ABSTRACT
The Phillimore Marriage Registers for England were published in the period 1896 to 1922 and have defined a standard layout format for the typesetting of marriage data. However, not all English parish churches had their marriage registers analysed and printed by the Phillimore organisation within this time period.
This paper tells the story of Wirksworth, a town in Derbyshire with a large church, licensed for marriages, yet whose marriage data was not released to the Phillimore organisation. Hence there is no printed Phillimore Marriages volume for Wirksworth. However, in recent years, a Wirksworth web site, created by John Palmer, has become famous as being probably the most comprehensive record of a parish’s activities anywhere on the Web.
Within a total of 120 MB of data on the web site, covering events in Wirksworth from medieval times to the present, is a set of data recording births, marriages and deaths transcribed from the original hand-written church register volumes.
The work described here covers the software tools and techniques that were used in creating a set of awk scripts to extract all the marriage records from the Wirksworth web site data. The extracted material was then automatically re-processed, typeset and indexed to form an entirely new Phillimore-style volume for Wirksworth marriages.

Categories and Subject Descriptors

General terms
Documentation, Languages

Keywords
Re-typesetting, Web-to-Print, troff, genealogy, hyperlinking, indexing
be the most comprehensive historical and contemporary record of a single parish, anywhere on the Web. Among approximately 128 MB of material on the Wirksworth site is a complete transcription, into hand-coded HTML, of the Wirksworth marriage registers, from 1608 to 1899. It should be explained that, in the years since the printed Phillimore volumes of the early 20th century, parish registers have generally become much more accessible to researchers. Since the enactment of the Parochial Registers and Records Measure, of 1987, most church registers have been moved to their respective County Record Offices. Those for Wirksworth now reside at the Derbyshire Records Office (DRO) in Matlock. Thus the transcription of the Wirksworth registers, by John Palmer, was conducted using microfilm copies made available by the Genealogical Society of Utah (GSU) supplemented by several visits to DRO to cross-check faded or near-illegible entries against the original hand-written registers.

3. The Web-based marriage entries
Register material on the Wirksworth site was initially generated from an MPRO database, but unfortunately this particular software dates back to the days of the MS DOS operating system. Although it now runs, without crashing, on more modern systems such as Windows XP this has been at the expense of losing some of its ability to generate subsets of the database (e.g. all the marriages) in various sorted orders and in various output formats. For these reasons, when it came to data gathering, the only way forward was to deal with the raw HTML coding on the Wirksworth Web site itself. The presentation of register information for marriages, christenings and burials, on this site, is understandably tied to searches based on family surnames. To increase the likelihood of success in the case of marriages, when details may be known for only one of the persons involved, a system of double entry is used whereby a marriage on a given date is indexed under both bride and the groom surnames.

3.1. First steps — merging and sorting
The infrastructure behind the Wirksworth web site registers is a very simple HTML tabular format. The various records comprising the table entries are all date tagged, but they make little or no use of more advanced HTML meta-notation. On any given date there may be records for Burials (B), christenings (C) and Marriages (M), all inter-
records into ascending date order via:

It now only remains to sort these pseudo-single-line

above, records are split over one or more lines. A standard

assist in the processing of multi-line records when, as

from each merged input record, the parenthesized places

tagscript

from

MADDOCK George(Wirksworth)/MELLOR Mary

M 1704apr27 MELLOR Mary(Wirksworth)/MADDOCK George

M 1704apr27 WOOD John(Biggin)/BRUKSHALL Mary

Now, it is a simple matter to reject any records beginning

with a B or a C, but then comes the trickier task of merging pairs of groom-first and bride-first marriage (M)

entries into a single record. To make things worse the

above example shows that it is common to have two or

more marriages on the same day and it will not always be

the case that bride and groom entries for a given marriage

will appear on successive lines, especially if the respective

wood surnames are far apart in the alphabet (as shown in the

WOOD-BRUKSHALL wedding, above).

It is the job of mergescript, seen at the lower right of

Figure 1, to identify and merge pairs of records relating to

the same marriage, though whether the single merged

record ends up as groom-first, or bride-first, is arbitrary; it

depends entirely on which of the surnames is earlier in the

alphabet. The towns, or villages, of origin for bride and

groom appear in parentheses in the Web site data and are

carried over into the output of mergescript, which for

the current example would now look like:

17040427 BRUKSHALL Mary(Biggin)/WOOD

John(Biggin)!!@

17040427 MADDOCK George(Wirksworth)/MELLOR Mary(Wirksworth)!!@

Note that the date of the marriage, at the start of each

record, has been converted into a totally numeric format to

enable date-based sorting to take place at a later stage. The

trailing !!@ is inserted as an end-of-record marker to

assist in the processing of multi-line records when, as

above, records are split over one or more lines. A standard

technique for joining lines with just such end-markers is

given in the classic AWK text[3].

It now only remains to sort these pseudo-single-line

records into ascending date order via:

sort +2n -3 mar-merged.txt > msorted.txt

yielding a large (approximately 2MB) file of all 9,380

marriages at Wirksworth from 1608–1899

3.2. Tagging and justifying

The bulk of the processing for Wirksworth marriages is
done by yet another awk script called tagscript. This

takes on the task of identifying whether a given merged

record is groom-first, or bride-first, by consulting a pre-declared set of strings, within tagscript itself, for male and

female first names. Every time an unknown first name is

encountered it is flagged up on the controlling terminal

window, so that it can be incorporated into an updated

version of tagscript.

A second awk script, called martabgen takes the output

from tagscript and takes the opportunity to parse out,

from each merged input record, the parenthesised places

of origin, for bride and groom, and to convert these into

standard Phillimore format. Other less common information

(usually contained in Comment: fields within the

original raw records) relates to previous marital status or to

occupation (e.g. ‘widow’, or ‘priest’) and is also carried

over.

Another major task undertaken by martabgen is to take

what may now potentially be a multi-line output record

and to split it according to the width of the first column in a

Phillimore marriage table entry (2.8 inches), which

accommodates approximately 40 characters of Caslon

Roman text at the point size in use (10 pt.). Each generated

line needs to be justified by adding or reducing inter-word

spacing. The default is to set a line slightly too tight rather

than over-stretching it. Every line after the first is ini-

tialised with six en-space characters to allow for the fact

that the second, and subsequent, lines are always indented

in Phillimore layout.

The martabgen script is set up to report an error if the

interleaved spacing on any line is too wide or too narrow.

Rather than incorporating a hyphenation algorithm into this

awk script, any necessary hyphenation is imposed by

hand for the tiny number of records (less than one in every

1000) that require it.

Each output record from the two scripts just described is

also invested with all of the tbl and troff coding necessary

for formatting the register pages (similarly to the job per-

formed by ascript in the earlier work described in [1]).

3.3. Final checking

The output from tagscript and martabgen looked

perfectly plausible as potential input to the common, hori-

zontal, processing chain of Figure 1, which begins by

invoking a script to add leader dots. However one small

further editing step was still needed.

The Palmer HTML database, extracted from the Wirks-

worth web site, is originally surname-sorted. Therefore

once it has been re-sorted into ascending date order it is

impossible to tell the ordering of the marriage services

themselves, if several of them took place on a single day.

By default same-day marriages are sorted alphabeti-

cally on grooms’ surnames. However, the precise actual

ordering is of great interest to local historians and is cer-

tainly made manifest in the registers themselves (and in

any microfilm copies of them). Fortunately, an independ-

ent, but unpublished, transcript of the Wirksworth Mar-

riage registers was discovered. This was made by Ivor

Neal of Middleton, a village near to Wirksworth.

The Neal database is sorted on a primary key of the mar-

riage date and a secondary key of the register ordering, for

any dates with multiple marriages. At the moment any cor-

rections to marriage ordering for the new Phillimore-style

Wirksworth volume are done by hand but consideration is

being given to generating this re-ordering automatically

from a comparison of the Neal database against the

tagscript/martabgen output.

Figure 2 shows the output of a typeset sample page from

the Wirksworth directory, generated by the processing

chain starting at the lower right of Figure 1. By sheer

chance the three marriages on 13th January 1705 really did
occur in accordance with the alphabetical order of groom surnames. But the three on 15th April did not, and the ordering has been corrected using the Neal database. Figure 2 also shows, on line 9, that it was necessary to hyphenate the place-name of ‘Alderwasley’.

1706]  
Wirksworth Marriages.

Thomas Dud & Elizabeth Weton, both of Idereshay ... ... ... 13 Jan. 1705
Cornelius Roper & Sarah Beardsley, both of Cromford ... ... ... 13 Jan. ..
William Smith & Mary Clouse, both of Ireton 13 Jan. ..
Robert Hill & Alice Buxton, both of this p. ... ... ... 17 Jan. ..
George Hodgkinson & Mary Bateman, both of Cromford ... ... ... 15 Apr. ..
Joseph Fox & Ann Needham, both of Alderwasley ... ... ... 15 Apr. ..
John Gregory & Sarah Stear, both of this p. ... ... ... 15 Apr. ..
Thomas Allen & Martha Kemp, both of this p. ... ... ... 19 Apr. ..
Thomas Wagstaff & Atlinah Gregory, both of Cromford ... ... ... 20 Apr. ..
Edmund Vallence & Elizabeth Smith, both of this p. ... ... ... 29 Apr. ..
William Short & Ruth Hops, both of Ashlehay ... ... ... 6 May ..
Stephen Wall & Ann Wood, both of this p. ... ... ... 5 June ..
Michael Clay & Elizabeth Barlow, both of this p. ... ... ... 1 July ..
John Simms & Dorothy Wean, both of Ashlehay ... ... ... 22 July ..
Thomas Wigley & Mary Keeling, both of Cromford ... ... ... 29 Aug. ..
William Valence & Elizabeth Fern, both of this p. ... ... ... 9 Oct. ..
John Tatum & Ann Clay, both of this p. ... ... ... 12 Oct. ..
John Mather & Elizabeth Wingfield, both of this p. ... ... ... 1 Nov. ..
Ralph Gell & Jane Hallworth, both of Middleton ... ... ... 7 Nov. ..
Robert Gothard & Grace Cawdale, both of Alderwasley ... ... ... 1 Dec. ..
John Smith & Mary Smith, both of Tisington ... ... ... 31 Dec. ..
Anthony Wilson & Alice Fone, both of this p. ... ... ... 1 Jan. 1706
Thomas Godbere & Mary Collinson, both of Middleton ... ... ... 9 Jan. ..
Robert Spencer & Elizabeth Hole, both of this p. ... ... ... 15 Jan. ..
Samuel Spencer & Ann Blackwell, both of this p. ... ... ... 15 Jan. ..

Figure 2: A sample typeset page of Wirksworth marriages

4. FINAL TESTING

At this stage the output from the processing of Wirksworth marriage records was ready to be fed into the common part of the processing chain in the centre of Figure 1. As related in [1] the precise leader dot patterns needed are extraordinarily sensitive to small variations in the set widths of characters on a given line. The crucial test was whether dotscript would cope adequately with input from a completely new processing chain in the lower part of Figure 1, and it certainly did so, as can be seen in Figure 2.

That being said the Wirksworth data did throw up some string widths that revealed, yet again, the need for sensitive fine-tuning when switching from one set of padding and leader patterns to another. A refined and revised version of dotscript, called wirkdots, was developed as a result of this new source of data. This refined version has now become the standard for all future work (as well as being retro-fitted into the source code for the previously produced Phillimore Register volumes).

Checks on the operation of the common indexing software, at the upper right of Figure 1, show that it copes flawlessly with the task of indexing Wirksworth entries.

5. CONCLUSIONS

In the real world, when transforming already-available material into another format, one seldom has any control of the way that the source material was created or published. That has certainly been the case for the current work. The test of success can only be whether the approach of pipelined software tools worked well for the new material and the extent to which existing tools from the previous work could be adapted and re-used. In these respects this latest work, which serves as a coda to the original investigations, can be accounted a success.

At present the Phillimore Registers for Derbyshire run to just 15 volumes, of which two have been republished as a result of the work described in [1]. The current work has been tentatively designated as Volume XVI to mark the fact that it is an unexpected extension of the original Phillimore Derbyshire series. When equipped with an index this new volume is likely to run to 500 pages or more. At present the first cut at typesetting and pagination is complete. The limiting step on progress is the need for patient cross-checking of the input data. It is hoped to publish Volume XVI in mid-2012

Acknowledgements

Grateful thanks are due to John Palmer whose patient and meticulous transcripts of Wirksworth marriage data, on his Web site, made the present project possible. The availability of Ivor and Valerie Neal’s independent transcript of Wirksworth marriages greatly eased the cross-checking of missing entries, and precise entry ordering, within the original registers. Finally, thanks are due to Margaret O’ Sullivan and James Davies, of DRO Matlock, who acted as a backup service for checking details of missing, or near-illegible, entries.

References


[2] David F. Brailsford, "Reconstructing typeset Marriage Registers using simple software tools", in Computer Science — Research and Development (Online first), Springer (22 December 2010). DOI: 010.1007/s00450-010-0145-x