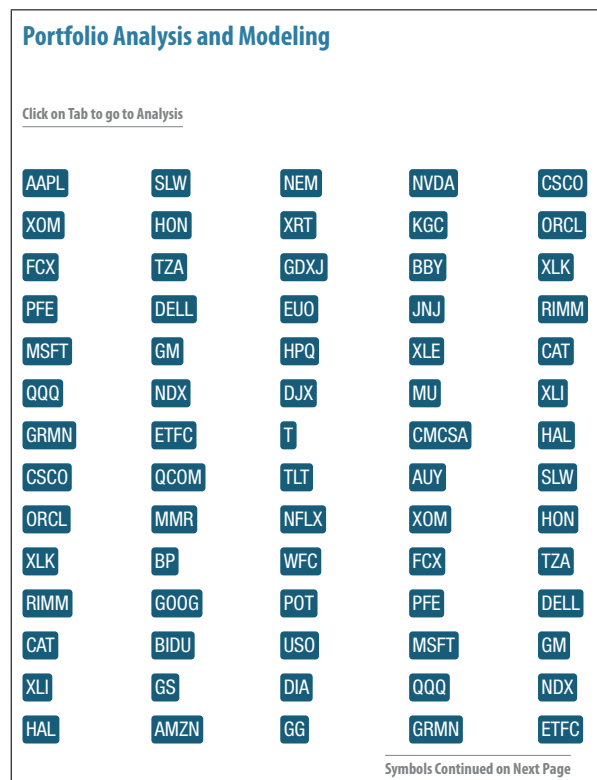
```

\def\gettabs{\loop
\expandafter\ifx
\csname tab\the\loopnum\endcsname\relax
 \else
\vskip6pt\hbox to 1in{%
%%
%% \hyperlink takes two arguments;
%% the first the name of the hypertarget,
%% and the second, the text that will link
%% to the hypertarget when clicked:
\hyperlink{link\the\loopnum}
{\plaintab{\csname tab\the\loopnum\endcsname}
\hskip12pt}
\hfill}}% <== end \hbox started above
\global\advance\loopnum by 1
\repeat}

```

If you are interested in how to make the tab with Tikz, here is the code:

```
\definecolor{dkblue}{cmyk}{.9, .53, .32, .2}
\def\plaintab#1{%
\hbox{\normalsize\sf
\begin{tikzpicture}
[rounded corners=3pt, inner sep=3pt]%
\node[rectangle,fill=dkblue]
{\Large\sf\color{white}
\vrule depth 3pt width 0pt height 15pt \relax
#1};
\end{tikzpicture}}}
```



**Figure 2:** One form of automated online report generation, this is a draft version of customized financial report. Each symbol automatically generated and is hyperlinked to the appropriate page of the report. The company analyzed depends on input from the client; the symbols and their linking is done through macros utilizing `cname`.

#### 4 Example: Redefinition of macro within the auxiliary file

Our final example shows a technique that might be used to solve various problems. In this case, we are using `\csname` to determine the highest level of security classification on a particular page, so that we can print the highest level on the top and bottom of the page (Unclassified, Classified, Secret and

Top Secret). Every paragraph on the page will be marked with one of these classification levels, in any order. Figures and tables and their captions will be marked as well.

This problem is difficult because we don't know initially the page number where each paragraph, table or figure, will appear. In addition we don't have a way of determining which is the highest level for any particular page.

There are many more complications to this general problem. For instance, how do we pass information on the level of a paragraph that has broken over pages, so that the part of the paragraph on the second page will contribute to the calculation of the highest level on the second page? For the sake of brevity, let's consider only the general mechanism here.

#### The solution

The solution comes from our convenient tool, `\csname`, and its ability to use a counter in its name to make a series of unique definitions.

To this we add the innovation of making further definitions in the auxiliary file using a conditional to determine whether the current classification number is the highest for the current page number. Only if it is the highest number will the `pagenumber` and the classification level be defined so that the information can later be accessed to be used in the running head and foot for that particular page.

#### 4.1 Setting up

We use a `\write` for every instance where a classification level is written in the text with the command `\secmark`. `\write` is only activated after the page is made up, so we are sure that we will be using the correct page number when we send the information to the auxiliary file.

Since we will have many `\write` commands in the `.tex` document, we will write to a new auxiliary file, `\jobname.lev` instead of using standard L<sup>A</sup>T<sub>E</sub>X auxiliary file, `\jobname.aux`. We name the new write: `\newwrite\collect`

However, we won't open the new auxiliary file in the body of the `.cls` file, since we need to input the current version of `\jobname.lev` and access the commands found in it before opening a new file.

To do this we can redefine `\document` so that when `\begin{document}` is used it will input the current `\jobname.lev` and only then open the new `\jobname.lev` to be populated with `\csname` commands produced in the next L<sup>A</sup>T<sub>E</sub>X run.

We can do this by saving the current version of `\document`, and then redefine it to use its original definition, plus inputting our auxiliary file, and opening the new file:

```
\let\savedocument\document

\def\document{\savedocument
%% We temporarily open \jobname.lev:
 \openin\@inputcheck\jobname.lev %
%% if end of file=@inputcheck, no file exists
 \ifeof\@inputcheck
%% if no file let users know they must
%% run LaTeX again:
 \typeout{^^J^^J
 !! Please run LaTeX again to get
 correct classification levels^^J^^J}
\else
%% Since we know now that there is a file
%% called \jobname.lev we input it and
we get the information from the last
%% LaTeX run:
\input \jobname.lev
\fi
%% And Now we can open the file where info
%% from the current LaTeX run can be saved:
\immediate\openout\collect=\jobname.lev
...
```

#### 4.2 The counter to be used

The next item we need is a counter to use when defining our `\csname` commands. Since we want a command that has the current page number in its name, we would be tempted to use the L<sup>A</sup>T<sub>E</sub>X page counter, `\c@page`.

However, in the case where the beginning of the document is in roman, and the body of the document is in arabic, we have the unfortunate result of having multiple pages with the same page number.

So instead, we make a new counter, and call it `\superpage`: `\newcount\superpage`.

We can use `\shipout` to advance this counter. (`\shipout` is the T<sub>E</sub>X primitive that is called every time a page is completed.) This gives us a number that is continuous through the document.

`\shipout` will be used to print the classification term on the top and bottom of the page. We use `\shipout` to generalize this solution, so that this system will work independently of any page style, and its headers and footers.

Now we can add this redefinition of `shipout` to our redefinition of `\document`. If the `hyperref` package has been used, we need to access `\shipout` through the `\AtBeginShipout` command, so we test to see if that command is available. If it is not defined, we redefine `\shipout` directly. However, if `\AtBeginShipout` is defined, we know that `hyper-`

`ref` has been used, and we use `\AtBeginShipout` to advance the counter and call the classification commands to print the classification level at the top and bottom of the page.

```
\newwrite\collect
\newcount\superpage

\let\savedocument\document

\def\document{\savedocument
 \openin\@inputcheck\jobname.lev %
 \ifeof\@inputcheck
\typeout{^^J^^J
 !! Please run LaTeX again to get
 correct classification levels^^J^^J}
\else
\input \jobname.lev
\fi
\immediate\openout\collect=\jobname.lev
%%
%% Now test to see if \usepackage{hyperref}
%% has been used:
 \expandafter\ifx
\csname AtBeginShipout\endcsname\relax
%%
%% If hyperref has not been used do this:
%%
\let\saveshipout\shipout
\long\def\shipout\ vbox##1{\saveshipout
\ vbox{\global\advance\superpage by 1
\ vbox to0pt{\vss\topofpage\vskip36pt}
##1\ vbox to0pt{\vskip6pt\bottomofpage\vss}
}}
 \else
%% hyperref is used, so we use \AtBeginShipout
%%
\AtBeginShipout{
\setbox\AtBeginShipoutBox=
\ vbox{% Counter advanced
\global\advance\superpage by 1
%% At top of page:
\ vbox to0pt{\vss
\centerline{\classfont\makeclassification}
\vskip12pt}
%% Contents of typeset page:
\box\AtBeginShipoutBox
%% At bottom of page:
\ vbox to0pt{%
\centerline{\classfont\makeclassification}
\vss}}
%% Hyperref error prevention, avoids
%% confusion when secmark is used in a section
%% head and will be found in a bookmark:
\pdfstringdefDisableCommands{%
\let\uppercase\relax
\let\secmark\secmarkletter}
\fi} %% end test to see if hyperref is used.
```

### 4.3 Doing the Writes, the Neat Part!

The `\secmark` macro works by sending a definition for the classification level on a particular page to `\jobname.lev` file, using a `\write` associating the page number with the level given. The `\write` will not be activated until the page is made up, so we are guaranteed to have the correct page number sent to the `.lev` file. This works as well for figure or table floats, since `\write` will send out the information to the `.lev` file only after the page is made up, and the page where the floats will appear is determined.

The `\write` sends information to the auxiliary file, `\jobname.lev`, including several conditional tests. The command looks messy and verbose, when the write is made, since we have to stop the expansion of many commands by preceding each one with `\string`, except for those commands that we want to expand immediately, in this case, the super page number:

```
\write\collect{%~J makes a blank line
%% in the \jobname.lev file so that
%% it is easier to see where each test ends.
~J~J
%%
%% First test to see if this definition has
%% already been made; if not, do a gdef (global
%% def is necessary for a definition in an
%% auxiliary file that will be input to another
%% file) using csname and the superpage number
%% to define the highest level on that page;
%% If it has been defined, test to see if
%% previous number is lower than previous
%% definition; if so, redefine.
%%
\string\expandafter\string\ifx\string\csname%
\space Level0nSuperPage\the\superpage
\string\endcsname\string\relax
\string\expandafter\string\gdef\string\csname
\space Level0nSuperPage\the\superpage
\string\endcsname{#1}
\string\else
\string\ifnum \string\csname\space
Level0nSuperPage\the\superpage\string\endcsname
\string<
#1
\string\expandafter\string\gdef\string\csname
\space Level0nSuperPage\the\superpage
\string\endcsname{#1}
\string\fi\string\fi
~J}%
```

... which makes more sense visually when we see how the code looks by the time it is expanded and appears in the `\jobname.lev` file. Here, the level sent for page 5 is ‘2’.

```
\expandafter\ifx
\csname Level0nSuperPage5\endcsname\relax
\expandafter\gdef
\csname Level0nSuperPage5\endcsname{2}
\else\ifnum\csname Level0nSuperPage5\endcsname<
2 \expandafter\gdef%
\csname Level0nSuperPage5\endcsname{2}
\fi\fi
```

This process can be repeated as many times as needed for each page, with only the highest number, determined by each test, being used to define `\csname Level0nSuperPage?\endcsname`.

### 4.4 Using the Level Information

As we saw earlier, the `\jobname.lev` file will be input with `\begin{document}`. This will bring in a series of definitions, one for each page, that associates a page number with a classification level. We can use this information with every shipout, when the macro `\makeclassification` will be called at the top and bottom of the page. Here is its definition:

```
\def\makeclassification{%
\ vbox{%
\ baselineskip=12pt
% Is there a definition for this page?
\ expandafter\ifx
\ csname Level0nSuperPage\the\superpage
\ endcsname\relax
% if not:
\ centerline{}
\ else
% if there is a definition:
\ centerline{%
\ ChangeNumIntoClassification{%
\ expandafter\csname
Level0nSuperPage\the\superpage
\ endcsname}}\vskip3pt\fi}}
```

`\ChangeNumIntoClassification`, seen above, uses the definition of `\csname Level0nSuperPage\the\superpage\endcsname` as its argument, which will yield a number from 1 to 4. This allows us to use `\ifcase` to trivially change that number into the classification term:

```
\def\ChangeNumIntoClassification#1{%
\ ifcase#1\or Unclassified \or Classified
\or Secret \or Top Secret
\ else%
! Please Run LaTeX Again to Get the
Classification Level !
\ fi}
```



## 5 Putting These Techniques to Use

Likely there are many more opportunities to use these techniques, particularly with off label uses for L<sup>A</sup>T<sub>E</sub>X such as report generation, or building e-documents on the fly, and other web oriented macro writing projects. To summarize:

We can use `\csname<counter>\endcsname` to generate a new and unique command every time an outer command is used and the counter advanced. For example, an `\endnote` command may be defined that generates a new definition every time it is used.

A `\csname...\endcsname` definition with a counter in its name can be used to generate a series of hypertext targets automatically.

In both of these cases, and in general, we can use a loop, with an internal counter advanced each time it is used, to access the new definitions.

We can stop the loop by testing to see if the most recent `\csname<counter>\endcsname` combination has been defined. Using this method to stop the looping has the advantage that we don't need to know in advance how many definitions were made; we will cycle through all available definitions before ending the loop.

Finally, we have the technique of sending information to an auxiliary file with a `\write` and making new `\csname<counter>\endcsname` definitions in the body of the auxiliary file, based on the results of a conditional test. When the auxiliary file is input into the root `.tex` file, we can then use the resulting definition in a variety of ways.

These techniques add to our understanding of the exceptional flexibility of Knuth's T<sub>E</sub>X language.

\* \* \*

May your explorations in this territory prove enjoyable and fruitful!

Preprint of

# My Boston: Some printing and publishing history

David Walden

Based on a presentation for the July 2012 T<sub>E</sub>X Users Group annual conference,  
Omni Parker House, Boston, Massachusetts.

**For practical reasons, the dozens of photographs, scans of historic book pages, etc., which are the focus of my presentation, have been left out of this preprint.** The slide numbers are in the margin notes.

slide  
1

During the four summers before each of my college years, I worked in a large cardboard box printing plant (big letter presses and lithography presses) in a small industrial town 40 miles east of San Francisco. Thus began my fascination with printing. I was also an avid reader of books and of *The New Yorker* magazine to which my father subscribed. I dreamed of eventually living in a big city with big libraries and thick newspapers. Thus, after college, I moved in 1964 to the Boston area (where I have remained ever since).

As I explored the Boston and Cambridge in the 1960s, I became aware of a number of publishing and printing activities, often by walking or driving by their then current locations and buildings. I also began to use the libraries and to frequent the bookstores. Compared with the small town in the Central Valley of California in which I had grown up (and even compared with San Francisco where I went to college), Boston was a mecca for someone interested in books, magazines, and the related printing, publishing and distribution world.

With TUG2012 (in some sense a publishing event) being held in Boston, I got to thinking about and then looking into the history of printing, publishing, libraries, bookstores, and so forth in Boston. In this note, a written variation on my TUG2012 presentation, I present a sketch of what I have learned.

2

## Contents

1. Colonial period, 1630–1775
2. Revolutionary War (1775–1783) and transition
3. Liberal elites of the mid-19th century
4. Boston is just another big American city, mid-1800s on
5. Personal observations, 1964–present

Acknowledgments

References and bibliography

3

A lot of this history happened close to the conference hotel because Boston was once essentially a small island (the neck of land to the mainland was sometimes under water at high tide).<sup>1</sup> The location of our conference hotel is close to the center of this small almost island. Thus, anything that happened in early Boston, and much of what took place “downtown” later, took place near the hotel location.

4

There are several maps which a reader might look at while reading this note (the conference hotel is at the southwest corner of Tremont and School Streets on all three maps):

- 1772 Bonner map of Boston  
<http://www.doak.ws/1722MapOfBostonJohnBonner.jpg>
- Boston Freedom Trail map  
<http://twoweekstotravel.com/wp-content/uploads/2010/12/Freedom-Trail.jpg>
- “Literary Boston, 1794-1862” map  
<http://bostonliteraryhistory.com/sites/default/files/bostonliteraryhistorymap.pdf>

---

<sup>1</sup> Krieger99 and Whitehill68.

Space and time do not allow a thorough presentation of the Boston-region history of printing, publishing, and the like. In particular, I have mostly not talked about the author part of the literary world.<sup>2</sup> Also, this is not a scholarly piece of research (my narrative is based on what I have read in secondary sources, been told by someone, or found in the Wikipedia). It also glosses over many details; as one example of many imprecise statements, I call the early college in Cambridge Harvard and ignore its early name. I hope my fragmentary narrative is suggestive of the actual history of events.

## 1 Colonial period, 1630–1775

The Pilgrims, who previously had left England to go to Holland in the Netherlands, came to Plymouth, just south of Massachusetts Bay, in 1620. Another Massachusetts-based outpost was attempted a Cape Ann in 1624. In 1628–1630 a succession of largely Puritan settlers arrived in the Massachusetts Bay Colony settling in locations from Salem to Boston.

The Puritans came to the Massachusetts Bay Colony fleeing what they felt was an incorrect approach the theology of the Church of England and the relationship of church and state (King James’s approach subordinated the church to the state). In particular, in 1630 Governor John Winthrop and other Puritan leaders arrived carrying a charter allowing the Massachusetts Bay Colony to be governed from the colonies rather than from England. Both Boston and Cambridge (a few miles up the Charles River) were settled by the Puritans around 1630. These largely Puritan immigrations to the Massachusetts Bay Colony continued for the next 10 years.

The arriving Puritans were idealistic about their new home, and John Winthrop gave a sermon quoting the Sermon on the Mount and saying that they in the Bay Colony would be a “citty on a hill,” watched by people throughout the world for the purity of their religious practice (and the way it was supported, i.e., enforced, by the government). Thus, they believed in education such that their citizens could study the Bible and read the laws and acts that governed them. By 1635 they had established the first public school in English North America, Boston Latin (there are signs on both sides of School Street outside the side door of the conference hotel noting early locations of the Boston Latin school). A couple of years later, Harvard College was established in Cambridge.

For all their concern to be free to practice their own religious reformations, the Puritans were not supportive of reformations by others. Roger Williams, among many others, was banished from the Bay colony. In 1630 to 1658, several Quakers who refused to remain banished were hung, including Mary Dyer whose statue is on the grounds of the Massachusetts State House.

In 1638 Rev. Joseph Glover contracted with Stephen Daye for the two of them and their families to travel from England to Cambridge (in the colonies) along with a printing press, type, and printing materials, where Daye would be responsible for setting up and running the printing press in Glover’s home.<sup>3</sup> This printing press (the first in British North America<sup>4</sup>) was at least nominally operated under the auspices of Harvard. Glover died before their ship reached Massachusetts, and Daye carried out his contract for the widow Elizabeth Glover. Stephen’s son Matthew was also involved with the printing activity. Stephen was a locksmith and Matthew had apprenticed as a printer, so historians suspect Matthew did most of the actual printing. In any case, there was probably a lot of on-the-job learning about printing.

After printing a couple of other documents of which no copies remain, in 1640 Stephen Daye printed the so-called Bay Psalm Book, the first book written and printing in British America. In most of the churches in the Massachusetts Bay Colony, the Bay Psalm Book replaced the earlier Psalm books the Puritans had brought with them from England—hence the popular name “Bay Psalm Book.”<sup>5</sup>

In 1649 Matthew Daye died and Samuel Green took over the printing activity. Green also did a lot of on-the-job learning. By 1656 Green had two presses. For 40 years, this activity was the “press of Harvard College,”<sup>6</sup> although there was not really enough work over the years to keep Green working full time. Green

<sup>2</sup> Wilson00. <sup>3</sup> Section 1 is substantially based on Thomas70 and Blumenthal89; see also Reese89. <sup>4</sup> The church in colonial Mexico had a Spanish language printing press 100 years earlier. <sup>5</sup> Its actual title was *The Whole Booke of Psalmes Faithfully Translated into English Metre*. <sup>6</sup> According to Lawrence Wroth in his contribution to Lehmann-Haupt52, p. 8.

stopped printing in 1692. After Green, printing in Colonial Cambridge was finished.

Green had 19 children, 8 with his first wife and 11 with his second wife, and many of Green's descendants became printers, forming a dynasty of printers extending up and down the east coast.

In addition to no liberalism in religious practice, there was no freedom for printing (at least within the Massachusetts Bay Colony). The goal of the print shop operated by the Dayes and then the Greens was to support the church and the commonwealth.<sup>7</sup>

Some of the publications over the year of existence of the press were: <sup>6</sup> “a book of capital laws . . . ; small pieces relating to the scholastic activities of the college; annual almanacs; a second edition of the ‘Bay Psalm Book’; catechisms; a document relating to the troubles with the Narragansett Indians; a platform of the prevailing Congregational faith; and numerous sermons and doctrinal treatises.”

“The press reached the highest point of its activity with the publication in 1663 of John Eliot’s translation of the whole Bible into the Indian tongue . . . .”<sup>6</sup> This was a massive effort, producing over 1,000 copies, and requiring a special shipment of paper from England. It was the first Bible printed in the western hemisphere. On the title page, Samuel Green is listed as the printer, and his apprentice Marmaduke Johnson is also listed.

To slightly paraphrase Wroth, this Cambridge press did its job of being, over half a century, an intellectual force in a new and rude environment.

From 1674 on, printing was also allowed in Boston, on a case-by-case basis. Marmaduke Johnson received permission to print in Boston, but died before he could do any printing. Some of the following Boston printers were:<sup>8</sup>

- John Foster, 1676–1680; licensed to do printing, he was the first person who actually did printing in Boston.
- Samuel Sewall, 1681–1684: he was a bookseller, licensed to do printing, who printed acts and laws and books for himself and others; Samuel Green Jr. was his printer.
- James Glen, who also printed for Sewall before going out on his own.
- Samuel Green Jr., who printing work for himself and for booksellers and was allowed to continue printing after Sewall’s death; Green died in 1690.
- Richard Pierce, 1684–1690, the 5th Boston printer, who printed for himself and booksellers.
- Bartholomew Green, who first worked for his father in Cambridge and then took over his brother’s activity in 1690. In 1704 he started the *Boston News-Letter* for the postmaster, who somehow asserted a right to have a newspaper.

There were a number of other printers in Colonial Boston, i.e., between 1700 and 1775 when the Revolutionary War started. All this is detailed in Thomas’s book.

Looking beyond printing, there were no strong lines between trades. Printers worked for booksellers, booksellers did some of their own printing, printers published newspapers, binding was often a separate trade but not always, and printers publishing newspapers did some of their own writing.

Isaiah Thomas lists about 90 booksellers between 1641 and 1771.<sup>8</sup> Initially there was a bookseller or two in Cambridge. Next there were booksellers in Boston, particularly along the street and slope known as Cornhill. Hezekiah Ushel was the first in Boston, 1650–1771

Booksellers sold (and sometimes printed) acts and laws, books on religion, school books, books on politics, imported books, and new printings of books pirated from Europe. The shops of booksellers also often were community meeting places.

Colonial Boston also has a rich history of newspapers.<sup>8</sup>

<sup>7</sup> As I understand it, the original purpose of copyrights (circa 16th century) was to control printing of books. The authorities only gave the “right to copy” to a chosen few who were allowed to print only what the authorities liked. There was somewhat of an English tradition of freedom of expression, but this was primarily about no prior restraint. Post-speech or post-publication, the authorities could punish expression they didn’t like. <sup>8</sup> Thomas70.

- The *Boston News-Letter*, 1704–1776, was started by the Boston postmaster and printer John Campbell. This was the first newspaper in Colonies. It had lots of editors over the years, and was printed through the siege of the 1770s.
- . The *Boston Gazette*, 1719–1798, was started by the next postmaster, William Brooker. Apparently postmasters thought they had a right to have a newspaper. Brooker hired James Franklin to do the printing. The *Gazette* had a long line of of successor organizations
- The *New-England Courant*, 1721–1727, was started by James Franklin, who lost his job with the *Gazette* within a couple of years.

Isaiah Thomas lists another ten Boston newspapers prior to 1775, including his own *Massachusetts Spy*, published in Boston from 1770 through April 1775. I will touch on this more in the next section. 17

Before leaving the topic of Boston’s colonial newspapers, I’ll say a little about the most famous colonial Boston-trained printer, Ben Franklin, who was born on Milk Street and baptized at the Old South Meeting House, and attended Boston Latin on School Street for two years.<sup>9,10</sup> 18

Ben’s much older brother James had gone to England to apprentice as a printer. He returned in 1717 with a press and a small amount of type. James’s shop was the corner of Court Street and Franklin Ave. (called Queen Street and Dorset Alley in pre-Revolutionary times<sup>11</sup>). Ben was indentured at age 12 to his brother James to learn the printing trade. The indenture was to last until age 21. 19

James started the *New-England Courant* in 1721, and it was the first “truly independent newspaper in the colonies and the first with literary aspirations.”<sup>12</sup> For disagreeing in print with the authorities, James was imprisoned for a month in 1722 for what he published. He was released on the condition that “James Franklin not publish the *Courant*,” so Ben became the publisher in name. However, Ben couldn’t be the publisher if indentured to James, and officially Ben’s indenture was ended although a follow-on secret document of indenture was made.

Ben contributed a lot to the *Courant*, including 14 humorous letters over six months under the name of the widow Silence Dogood. However, Ben and James didn’t agree on things, and in 1723 Ben broke his secret indenture and went to Philadelphia, knowing that James could hardly admit that such a secret indenture existed.

By 1727, James, faced with continuing suppression of his press, had moved his printing business to Newport, RI. 20

## 2 Revolutionary War (1775–1783) and transition

The stamp act of 1775 was an incendiary event that produced resistance to British rule in the colonies. This was a tax by the British Parliament on printed materials in the colonies—the printed materials had to be produced on paper carrying an embossed revenue stamp. It was repealed a year later, but Parliament continued to assert its power to regulate the colonies and other taxes and regulations were imposed.

As resistance to British control increased, the colonial press participated and got in trouble. One of the printers involved in the resistance was Isaiah Thomas, to whose book I have frequently referred.<sup>13,14</sup> 21

Isaiah Thomas was born in 1749, but his widowed mother could not support him, and at age 6 he was apprenticed to a printer with an indenture to age 21. There he did both personal jobs for the childless printer and his wife and printing jobs. In particular, according to Thomas himself, he “set types, for which purpose he was mounted on a bench eighteen inches high, and the whole length of a double frame which contained case of both roman and italic.”

The printer was not too skilled, but Thomas quickly acquired the craft. A decade or so later, he escaped from his indenture and went elsewhere on the east coast to try to learn more about printing. By 1770 he was back in Boston, initially briefly in partnership with his former master. In 1771 Thomas started his own newspaper, the *Massachusetts Spy*. 22

As time went on and discontent with England grew in the colonies, Thomas used his *Massachusetts Spy* to support the causes of the Founding Fathers against England, and his print shop became known as the 23

<sup>9</sup> Franklin40. <sup>10</sup> Isaacson04. <sup>11</sup> Drake70. <sup>12</sup> Isaacson04. <sup>13</sup> Thomas70. <sup>14</sup> Blumenthal89.

“Forge of Sedition” as many resistance meetings were held there. Thomas himself refers to his press as the “sedition machine.” A 1774 edition of the paper shows the famous “join or die” slogan (first published in a cartoon and essay by Ben Franklin in Philadelphia), meaning that the colonies must join together or they would die separately.

24

Before April 19, 1775, the day of the British march on Lexington and Concord, Thomas smuggled a press out of Boston to Worcester, 40 miles to the west. He snuck out of Boston on April 18, 1775, and briefly joined the Lexington militia. Then two days later he traveled to Worcester where on May 3 he restarted publication of the *Massachusetts Spy*, including the first printed accounts of the Battle of Lexington and Concord.

25,26

After the war Thomas stayed in Worcester and, after some struggle, he began to develop a successful business. The business did well and Thomas became the “country’s leading printer, publisher, editor, and bookseller.”<sup>15</sup> In Worcester, he had a big printing plant, a bindery, and a paper mill; he had branch offices including in Boston and partnerships with a number of other publishing-world companies throughout the new country.

27

“Thomas retired in 1802 and devoted the rest of his productive and long life to collecting, scholarship, and philanthropy.”<sup>15</sup> He wrote the marvelous and comprehensive 1810 book, *The History of Printing in America*. “In 1812 he established the American Antiquarian Society to house his remarkable library of 8,000 volumes, with a mission to collect, preserve and make available the printed record of the United States for future generations. He served as president of AAS until his death in 1831.”<sup>15</sup>

Thomas’s Old No. 1 printing press, his first press in Boston, resides at the Antiquarian Society in Worcester.

28

After the Revolutionary War, the press was no longer regulated, and the technologies of the industrial revolution were applied in the publishing and printing business. I have just mentioned the example of Isaiah Thomas and his success.

29

Initially rotary presses were available and, eventually, much more efficient typesetting machines. Presumably using this technology, the *Boston Advertiser* was founded in 1813, Boston’s first daily newspaper.

30

However, in the years of the first third of the 1800s, Boston ceased to be the publishing center of what was now the new country. The big publishing centers were New York and Philadelphia. Other cities such as Baltimore, Cincinnati and New Orleans also developed strong publishing activities.

### 3 Liberal elites of the mid-19th century

31

Although Boston was no longer the country’s center of publishing, in 1800s Boston was the center of an important philosophical and literary movement.<sup>16,17,18,19,20</sup>

The Puritans came here as Congregationalists (see Figure 1), but individuals still needed to follow the doctrine and creed of their congregation (and to sign the Freeman’s Oath to be a citizen of the Massachusetts Bay Colony). As mentioned earlier, if someone wanted to promote some other version of religion, the Puritans kicked that person out of the Colony with lethal punishment for not staying out.

As the Revolution drew near, the churches largely favored the revolution, and I suppose in some sense this was at least a partial departure from the conservatism inherent in the colonial Puritan churches.

In the early 1800s, the Unitarian approach became more popular: people could be religious using their own reason and not reliance on doctrine and creed. Eventually Harvard appointed Unitarians as president and the divinity professor, and thus ministers coming out of Harvard were Unitarians, and in time a majority of the First Parish churches around the Massachusetts Bay region switched from Congregational to Unitarian.

Kings Chapel, across the street from the conference hotel, is a special case. Before the Revolution it was Anglican. After the Revolution it became Unitarian, but they liked their Anglican Book of Common Prayer and rewrote it to be consistent with Unitarianism. (Today they state, “We are Unitarian in theology, Anglican in worship service, and Congregational in governance.”)

<sup>15</sup> AAS12. <sup>16</sup> Barry12. <sup>17</sup> Mayerson80. <sup>18</sup> Phillips06. <sup>19</sup> Rose81. <sup>20</sup> Wilson05.

#### Church of England under the reign of King Charles (1625–1649)

- Church subordinate to state (Biblical justification); much hierarchy, ritual, and decoration between the individual and God; anti-Calvinism (predestination)

#### Puritans in the Massachusetts Bay Colony

- Congregational governance without ritual or decoration; theocratic state; Calvinist; Biblical literalism; exclusive

#### Unitarians

- Each individual could find Christian truth through intellectual freedom, reason and empirical evidence (e.g., from the Bible but without literalism); one God (not Trinity); no original sin or predestination; concern with moral and social harmony of a diverse population; anti-revivalist; use of writings, lectures, etc., to reach out

#### Transcendentalists

- Religion: intuition and spirituality rather than reason; Eclecticism instead of Christianity; practically, more of a philosophy and idealism than a religion
  - deliberately no component of being an organized religion
- Social movement: arguing for (and practicing) the goodness and self-reliance of man (partly as a reaction to urbanization and capitalism)
  - impractical
- Literature: believed literature was a contributor to the betterment of man; called for a new, American, literature
  - wrote some venerable works
  - attitudes evolved to acceptance of the progress of the Industrial Age and international trade (and acceptance of capitalism)
  - further development of literature and a cultural environment
- Broader world of social improvement
  - active in the abolition movement, women's right, education reform, and improving conditions of people in unfortunate circumstances

Figure 1: The evolution of spiritual doctrine from the Puritans to the Transcendentalists (my superficial approximation of the transitions).

All this thought about reason, individual goodness and personal relationship with god, etc., helped lead to the Transcendentalists and their idealism. The Massachusetts/Cambridge/Harvard liberal elites we hear so much about in politics today were in Cambridge, Boston, and Concord over 150 years ago.

In addition to their philosophy and idealism, the transcendentalists sought to create a new American literature.

One of the important meeting places was Elizabeth Peabody's bookstore, which still exists today (as a restaurant) on West Street near here. The plaque on the wall the building gives a good summary of the importance of Elizabeth Peabody and her bookstore:

Elizabeth Peabody, the first female publisher in Boston, maintained a home and business here in the 1840s. Her bookshop was the first in the city to offer books by foreign authors; and she published the periodical *The Dial* with Ralph Waldo Emerson. The shop was the meeting place for transcendentalists and intellectuals. Journalist Margaret Fuller [who lived on Winter Street, I believe, a couple of blocks away] gave lectures here called 'Conversations,' which were an important part of the early American feminist movement.

*The Dial: A Magazine for Literature, Philosophy, and Religion* was “an organ for the dissemination of Transcendental thought.”<sup>20</sup> The lectures were called “conversations” because women were not supposed to do public speaking in the 1840s. 33

Another meeting place was the Old Corner Bookstore, which in the mid- to -late-1800s was both a publishing location for and meeting places of the transcendentalists and other intellectuals. I’ll return to the Old Corner Bookstore in the next section. 34

Founded in 1885, the Saturday Club at the Parker House hotel (our conference hotel) was another meeting place of intellectuals. (Not only did they do a lot of writing, they apparently like to spend a lot of time in each other’s company talking about their thinking.) Here Charles Dickens gave a preliminary reading of the Christmas Carol to the Saturday Club before he did the public reading next door at the Tremont Temple. Here they conceived and started the *Atlantic Monthly* magazine in 1887. 35

It was in volume 1, issue 6, of the *Atlantic Monthly* that Oliver Wendell Holmes wrote in his series “Autocrat at the Breakfast Table” that Boston (specifically the State House) was the “Hub of the Solar System,” suggesting Boston was the center of everything commercial and intellectual at that time. The Massachusetts elites were not modest. At various times, the Old Corner Book Store and other famous downtown Boston locations have been called “the hub of the hub.”

## 4 Mid-1800s on, Boston is just another big American city 36

Despite Boston’s claim as a intellectual and cultural center, by the mid 1800s, it was just another big American City, in general publishing terms. The history leading up to this brings me back to the Old Corner Book store.<sup>21,22</sup> 37

Going down the left column of Figure 2, we have the history of the Old Corner Bookstore building, which in time became a building housing booksellers and publishers, including eventually the important publisher known as Ticknor and Fields. A little while after the *Atlantic Monthly* was established, Ticknor and Fields acquired the Old Corner Bookstore. Then, after Ticknor’s death, Fields moved the publishing business and the Atlantic to Tremont Street.

Also shown in the figure, coming out of the 1700s, were the predecessor organizations to Little Brown.

Finally (also on the figure), Henry Houghton, just out of college, started work as a printer, eventually acquired his own business; and he established the Riverside Press in 1852 in Cambridge where he also did printing for Little Brown and later for the *Atlantic Monthly*. During the economic downturn resulting from the Civil War, Houghton acquired book plates from various failing publishers, and eventually went into publishing himself (with Hurd as a New York partner), which caused Little Brown to drop Riverside Press as a printer.

In time the successor partners to Ticknor and Fields merged with Houghton’s company, George Mifflin joined and in time, with Houghton growing old and Mifflin by that time a partner, the company became Houghton Mifflin.

In the early 1900s, the editors of the *Atlantic Monthly* bought out the magazine from Houghton Mifflin.

From then until circa 1970, Houghton Mifflin and Little Brown were the “big two” Boston publishers, and the *Atlantic Monthly* was a Boston institution. At that point, these historic institutions to a considerable extent ceased to be publishers and became financial assets to be bought and sold. 38

Of course there were other publishers in Boston besides the Big Two, including specialty publishers. Two examples of specialty publishers are the Beacon Press (1854–present) and Daniel Berkeley Updike’s Merrymount Press (1893–1941).

The Beacon press was and is the publisher for the Unitarians, now UU, church. Its first location was on Bromfield, the street parallel to School Street behind the hotel. It later moved to Washington Street, and then to Beacon Hill. Its current building is on Joy Street, a block from the Beacon Street headquarters of the UUs. In the 19th century it primarily printed sermons and other books related to Unitarian theology. Since the 20th century it was printed many non-religious books consistent with its mission to publish works that “affirm and promote . . . the inherent worth and dignity of every person; justice, equity and compassion

---

<sup>21</sup> Hall10. <sup>22</sup> Tebbel72 and Tebbel75.



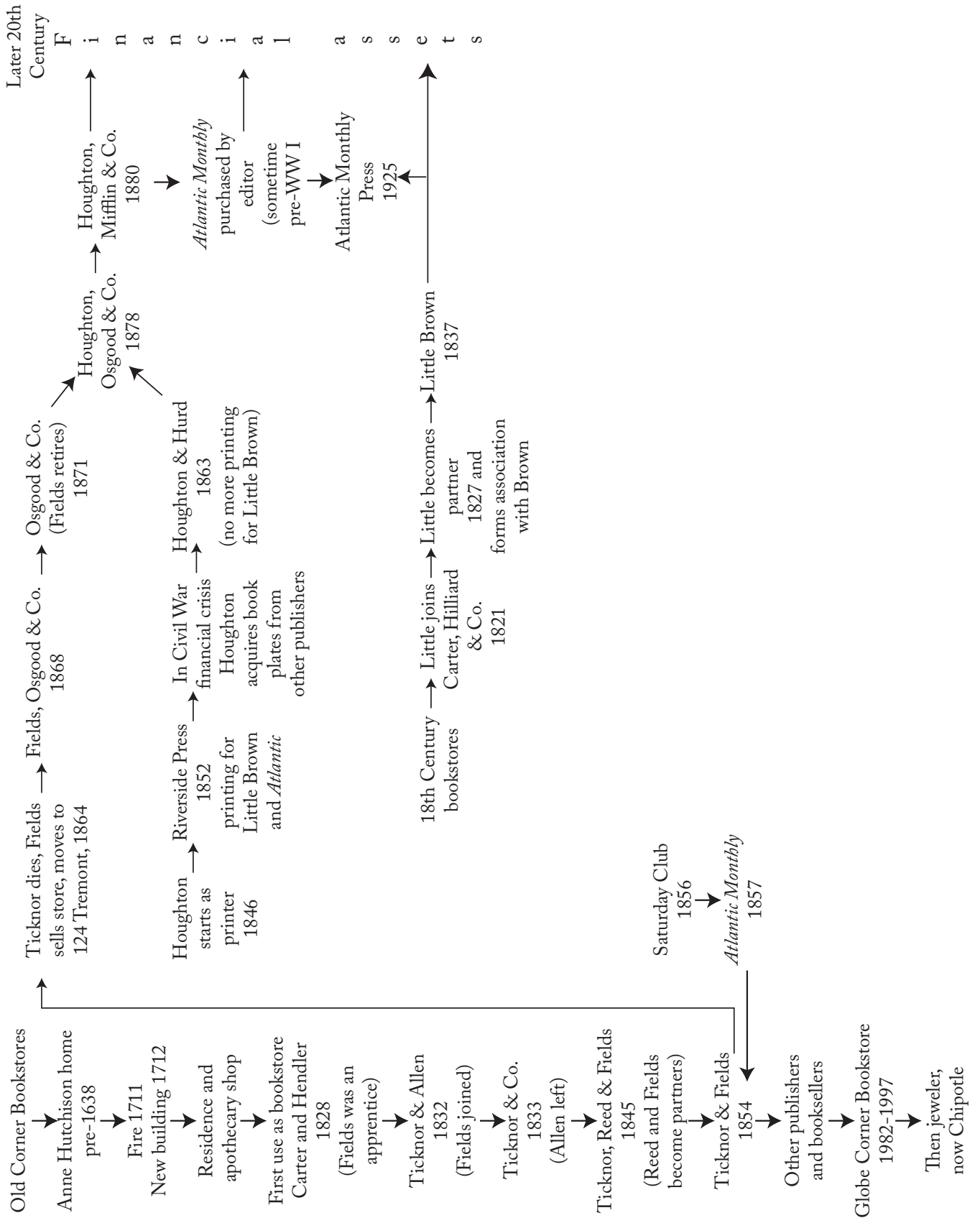


Figure 2: Some transitions in Boston publishing, from the middle years of the 19th century

in human relations; acceptance of one another; a free and responsible search for truth and meaning; the right of conscience and the use of the democratic process in society; the goal of the world community with peace, liberty, and justice for all; respect for the interdependent web of all existence; and the importance of literature and the arts in democratic life.”<sup>23</sup>

39

Daniel Berkeley Updike was a fine book publisher,<sup>14</sup> who had previously gained experience for over a decade at Houghton Mifflin and its Riverside Press.<sup>24</sup> Updike also was greatly interested in the history of printing types, and in 1922 published his classic book, *Printing Types, Their History, Forms and Use: A Study in Survivals*.<sup>25</sup>

40

There were lots of Boston papers in the 1800s and 1900s, for example: *Boston Daily Advertiser*, 1813; *Boston Journal*, 1833; *Boston Evening Traveler*, 1845; *Boston Herald*, 1846; *Boston Globe*, 1872; *Boston American*, 1904; plus smaller town papers. There were also lots of mergers and acquisitions, i.e., industry consolidation as the mid-1900s neared.

41

Boston’s newspaper row was on Washington Street, down the one-block length of School Street from the conference hotel. In the days before radio and TV were common, crowds stood in Washington Street to hear the latest news, e.g., of an election or prize fight.

42

An alley off of Washington Street a short way from the School Street intersection is named Pi Alley. Purportedly it is named Pi Alley because of all the newspapers in the area and the fact that a box of type dropped and scrambled on the floor was known as “pied type.”

43

In the colonial days, the printers typically had very little type—maybe one or two sizes, maybe not italic, maybe not two different typefaces. What type they did have came from Europe.

By 1800, Williams Caslon’s types had made it to the new country in many variations. Isaiah Thomas’s specimen book of types featured Caslon on its cover: “Being as large and complete an assortment as is to be met with in any one printing-office in America. Chiefly manufactured by that great artist William Caslon, Esq., of London.”<sup>14,26</sup>

Type foundries took a while to get going in the United States. The first successful one in Boston was the Dickinson type foundry which was founded in 1839. There were a few other foundries by the time of the great Boston fire of 1872, in which all the type foundries were destroyed.

There were five type foundries in Boston by the time of the American Type Founders (ATF) consolidation of 1892: “the Dickinson Type Founders, Boston Type Foundry, New England Type Foundry, Curtis & Mitchell Type Foundry, and the H.C. Hansen Type Foundry. The H.C. Hansen Type Foundry was started in late 1872 after the fire (Hansen had been an employee of Dickinson). The New England and Curtis & Mitchell foundries soon disappeared. The Dickinson and Boston foundries were absorbed by ATF. H.C. Hansen, later with his sons, remained in existence until 1922 as an independent type foundry.”<sup>27</sup>

No doubt a good bit of type design also went on in Boston. Two individuals well known for the type designs were Bruce Rogers and Bertram Grosvenor Goodhue.<sup>14</sup>

Goodhue (1869–1924) was an celebrated architect who also did book and type design, in particular the Merrymount type for the Merrymount press and Cheltenham type (original known as Boston Old Style).

44,45

Rogers (1870–1957) is viewed by some people as the greatest book designer of the 20th century. He worked at Riverside Press from 1895 to 1911 (he took over Updike’s position<sup>24</sup>) where he created many fine editions. Two of the types he designed are Montaigne (at Riverside Press) and Centaur (for New York’s Metropolitan Museum of Art). Centaur remains widely available (for example, it comes with Microsoft and Adobe products). I particularly like the slanted hyphen of Rogers’s original Centaur, which apparently and unfortunately is no longer slanted in the digital recreation. Rogers practiced so-called “allusive typography” where the type and ornamentation were matched to the content of the book.

<sup>23</sup> Beacon12. <sup>24</sup> Kelly11. <sup>25</sup> Barbara Beeton reports (2012-06-21 email) that “Updike’s archives from the Merrymount press are now in the special collections of the Providence Public Library; they include an incredible variety of specimen sheets and other material of interest to typographers and book designers.” <sup>26</sup> As an homage to the popularity of Caslon type in the eras under discussion (it was also used for the Declaration of Independence), I have used Adobe Caslon Pro for Figure 2 of this preprint and for the text of my presentation slides. <sup>27</sup> Devroye02.

## 5 Personal observations, 1964–present

46

This last section, covering the period of time I have lived in Boston and adjacent towns, is about my personal experience and observations rather than trying to cover printing and publishing history.

As I mentioned at the start of this note, Boston seemed a literary mecca when I first arrived here in 1964. In Harvard Square, near where I originally lived, the Harvard Coop and the Paperback Booksmith were my primary bookstores, but there were a variety of other stores selling new books in the Harvard Square neighborhood. Also, there seemed to be dozens of used bookstores. The Out of Town News in the middle of Harvard Square and Nini's Corner across the street had vast numbers of magazines and newspapers from around the world for sale.

I had also not previously been where I could easily use the libraries of multiple communities. Most of the time since I arrived in Boston I have had library cards for three different library systems, e.g., originally Cambridge, Boston, and Belmont (on my way to and from work). Also, because I originally worked for MIT, I could use its libraries and occasionally found a reason and way to access one of the Harvard libraries.

In my early years in the region, Boston also had many bookstores selling new books and even more used-book stores, it seemed, than in Cambridge.

47

I also liked the major newspaper options: *Globe* (more liberal), *Herald* (more conservative), and *Christian Science Monitor* (a wonderful 6-day-a-week paper providing unbiased world news). Over time some interesting weekly newspapers were founded as well as the African-American *Bay State Banner*.

48,49,50,51

As I got to know Boston, I also got to know the locations of such publishing institutions as the Riverside Press in Cambridge, Little Brown at the corner of Joy and Beacon Streets on Beacon Hill facing the Boston Common, the *Atlantic Monthly* on Arlington Street facing the Boston Public Garden, and Houghton Mifflin which in 1972 moved into the high rise building diagonally across the intersection from the conference hotel.

Much has changed in the nearly 50 years I have lived in or around Boston.<sup>28</sup> Sometime (circa 1980 perhaps) the big chains of bookstores began displacing the independent bookstores and small chains, for example, Barnes and Noble, Walden books, Border's, eventually Waterstones (from the UK) came to Boston.

A few years ago, Out of Town News stopped carrying most of the foreign newspapers it used to carry.

Also at some point, the big two publishers (Little Brown and Houghton Mifflin) seemed to become primarily financial assets to be bought and sold rather than institutions dedicated to publishing. The Riverside Press is gone from Cambridge, and the *Atlantic Monthly* is now part of a non-Boston publishing empire.

52

A notable exception to the decline of publishing in Boston is "David R. Godine, Publisher." Godine started his publishing business printing his own books in nearby Brookline. Dedicated to publishing quality books, the business has slowly grown. When I first became aware of it, it was based in the beautiful Victorian Ames-Webster mansion at the corner of Dartmouth Street and Commonwealth Avenue in Boston (it is now based about 400 feet from the conference hotel).

53

54

Another exception is International Data Corporation, now part of the International Data Group. This company publishes business data and 300 magazines in 85 countries. It founded the Dummy series of books (later sold to Wiley). Its headquarters is across the street from the Boston Public library in a high rise building (known locally as the Darth Vader building — not a compliment).

Of course, Godine and IDC are not the only positive notes, but they are an exception to the apparent general decline of publishing and printing in Boston.

As we have moved into the Internet era, the *Globe* and *Herald* newspapers were bought by out-of-town newspaper empires (the *Herald* has since become independent again but is struggling). The *Christian Science Monitor* became primarily an on-line newspaper. And most of the chain bookstores have succumbed to the competition of on-line bookstores such as Amazon and to the popularity of the e-book. With our present American culture of "no new taxes," the city and town libraries have cut their hours. None of this is exceptional for any major metropolitan area in the United States.

Nonetheless, Boston still remains a major urban, educational, and cultural center with some pretty nifty literary resources, particularly its libraries.

55,56,57,  
58,60

<sup>28</sup> An average of 3.5 miles from the conference hotel and never more than 8.3 miles.

Some notable libraries are at Harvard (founded 1636) libraries (with 80 libraries and 15 million books); the Boston Athenæum (founded in 1807 ); the New England Historical and Genealogical Society (founded 1845); Boston Public Library (founded 1848, the first large city public library in the country and the first circulating library,<sup>29</sup> now with many branches and 15 million books), and MIT (founded 1865) libraries (divided into several sub libraries with 3 million books).

In addition to being notable, these libraries are close enough together to require minimal travel time among them—so close together it is probably faster to take public transportation than to find parking. A walk passing each of them would only take about 90 minutes: one mile from the Athenæum to the NEHGS; two blocks from the Athenæum to the BPL; 1.5 miles from the BPL to MIT (1 miles as the crow flies across the Charles River, or as the Tech dinghy sails); and 1.7 miles from MIT to Harvard.

The youngest of these libraries is almost 150 years old. Obviously it is possible for a literary institution to withstand and adapt to the evolution of culture and economics, I suspect because they have never viewed profit as a key aspect of their missions.

Furthermore, since 1964, accessibility to library materials has become even easier. There were always many town libraries and libraries at the dozens of other colleges and universities in and around Boston. Now lots of the libraries are in library networks, where one can ask for a book at any of the libraries in the network to be sent to one's own library in that network. Also, there is the Massachusetts Virtual Library which supports exchange of books among networks. There is so much exchange going on that there is a company which has made a business of vans driving among the various libraries and networks of libraries doing deliveries and returns of exchanged books.

I am sure much of this is no different than what is going on in other states in the United States and in other areas around the world. Still, being based near Boston is particularly convenient for using libraries for research projects.

For bookstores, today Barnes and Noble is the only big chain still in Boston, with a store at the Prudential Center in Back Bay and operating some of the university and college bookstores. It is a little hard to find a general purpose independent store focusing on new books in Boston. Commonwealth Books and Brattle Book Shop primarily sell used books, and are within easy walking distance of the conference hotel.

In Cambridge, the MIT Press bookstore and the MIT Coop bookstore, selling new books, are across the street from each other. Moving on to Cambridge's Central Square and then Harvard Square, there are still a few used bookstores (e.g., Rodney's). Selling new books, in Harvard Square there are still the Harvard Book Store (founded in 1932 and never a part of Harvard), Grolier Poetry Bookshop (1927), and Schoenhof's Foreign Books (1856). Harvard Book Store and Grolier are still independently owned, in each case by a relatively new owner concerned that an institution not go out of business. Schoenhof's is no longer independently owned.

A particular favorite bookstore for me is the Harvard Book Store. It has no relationship to Harvard University except to have Harvard buildings on three sides of it. They have a large selection of new books and used books, and they offer bicycle delivery in Cambridge and nearby. They also have an Espresso Book Machine (EBM), which they have named Paige M. Gutenberg, for in-store on-demand printing of a customer's self-published book or millions of legally printable books from Google books, publishers, and other on-list books archives.<sup>30</sup>

A visit to the Harvard Book Store and their in-store book-printing machine brings us geographically full circle. The Harvard Book Store is a three minute walk from the location of Stephen Daye's original 1639 print shop. And by doing in-house printing, the bookstore has in some sense come full circle in the history of American printing and bookselling—back to its roots where booksellers often printed and published books and where printers sometimes had retail sales of books they printed.

In fact, I took the opportunity of visiting Paige M. Gutenberg to have a facsimile copy of the *Bay Psalms* printed for me. And that brings this talk full circle.

<sup>29</sup> Whitehill56. <sup>30</sup> The EBM is another publishing activity where Jason Epstein has been a prime mover. Epstein was previously editorial director at Random House for 40 years and personally edited many famous authors; he co-founded the *New York Review of Books*; he founded the Library of America line of books; and he published the *Reader's Catalog* in the mid-1980s to make 40,000 books available through a phone-call purchase.

## Acknowledgments

68

Steve Peter gave me pointers to useful books. Jeff Mayersohn of Harvard Book Store gave me pointers to people and places, loaned me books, and gave me a print-on-demand copy of the *Bay Psalms*. The librarians at the Boston Athenæum and Boston Public Library helped me find books. Karl Berry caught many typographical errors and made other editorial suggestions. Jeffrey Stanett and Ryan Shea Paré of First Printer restaurant answered questions and allowed me to take photographs of their printing-history artifacts.

## Bibliography and references

- AAS12.** American Antiquarian Society website, 2012: <http://www.americanantiquarian.org/>
- Barry12.** John M. Barry, *Roger Williams and the Creation of The American Soul: Church, State, and the Birth of Liberty*, Viking Press, New York, 2012.
- Beacon12.** Beacon Press website, 2012: <http://www.beacon.org/>
- Blumenthal89.** Joseph Blumenthal, *The Printed Book in America*, published for the Dartmouth College Library by University Press of New England, Hanover, NH, trade paperback edition, 1989 (reprint of the 1977 edition published by David R. Godine, Boston.).
- Drake70.** Samuel Adams Drake, *Old Landmarks and Historic Personages of Boston*, Singing Tree Press, 1970 (facsimile reprint of the edition published in 1900 by Little, Brown, and Company, Boston, which was the “new and revised” version of the 1974 original published by James R. Osgood and Company, Boston).
- Devroye02.** Luc Devroye website, 2002 (it’s wonderful; look at it): <http://luc.devroye.org/hansen/>
- Franklin40.** *The autobiography of Benjamin Franklin*, with postscript by Richard B. Morris, Washington Square Press, New York, 1940.
- Hall10.** David D. Hall, general editor, *A History of the Book in America* (5 volume set), the University of North Carolina Press, Chapel Hill, NC, 2010: Volume 1, *The Colonial Book in the Atlantic World*, edited by Hugh Amory and David D. Hall; Volume 2, *An Extensive Republic: Print, Culture, and Society in the New Nation, 1790–1840*, edited by Robert A. Gross and Mary Kelley; Volume 3, *The Industrial Book, 1840–1880*, edited by Scott E. Casper, Jeffrey D. Groves, Stephen W. Nissenbaum, and Michael Winship; Volume 4, *Print in Motion: The Expansion of Publishing and Reading in the United States, 1880–1940*, edited by Carl F. Kaestle and Janice A. Radway; Volume 5, *The Enduring Book: Print Culture in Postwar America*, edited by David Paul Nord, Joan Shelley Rubin, and Michael Schudson.
- Isaacson04.** Walter Isaacson, *Benjamin Franklin: An American Life*, Simon & Schuster Paperbacks, New York, 2004.
- Kelly11.** Jerry Kelly, *The Art of the Book in the Twentieth Century*, Rit Cary Graphic arts press, Rochester, NY, 2011.
- Krieger99.** Alex Krieger and David Cobb, editors, *Mapping Boston*, MIT Press, Cambridge, MA, 1999.
- Lehmann-Haupt52.** Hellmut Lehmann-Haupt (in collaboration with Lawrence C. Wroth and Rollo G. Silver), *The Book in America: A History of the Making and Selling of Books in the United States*, second edition, R. R. Bowker Company, New York, 1952.
- Mayerson80.** Joel Myerson, *The New England Transcendentalists and the Dial: A History of the Magazine and Its Contributors*, Fairleigh Dickinson University Press, Rutherford, NJ.
- Phillips06.** Jerry Phillips and Andrew Ladd, *Romanticism and Transcendentalism (1800–1860)*, Facts on File, Inc., New York, 2006.
- Reese89.** William S. Reese, The First Hundred Years of Printing in British North America: Printers and Collectors, <http://www.reese.com/papers/first100.htm>
- Rose81.** Anne C. Rose, *Transcendentalism as a Social Movement, 1830–1850*, Yale University Press, New Haven, CT, 1981.

**Tebbel72, 75, 78, and 81.** John Tebbel, *A History of Book Publishing in the United States*, R. R. Bowker, New York: Volume 1, *The Creation of an Industry*, 1972; Volume 2, *The Expansion of an Industry, 1865–1919*, 1975; Volume 3, *The Golden Age between Two Wars, 1920–1940*, 1978; Volume 4, *The Great Change, 1940–1980*, 1981.

**Thomas70.** Isaiah Thomas, *The History of Printing in America with a Biography of Printers & an Account of Newspapers*, M. M. McCorison editor, Imprint Society, Barre, MA, 1970 (new edition based on the two volume 1874 edition published by the American Antiquarian Society, Worcester, MA, which was derived from the original 1810 book by Thomas).

**Updike66.** Daniel Berkeley Updike, *Printing Types, Their History, Forms, and Use: A Study in Survivals*, Volume II, The Belknap Press of Harvard University Press, Cambridge, MA, 1966.

**Whitehill56.** Walter Muir Whitehill, *Boston Public Library: A Centennial History*, Harvard University Press, Cambridge, MA, 1956.

**Whitehill68.** Walter Muir Whitehill, *Boston: A Topographical History*, second edition, enlarged, Belknap Press of the Harvard University Press, Cambridge, MA 1968.

**Wilson05.** Leslie Perrin Wilson, “No Worthless Books: Elizabeth Peabody’s Foreign Library, *The Papers of the Bibliographical Society of America*, Volume 99, Number 1, 2005, pp. 113–152.

**Wilson00.** Susan Wilson, *Literary Trail of Greater Boston: A tour of sites in Boston, Cambridge, and Concord*, Houghton Mifflin, Boston, 2000.

**New from  
Center for the  
Study of  
Language  
and  
Information**



**Digital  
Typography**  
Donald E. Knuth  
Paper \$40.95  
*Second Printing*

**Companion  
to the  
Papers of  
Donald Knuth**  
Donald E. Knuth  
Paper \$35.00  
*Second Printing*

**Selected  
Papers on  
Fun and  
Games**  
Donald E. Knuth  
Paper \$40.00

**Selected  
Papers on  
Computer  
Science**  
Donald E. Knuth  
Paper \$30.00  
*Fourth Printing*

Distributed by the University of Chicago Press  
[www.press.uchicago.edu](http://www.press.uchicago.edu)

**Congratulations! You get a free ebook.**



PDF

Mobi

ePub

APK

DAISY



You get lifetime access to ebooks you purchase through oreilly.com, and whenever possible we provide them to you in five DRM-free file formats—PDF, ePub, Kindle-compatible .mobi, DAISY, and Android .apk—that you can use on the devices of your choice. Our ebook files are fully searchable, and you can cut-and-paste and print them. We also alert you when we've updated your ebooks with corrections and additions.

Visit [oreilly.com/go/ugebook](http://oreilly.com/go/ugebook) to register for your free ebook.

**O'REILLY**

©2011 O'Reilly Media, Inc. O'Reilly logo is a registered trademark of O'Reilly Media, Inc. All other trademarks are the property of their respective owners. 11331

---

**Do you need on-site training for L<sup>A</sup>T<sub>E</sub>X?**

Contact Cheryl Ponchin at

[cponchin@comcast.net](mailto:cponchin@comcast.net)

Training will be customized for your company needs.

Any level, from Beginning to Advanced.

---