
Access from the University of Nottingham repository:
http://eprints.nottingham.ac.uk/28127/1/eprint-dfb2011.pdf

Copyright and reuse:

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

- Copyright and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners.
- To the extent reasonable and practicable the material made available in Nottingham ePrints has been checked for eligibility before being made available.
- Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.
- Quotations or similar reproductions must be sufficiently acknowledged.

Please see our full end user licence at:
http://eprints.nottingham.ac.uk/end_user_agreement.pdf

A note on versions:

The version presented here may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the repository url above for details on accessing the published version and note that access may require a subscription.

For more information, please contact eprints@nottingham.ac.uk
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

1. INTRODUCTION

In previous work [1,2] a set of software tools was developed which enabled scanned pages of printed English Marriage Registers to be re-typeset, and indexed. The printed originals were published in the early years of the 20th century by the Phillimore company as an aid to genealogical researchers. These printed registers were, in turn, transcribed from the original hand-written registers, kept in local parish churches throughout England and Wales. For the purposes of the previous work the 15 Phillimore volumes relating to the county of Derbyshire were used as the test corpus. As a proof of concept the first two of these volumes have been retypeset and indexed. They have now been published and are available through Amazon (UK).

The idea of using a pipelined sequence of simple software tools, for the task of retypsetting whole Phillimore volumes from OCR scans of bitmap pages, was in the hope that the various software components could be re-used if other sources of marriage raw material came to light.

For reference, the original processing chain from scanned bitmap pages to re-typeset volumes is shown starting at the upper left of Figure 1. In this present paper we investigate how a new processing chain, shown in the lower half of Figure 1, was developed to cope with a Web-based source of information for an extra Derbyshire parish, not included in the original Phillimore set. This new source of information is able to feed into the existing processing chain along a common central spine (shown in bold type in the centre of Figure 1). Also included in that Figure is the indexing process common to both input chains.

2. The parish of Wirksworth

Wirksworth is a town in the county of Derbyshire, with a population of about 5000 in the town itself, rising to more than 9000 if immediately surrounding villages are included. It has a substantial parish church, set in its own grounds, with very much the air of a small cathedral.

The Wirksworth parish was not included in the original Phillimore Derbyshire series of registers. Of course, there are numerous other English parishes that were also not included, either because permission to copy the registers was not given, or simply due to lack of volunteers to carry out the transcription work.

A remarkable Web site for the Wirksworth parish was started in 1991, by John Palmer, and is available at http://www.wirksworth.org.uk. It is believed to
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's 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-
mixed. For example a search for records relating to 27th April 1704 gives:

C 1704apr27 BLOUNT Thomas=(son)Thomas/(Wirksworth)  
M 1704apr27 BRUKSHALL Mary(Biggin)/WOOD John  
M 1704apr27 MADDOCK George(Wirksworth)/PELLOR 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 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)/PELLOR 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 > marsedsorted.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 initialised 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 performed 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, horizontal, 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 Wirksworth 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 alphabetically on grooms’ surnames. However, the precise actual ordering is of great interest to local historians and is certainly made manifest in the registers themselves (and in any microfilm copies of them). Fortunately, an independent, but unpublished, transcript of the Wirksworth Marriage 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 marriage date and a secondary key of the register ordering, for any dates with multiple marriages. At the moment any corrections 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.

<table>
<thead>
<tr>
<th>Date</th>
<th>Groom</th>
<th>Bride</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 Jan. 1705</td>
<td>Thomas Dud &amp; Elizabeth Wetton, both of Idereshay...</td>
<td></td>
</tr>
<tr>
<td>13 Jan. 1705</td>
<td>Cornelius Roper &amp; Sarah Beardsley, both of Cromford...</td>
<td></td>
</tr>
<tr>
<td>13 Jan. 1705</td>
<td>William Smith &amp; Mary Clouse, both of Irecton...</td>
<td></td>
</tr>
<tr>
<td>17 Jan. 1705</td>
<td>Robert Hill &amp; Alice Buxton, both of this p...</td>
<td></td>
</tr>
<tr>
<td>15 Apr. 1705</td>
<td>Joseph Fox &amp; Ann Needham, both of Alderwasley...</td>
<td></td>
</tr>
<tr>
<td>15 Apr. 1705</td>
<td>John Gregory &amp; Sarah Steel, both of this p...</td>
<td></td>
</tr>
<tr>
<td>19 Apr. 1705</td>
<td>Thomas Allen &amp; Martha Kemp, both of this p...</td>
<td></td>
</tr>
<tr>
<td>20 Apr. 1705</td>
<td>Thomas Wagstaff &amp; Atlinah Gregory, both of Cromford...</td>
<td></td>
</tr>
<tr>
<td>29 Apr. 1705</td>
<td>Edmund Vallence &amp; Elizabeth Smith, both of this p...</td>
<td></td>
</tr>
<tr>
<td>6 May 1705</td>
<td>William Short &amp; Ruth Hops, both of Ashlehay...</td>
<td></td>
</tr>
<tr>
<td>5 June 1705</td>
<td>Stephen Wall &amp; Ann Wood, both of this p...</td>
<td></td>
</tr>
<tr>
<td>1 July 1705</td>
<td>Michael Clay &amp; Elizabeth Barlow, both of this p...</td>
<td></td>
</tr>
<tr>
<td>22 July 1705</td>
<td>John Simms &amp; Dorothy Wean, both of Ashlehay...</td>
<td></td>
</tr>
<tr>
<td>29 Aug. 1705</td>
<td>Thomas Wigley &amp; Mary Keeling, both of Cromford...</td>
<td></td>
</tr>
<tr>
<td>9 Oct. 1705</td>
<td>William Valence &amp; Elizabeth Fern, both of this p...</td>
<td></td>
</tr>
<tr>
<td>12 Oct. 1705</td>
<td>John Tatam &amp; Ann Clay, both of this p...</td>
<td></td>
</tr>
<tr>
<td>1 Nov. 1705</td>
<td>John Mather &amp; Elizabeth Wingfield, both of this p...</td>
<td></td>
</tr>
<tr>
<td>7 Nov. 1705</td>
<td>Ralph Gell &amp; Jane Hallworth, both of Middleton...</td>
<td></td>
</tr>
<tr>
<td>1 Dec. 1705</td>
<td>Robert Godthar &amp; Grace Cawdale, both of Alderwasley...</td>
<td></td>
</tr>
<tr>
<td>1 Jan. 1706</td>
<td>John Smith &amp; Mary Smith, both of Tisington...</td>
<td></td>
</tr>
<tr>
<td>1 Jan. 1706</td>
<td>Anthony Wilson &amp; Alice Fone, both of this p...</td>
<td></td>
</tr>
<tr>
<td>9 Jan. 1706</td>
<td>Thomas Godbere &amp; Mary Collinson, both of Middleton...</td>
<td></td>
</tr>
<tr>
<td>15 Jan. 1706</td>
<td>Robert Spencer &amp; Elizabeth Hole, both of this p...</td>
<td></td>
</tr>
<tr>
<td>15 Jan. 1706</td>
<td>Samuel Spencer &amp; Ann Blackwell, both of this p...</td>
<td></td>
</tr>
</tbody>
</table>

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, "Reconstituting 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