The PracTEX Journal, 2008, No. 3
Article revision 2008/11/16

Clinical trials management on the internet — II. Using LaTeX, PostScript, and SAS to produce barcode label sheets

Paul A. Thompson, Ph.D.

Abstract In clinical trials, it is often necessary to print labels with barcodes to identify samples. The availability of open-source tools for barcode management is still somewhat limited. Until recently, no LaTeX tools existed for the manipulation and encoding of barcodes. Using direct PostScript, barcode can be defined for strings to be printed on labels. Using LaTeX the labels can be queued up into appropriate sizes for specific label sheets, and tnen converted into `.pdf` files. Using SAS, the label sheets can be ordered in a web environment, queued up into appropriate files, and returned to users in a printable file.

Introduction

In the process of managing clinical trials, it is necessary to collect a variety of types of samples from participants. These samples might include blood, serum, plasma, and genetic samples, and tissues from biopsies. The types are limited only by the needs of the particular research project.

When samples are obtained, they must be processed. The samples are first placed into some container, which is labeled using an adhesive label which includes a barcode. The vial is then handled in a variety of ways, including heating, freezing, shaking, spinning, centrifuging, and others. The label is often very important. Such labels must include many types of information, including, but not limited to, the date of collection, the special number identifying the sample (the accession number), and other information as required by the protocol.

Once the sample is obtained, it is transfered into the special vial required for storage, and the label is applied immediately. This is required since many samples must be stored at very cold temperatures (-70 deg F in many cases). If the sample is not labeled before storage, it is often not possible to add the label, due to the adhesives; they stick fine when applied at room temperature, but have a great problem with adherence when applied directly to sample vials already at the storage temperature of -70 deg.

The production of label sheets in a timely, speedy manner is really important. For this reason, use of the internet for data management, as is done in the Division of Biostatistics, Washington University School of Medicine, needs to include the production of sheets of labels. The internet facilitates the distribution of information from the central data management location to distant locales.

In managing the trial, different visits can have sheets of labels for different types of samples. If these sheets are printed at the start of the trial, a very large number of such sheets will need to

be printed, filed, retained, tracked, and managed over the course of the trial. In one trial which is managed by the Division of Biostatistics (the HALT-PKD trial), there are 8 sheets of labels needed for a single individual, which need to be printed over a 7 year period. If all these sheets were printed in a gang-printing fashion at the start, it would pose a storage challenge to retain these sheets. Some would, in the vicissitudes of time, be lost, and need to be replaced. So, for these many reasons, printing the sheets of labels in a "just-in-time" manner is the most sensible approach for the management of the labels.

One approach to the production of labels with barcoded information uses LaTeX, `SAS`, and direct PostScript coding to produce such labels. This discussion demonstrates the manner of producing such labels, and introduces a new `pstricks` style, `pst-barcode`, which also can be used to do the process. This style, written in 2005, was not available when the barcoding process discussed in this paper was first devised, so this can be considered an update.

The basic approach to internet-based data management is described in a companion article (Thompson, 2008). In that discussion, the basic notions of internet data management using `SAS` and the `SAS/IntrNet` suite of products are described. The Web Data Entry System (WDES; Trinkaus et al. 2000, Thompson et al. 2002, Thompson 2003a, Thompson 2003b) is described as well. Using `SAS` and the WDES, the process of requesting both forms and labels can be supported using a well-developed infrastructure. The essential component to this approach is a scripting system which fulfills the function of both accessing information stored in some database system, forming up the printed information using a system which can be scripted (set up in a batch-processing mode). This function is fulfilled by the `SAS` system.

"Just-in-time" label sheet printing

Printing of sheets of labels for a trial often requires a "just-in-time" approach, where sheets of labels are generated, printed, and used within a short time in the life cycle of the trial. Using this approach, a participant who is due for an upcoming visit is selected, and sheets of labels are generated and printed. The labels are then used within a day or two, as samples are drawn and processed into storage vials.

The problem with the generation of labels is that the infrastructure to support the printing of them is not easy to find or build. To add to the complexity of the issues involved with forms printing (as discussed in Thompson, 2008), the labels generation process has the additional proble of the lack of generally available barcode production tools. Barcode tools are available from commercial tools, but these are often consequently expensive.

PostScript production of barcodes

PostScript is a language for document presentation, production, and dissemination. It is far less accessible than is LaTeX, although the two approaches to document preparation are quite similar to one another, since the level of coding is quite low in PostScript. PostScript is to LaTeX as assembler language is to Fortran. PostScript is the primitive while LaTeX is the high-level production language.

Production of sheets of labels

As part of the data management system for the HALT-PKD clinical evaluation of ACE/AARB medications for PKD, form packets are produced on the fly for participants for 20 different visits. For the baseline visits, a forms packet is produced with 2 sheets of labels, where each sheet has 36 labels; the second sheet has only a few spots printed. For later visits, form packets are produced with one or two sheets of labels. In many visits, no labels are printed.

Steps to sheets of labels packet production

The basic steps for production of sheets of labels are as follows:

1. User accesses forms request page on WDES portal. Participant ID, visit, and visit date are entered. Visit choice pre-populates forms selection, using `JavaScript` methods stored in a `JavaScript` file linked to the forms packet page. If a given form is not needed, it may be de-selected prior to packet production. Certain forms, if necessary for a visit, result in the generation of accession numbers, and consequently result in the generation of bar-coded labels.
2. Forms selection, with consequent label generation requirements, returns to the system.
3. Forms generation process determines if sample accession numbers are required for the visit. This is done by accessing the accession number database. Accession numbers are obtained and converted into `SAS` macro variables.
4. `SAS` macro program for generation of labels and printing them on sheets of labels is run. The information to be printed on the barcoded label sheets of labels is stored as values of macro variables, as are the ID numbers, dates, visits, and accession numbers. A file to build the sheet of barcoded labels is built by writing LaTeX code to a file, as follows:
   (a) Header code for the entire process;
   (b) LaTeX command macros to print the accession number and various types of printed text;
   (c) Closing text to complete the page.
5. A second script file is constructed which contains the commands to process the LaTeX file to the output `.pdf` file.
6. The final `.pdf` file is moved to a location on the system which is associated with a `url` address.
7. `SAS` then completes processing by writing a page which includes a link tag which identifies the forms packet encoded as a `.pdf` file.
8. Page is returned to user for forms access. Forms may be examined, saved locally, printed, or ignored.
9. `SAS` then completes processing by writing a page which includes a link tag which identifies the sheets of labels encoded as a `.pdf` file.
10. Page is returned to user for access to sheets of labels. Sheets of labels may be examined, saved locally, printed, or ignored.

Required LaTeX packages

There are several basic packages used for the sheets of labels production process. These packages are:

1. `geometry`: enables the dimensions of a page to be set in a simple interface.
2. `pstricks`, `pstricks-add`: The `pstricks` family offers a flexible method of inserting graphical information into a file. The `pspicture` environment allows graphical items (including text) to be inserted at any location on a page. Inside the `pspicture` environment, the `\rput` command inserts text at a given location, with optional choices about orientation.
3. `pst-barcode`: The `pst-barcode` substyle produces barcode output.
4. `lscape`: Produces landscape-oriented pages.

It is always possible to write barcode-generating code by writing PostScript code which performs the task at hand. With PostScript, this can be difficult, since the language is not particularly clear, and is at this time somewhat antequated. Thus, it is far more convenient to use a LaTeX style such as `pstricks`, with the associated substyle `pst-barcode`, to perform the task.

pst-barcode code

The `pstricks` base style and associated `pstricks`-related style `pst-barcode` offer a flexible approach to the construction of barcode blocks. Using this style, a large number of basic barcode representations can be handled by simple commands. Code block 1 shows the required code for barcodes of the "code39" or "code 3 of 9" format. In this case, `text` is presented using a block .75 in long, .5 in high. By default, the presented code is encoded with "*" symbols before and after the string, which are suppressed with `hidestars`.

```
1 \psbarcode{text}{width=.75 inkspread=1 hidestars height=.5}%
2   {code39}
```

Code Block 1: Code for form page production for one portrait-oriented page

LaTeX code for label sheet production

Using `SAS`, PostScript, and LaTeXtogether, sheets of labels can be produced using the algorithm defined above. The specific LaTeXcode required for the processing of a single sheet of labels page is shown in Figure 2. In this code, we see several things:

1. Line 1-6: Initialization of LaTeXjob, including package specification. `geometry` statements define a full page (8.5 in $\times$ 11 in) for use. Start up document processing. Converts orientation from portrait to landscape.
2. Line 7: Set several conditions.

3. Lines 9-14: LaTeX command defined which creates a single label. Options can easily be manipulated in this approach, since changing values in this single command definition defines all implementations of the command usage. In this implementation, the label to be converted into the barcode representation is also printed in the first line below the barcode. The `\parbox` specification defines a box with a defined width (92 pt) and heigth (96 pt) for a consistent presentation. Additionally, there is a LaTeX-defined skip between elements, and a LaTeX-defined skip between rows. The specification of the barcode block height and width (specified in the `\psbarcode` command options) and label height and width are defined by trial-and-error.
4. Line 17-51: A single page of labels is defined. The label sheet has 6 rows and 6 columns. Each column is defined by the width specification of the `\parbox`, and each row is defined by the height specification of the `\parbox`.
5. Line 52: Generate new page
6. Line 53-63: Second page of labels is defined. This page is not complete.
7. Lines 64-65: Complete file

```
1  \documentclass[letterpaper,10pt]{article}
2  \usepackage{geometry,pstricks,pst-barcode,lscape}
3  \geometry{letterpaper,tmargin=29pt,lmargin=1pt}
4
5  \begin{document}
6  \begin{landscape}
7  \pagestyle{empty}  \footnotesize  \newcommand{\spc}{20pt}
8
9  \newcommand{\ebxset}[7]{
10 \hspace*{#6}
11 \parbox[t][96pt][l]{92pt}{
12 \begin{pspicture}(3,.75in)
13 \psbarcode{#1}{width=.875 inkspread=1 hidestars height=.23}%
14   {code39}
15 \end{pspicture} \vspace*{2pt} \\ #1 \\ #2 \\ #3 \\ #4 \\ #5} }
16
17 \noindent
18 \ebxset{31027004}{Line2}{Line3}{Line4}{Line5}{1pt}
19 \ebxset{31027004-A}{Line2}{Line3}{Line4}{Line5}{\spc}
20 \ebxset{31027004-B}{Line2}{Line3}{Line4}{Line5}{\spc}
21 \ebxset{31027004-C}{Line2}{Line3}{Line4}{Line5}{\spc}
22 \ebxset{31027004-D}{Line2}{Line3}{Line4}{Line5}{\spc}
23 \ebxset{31027004-E}{Line2}{Line3}{Line4}{Line5}{\spc}  \\
24 \ebxset{71027444}{Line2}{Line3}{Line4}{Line5}{1pt}
25 \ebxset{71027444-A}{Line2}{Line3}{Line4}{Line5}{\spc}
26 \ebxset{71027444-B}{Line2}{Line3}{Line4}{Line5}{\spc}
```

```
\ebxset{71027444-C}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{71027444-D}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{71027444-E}{Line2}{Line3}{Line4}{Line5}{\spc}  \\
\ebxset{44332211}{Line2}{Line3}{Line4}{Line5}{1pt}
\ebxset{44332211-A}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{44332211-B}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{44332211-C}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{44332211-D}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{44332211-E}{Line2}{Line3}{Line4}{Line5}{\spc}  \\
\ebxset{68754231}{Line2}{Line3}{Line4}{Line5}{1pt}
\ebxset{68754231-A}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{68754231-B}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{68754231-C}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{68754231-D}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{68754231-E}{Line2}{Line3}{Line4}{Line5}{\spc}  \\
\ebxset{97860975}{Line2}{Line3}{Line4}{Line5}{1pt}
\ebxset{97860975-A}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{97860975-B}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{97860975-C}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{97860975-D}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{97860975-E}{Line2}{Line3}{Line4}{Line5}{\spc}  \\
\ebxset{35353421}{Line2}{Line3}{Line4}{Line5}{1pt}
\ebxset{35353421-A}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{35353421-B}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{35353421-C}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{35353421-D}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{35353421-E}{Line2}{Line3}{Line4}{Line5}{\spc}
\newpage
\noindent
\ebxset{65353421}{Line2}{Line3}{Line4}{Line5}{1pt}
\ebxset{65353421-A}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{65353421-B}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{65353421-C}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{65353421-D}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{65353421-E}{Line2}{Line3}{Line4}{Line5}{\spc} \\
\ebxset{83353421}{Line2}{Line3}{Line4}{Line5}{1pt}
\ebxset{83353421-A}{Line2}{Line3}{Line4}{Line5}{\spc}
\ebxset{83353421-B}{Line2}{Line3}{Line4}{Line5}{\spc}
\end{landscape}
\end{document}
```

Code Block 2: Code to produce label sheets in LaTeX

Script file to process LaTeXfile

The LaTeX file, as defined in Figures 1 and 2, is processed using the script file shown in Code Block 3. Although each command can be run directly as a "shell-out" command, it is better to write a script file, and then "shell-out" to process the script file. This allows the user to check for errors by running the command file separately from the system, when things do not work correctly. This allows for much easier debugging.

```
1  #!/bin/sh
2  cd /users/pkdtn/IntrNet/sasfiles/forms/setup
3  'mv' mactest_XX_F5.tex hmactest_XX_F5.tex
4  'rm' mactest_XX_F5.*
5  'cp' hmactest_XX_F5.tex /users/pkdtn/private_html/forms/
6        mactest_XX_F5.tex
7  'cp' bmactest*.ps /users/pkdtn/private_html/forms
8  cd /users/pkdtn/private_html/forms
9  /usr/bin/latex mactest_XX_F5
10 /usr/bin/latex mactest_XX_F5
11 /usr/bin/dvips -o mactest_XX_F5.ps mactest_XX_F5
12 /usr/bin/ps2pdf mactest_XX_F5.ps gXX_F5.pdf
13 chmod 777 /users/pkdtn/private_html/forms/gXX_F5.pdf
```

Code Block 3: Code for form package production

Producing a sheet of labels

The code to produce a sheet of labels is shown above in Figures 1,2, and 3. The steps to the production of a sheets of labels packet are shown above in the section "Steps to sheets of labels packet production". The final information needed is the `SAS` code needed for sheets of labels packet production. The `SAS` code is shown in Figure 4.

Here is a brief description of some of the code:

1. Lines 4: The `LIBNAME` statement defines a reference to the database location
2. Lines 8-13: Access database containing accession numbers. Using the function `PROC SYMPUT`, create macro variables `_an1` and `_an2` to store these numbers. When a macro variable is created, it can be "resolved" (variable reference replaced by value) using the ampersand form.
3. Line 20: Start-up information for the barcode sheet is stored in a file. This file is used to initiate the barcode processing.
4. Lines 21-34: Label specifications are written out to the file. The `FILE` statement designates the file to be used. The `PUT` statement writes information to that file. The commands contain the specification `&_an1` and `&_an2`, which converts the macro variable specifications to the

values stored in the macro variables (this process is termed "macro resolution"). The trailing "." is the macro variable delimiter, and is eliminated during the macro resolution process.

5. Lines 35-39: Complete file for LaTeX processing.
6. Lines 43-50: Create a script file to process the LaTeXcode. Generally it is best to use absolute paths rather than relative paths to ensure that the system uses the correct version of the program, and that no issues arise about location of programs.
7. Lines 55-56: Set permissions and process script file. The command `X "command"` executes the command in the quotes at the operating system level.
8. Lines 60-82: Create and return the page containing the reference to the `.pdf` file. The `_webout` specification indicates that the `html` file being created is returned to the originating user.

```
1  /* ********************************************************** */
2  /* SAS symbolic directory name                                */
3  /* ********************************************************** */
4  LIBNAME HPKD "/users/data/halt";
5  /* ********************************************************** */
6  /* Read resource file and convert accession numbers to macros */
7  /* ********************************************************** */
8  DATA _NULL_;
9       SET HPKD.ACCNFORM;
10      WHERE (HALTID = "&_haltid");
11      IF (SAMP = 2) THEN CALL SYMPUT("_an1",PUT(ACCN,8.));
12      IF (SAMP = 3) THEN CALL SYMPUT("_an2",PUT(ACCN,8.));
13 RUN;
14 /* ********************************************************** */
15 /* Create file containing PostScript code to produce sheet    */
16 /* Copy macro file from barmacs.ps to current file            */
17 /* Add new lines                                              */
18 /* Complete file with terminus lines                          */
19 /* ********************************************************** */
20 x "cp /users/halt/scripts/barmacs.tex gXX.tex";
21 DATA _NULL_;
22      FILE "/users/halt/scripts/gXX.tex" MOD;
23      PUT "\ebxset{&_an1.}{Line2}{Line3}{Line4}{Line5}{1pt}";
24      PUT "\ebxset{&_an1.-A}{Line2}{Line3}{Line4}{Line5}{\spc}";
25      PUT "\ebxset{&_an1.-B}{Line2}{Line3}{Line4}{Line5}{\spc}";
26      PUT "\ebxset{&_an1.-C}{Line2}{Line3}{Line4}{Line5}{\spc}";
27      PUT "\ebxset{&_an1.-D}{Line2}{Line3}{Line4}{Line5}{\spc}";
28      PUT "\ebxset{&_an1.-E}{Line2}{Line3}{Line4}{Line5}{\spc}\\";
29      PUT "\ebxset{&_an2.}{Line2}{Line3}{Line4}{Line5}{1pt}";
30      PUT "\ebxset{&_an2.-A}{Line2}{Line3}{Line4}{Line5}{\spc}";
```

```
     PUT "\ebxset{&_an2.-B}{Line2}{Line3}{Line4}{Line5}{\spc}";
     PUT "\ebxset{&_an2.-C}{Line2}{Line3}{Line4}{Line5}{\spc}";
     PUT "\ebxset{&_an2.-D}{Line2}{Line3}{Line4}{Line5}{\spc}";
RUN;
DATA _NULL_;
     FILE "/users/halt/scripts/gXX.tex" MOD;
     PUT "\end{landscape}";
     PUT "\end{document}";
RUN;
/* ********************************************************** */
/* Produce script file for subsequent processing              */
/* ********************************************************** */
DATA _NULL_;    * Make file for sheets of labels packet script;
     FILE "/users/halt/scripts/curscript";
     PUT "#!/bin/sh";
     PUT "/usr/bin/latex gXX";
     PUT "/usr/bin/latex gXX";
     PUT "/usr/bin/dvips -o gXX.ps texfile";
     PUT "/usr/bin/ps2pdf gXX.ps gXX.pdf";
RUN;
/* ********************************************************** */
/* Set script file with proper permissioning, and execute     */
/* script file from within SAS                                */
/* ********************************************************** */
x "chmod 550 curscript";   * Set permission to executable ;
x "curscript";
/* ********************************************************** */
/* Prepare web page and transmit to calling program           */
/* ********************************************************** */
DATA _NULL_;
     FILE _WEBOUT;    _WEBOUT is special web-destination name;
     PUT '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">';
     PUT '<HTML><HEAD><TITLE>Label Sheet</TITLE>';
     PUT '<SCRIPT SRC="/haltpkd/java/mainjava.js"></SCRIPT>';
     PUT '<SCRIPT SRC="/haltpkd/java/formjava.js" ></SCRIPT>';
     PUT '<LINK HREF="/haltpkd/css/maincss.css"></LINK>';
     PUT "</HEAD><BODY>";
     PUT '<FORM ACTION="/cgi-bin/haltpkd/broker"  '
         'METHOD="POST" name="formform">';
     PUT '<INPUT TYPE="HIDDEN" NAME="_brkname" '
         'VALUE="/cgi-bin/haltpkd/broker">';
     PUT "<BR>";
```

```
     PUT '<INPUT TYPE="HIDDEN" NAME="_program"
          VALUE="frmmain.sas">';
     PUT '<INPUT TYPE="HIDDEN" NAME="_service" VALUE="haltpkd">';
     PUT '<INPUT TYPE="HIDDEN" NAME="_debug" VALUE="0">';
     PUT '<input TYPE="HIDDEN" NAME="_prtype"     VALUE="2">';
     PUT "<H3>HALT-PKD Participant: ID &_haltid,
          Visit &_visit</H3><BR>";
     PUT "<H3>Current sheets of labels package</H3><BR>";
     PUT '<A HREF="/haltpkd/sheets of labels/gXX.pdf">';
     PUT "Sheets Of Labels packet in PDF format</A><BR><BR>";
     PUT "</FORM></BODY></HTML>";
RUN;
```

Code Block 4: SAS Code for overall packet production

Conclusions

Methods for management of production of sheets of labels over the web require the integration of technologies for database management, text production, and web technologies. The entire process must be integrated together using a scripting tool that can facilitate the three components noted above. The SAS system provides a convenient platform for all of the components except the text production. The text production component can be nicely managed using LaTeXand documents encoded using the encapsulated PostScript format. Using the scripting capabilities of the SAS system, the entire process of

1. making a request for a sheet of labels;
2. getting information from a database to support the request;
3. producing the code to make the sheets of labels; and
4. returning the final sheets of labels packet to the requesting user

can be accomplished within a single program, in real time. Timing estimates are shown in Thompson (2008).

References

Thompson, P. A. (2003a). The Web Data Entry System: Reference Guide, Version 2.5.

Thompson, P. A. (2003b). The Web Data Entry System: User's Guide, Version 2.5.

Thompson, P. A. (2008). Clinical trials management on the internet — I: Using LaTeX and SAS to produce customized forms. *Practical TEX*, Submitted.

Thompson, P. A., S. A. Littlewood, A. J. Adelman, and J. P. Miller (2002). The Web Data Entry System: Methods for web data entry and SAS data management. In SAS Institute Inc (Ed.), *Proceedings of the 27th Annual SAS Users Group International Meeting*. Cary, NC, USA: SAS INstitute Inc.

Trinkaus, K. M., P. A. Thompson, and J. P. Miller (2000). Using SAS/IntrNet software to support distributed data entry. In SAS Institute Inc (Ed.), *Proceedings of the 25th Annual SAS Users Group International Conference*, pp. 1212–1217. Cary, NC, USA: SAS Institute Inc.