J Pathol Inform

Technical Note

Reqscan: An open source solution for laboratory requisition scanning, archiving and retrieval

Eviatar Bach[1], Daniel T. Holmes[2,3]

[1]Faculty of Arts and Science, University of British Columbia, 2329 West Mall, Vancouver, BC, V6T 1Z4, Canada, [2]Department of Pathology and Laboratory Medicine, St. Paul's Hospital, 1081 Burrard St., Vancouver, BC, V6Z 1Y6, Canada, [3]Department of Pathology and Laboratory Medicine, University of British Columbia, Rm. G227-2211 Wesbrook Mall, Vancouver, BC, V6T 2B5, Canada

E-mail: *Daniel T. Holmes - dtholmes@mail.ubc.ca
Corresponding author*

Received: 21 September 2014 Accepted: 23 November 2014 Published: 29 January 15

This article may be cited as:
Bach E, Holmes DT. Reqscan: An open source solution for laboratory requisition scanning, archiving and retrieval. J Pathol Inform 2015;6:3.

Available FREE in open access from: http://www.jpathinformatics.org/text.asp?2015/6/1/3/150256

Abstract

Requisition storage and retrieval are an integral part of the outpatient laboratory testing process. It is frequently necessary to review an original requisition to confirm the ordering physician, patient demographics, diagnostic information, and requested tests. Manual retrieval of a paper requisition is time-consuming and tedious. Although commercial solutions exist for the scanning and archiving of barcoded paper requisitions, the tools to accomplish this are freely available from the open source software community. We present a simple dedicated piece of software, Reqscan, for scanning patient laboratory requisitions, finding all barcode information, and saving the requisition as a portable document format named according the barcode(s) found. This Python application offers a simple solution to patient requisition digitization. Reqscan has been successfully tested and implemented into routine practice for storage and retrieval of outpatient requisitions at St. Paul's Hospital, Department of Pathology and Laboratory Medicine in Vancouver, British Columbia, Canada.

Key words: Barcode, digitization, electronic medical record, laboratory, patient record, requisition, scanning

Access this article online
Website:
www.jpathinformatics.org
DOI: 10.4103/2153-3539.150256
Quick Response Code:

INTRODUCTION

Requisition storage and retrieval are an integral part of any routine or specialized outpatient clinical laboratory. Patients receive a paper requisition from their physician indicating the required tests, and the patient brings it to a satellite bleed-station or a hospital for specimen collection. The process of specimen collection involves a number of individuals: The ordering physician or their surrogate, data entry clerk, phlebotomist, and laboratory assistant or technologist. As such, there is ample room for errors at the level of order-entry.

Outpatient requests to our laboratory come exclusively in the form of paper requisitions. These are either brought to the hospital bleed-station or, for referred-in specimens, come from another laboratory along with the collected specimen. If the patient has no record in our laboratory information system (LIS), a record is created, the specimens are then electronically recorded as "arrived," a barcode label is created for each tube and then the specimen is appropriately routed for testing in high volume and/or specialized areas as needed. In the past, when there was a question about the possibility of order-entry error (misidentification of ordering physician, demographic information in question, unusual test-request or test combination, question from referring laboratory, possible specimen mix-up), we would manually retrieve the paper requisition from filing cabinets located in the accessioning area by date. These were stored in the order they were received. After a period of about 6 weeks, these requisitions are placed into a long-term storage facility for a period required by the accreditation body. As we receive up to ≈ 1000/day, the task of finding a requisition is laborious. However, since a one-dimensional barcode label with the patient's unique provincial health number (PHN) is automatically generated along with the barcodes produced for each of the collected samples, it seemed natural to apply this label to each requisition and build a solution to electronically store the requisitions for more rapid retrieval.

While a number of commercial products for requisition storage and retrieval exist, our budgetary constraints did not permit us to pursue these options. Given the simplicity of our needs, namely, to be able to find a requisition for confirmation of demographic, diagnostic and test-request information, it seemed reasonable to use a number of available open-source tools to create a program capable of scanning a patient requisition and storing the it by according to the content of any barcode(s) present.

The goal of this project was to produce a simple, entirely open-source tool solution to this problem.

APPROACH

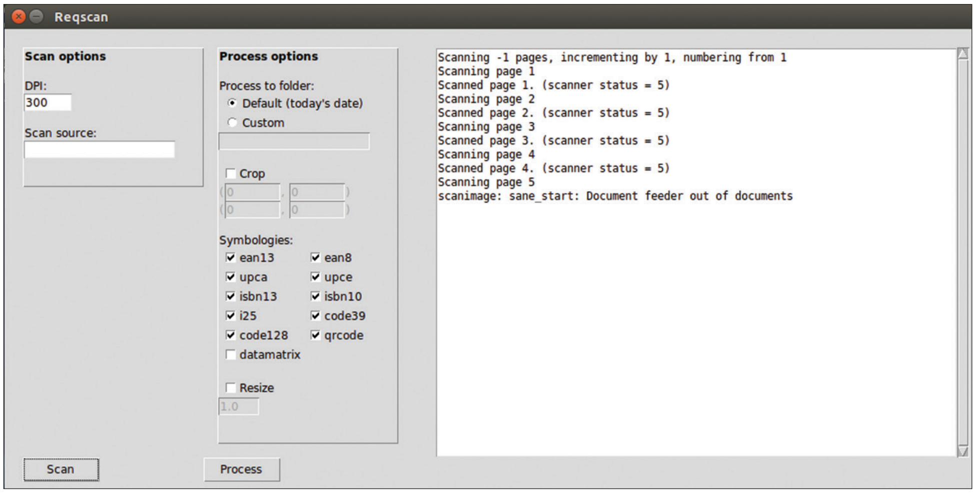
The software we developed, dubbed "Reqscan," requires an automatic document feed (ADF) scanner connected to a computer running GNU/Linux. It is written in Python, a cross-platform scripting language, which functions as a "glue language" to interface between the separate components. Python and all other required software tools for this project (so-called "dependencies"), are free and open-source.

Reqscan interfaces with scanners using Scanner Access Now Easy (SANE), a public domain application programming interface which supports thousands of devices. The program scans all pages in the ADF source (which can be specifically selected if there is more than one; for example, a front and rear loader), and the files are saved to a temporary directory as TIFF files according to the scanning resolution chosen. They are then processed using ZBar,[1] a barcode scanning library, and saved in portable document format (PDF) in a folder named after the current date, with file names consisting of the barcode data. If more than one barcode is found on a given page, the requisition is saved as multiple separate PDFs each having a unique barcode name corresponding to the barcode content. If the same barcode is found on different pages, the files are saved as *(barcode_ content)*.pdf, *(barcode_content)*_2.pdf, *(barcode_ content)*_3.pdf etc., where *(barcode_ content)* represents the encoded information.

Errors (such as records with missing barcodes) are handled robustly, such that jobs with hundreds of records will not terminate due to an error with individual files. The scans of the failed files are saved as individual PDF files named failed_1.pdf, failed_2.pdf etc., These can easily be concatenated for rapid review if desired as we discuss below.

Both a command-line and a graphical interface [Figure 1] are available. The process is split into two tasks: Scanning and processing. This allows multiple scan batches to be done prior to processing.

Reqscan supports all the barcode symbologies supported by ZBar (EAN-13, EAN-8, UPCA, UPCE, ISBN-13, ISBN-10, i25, Code 39, Code 128, and QR). We have also undertaken a preliminary implementation of Data Matrix



DPI: 300

Scan source:

Process to folder:
● Default (today's date)
○ Custom

□ Crop
(0, 0)
(0, 0)

Symbologies:
☑ ean13 ☑ ean8
☑ upca ☑ upce
☑ isbn13 ☑ isbn10
☑ i25 ☑ code39
☑ code128 ☑ qrcode
□ datamatrix

□ Resize
1.0

```
Scanning -1 pages, incrementing by 1, numbering from 1
Scanning page 1
Scanned page 1. (scanner status = 5)
Scanning page 2
Scanned page 2. (scanner status = 5)
Scanning page 3
Scanned page 3. (scanner status = 5)
Scanning page 4
Scanned page 4. (scanner status = 5)
Scanning page 5
scanimage: sane_start: Document feeder out of documents
```

Figure 1: Reqscan window displayed on Ubuntu Linux 14.04

barcodes using libdmtx but, unlike the other symbologies implemented through ZBar, currently this is not searching for multiple Data Matrix barcodes per page.

Retrieval of previously scanned requisitions is a simple matter of searching the file system (or an appropriately dated folder) for a PHN using GNU/Linux built-in file searching commands (e.g. “find” or “locate”) or search options in GUI file managers.

Reqscan is available for download for free at https://github.com/eviatarbach/reqscan under the GNU General Public License Version 3.

RESULTS

Reqscan has been implemented at St. Paul’s Hospital in Vancouver, BC, Canada since January of 2013, scanning all incoming outpatient requisition forms daily. Between early January and mid-March 2013, the second author invested approximately 10 h implementing customizations specific to the work-flow of his laboratory setting. Specifically, barcode data containing undesirable characters, e.g.,: %, |, and, $, #, and ^ were removed by renaming using Linux command-line shell scripts. Additionally, a short shell script using the open source tool, PDF toolkit (PDFtk),[2] to concatenate pages named “failed_1.pdf”, “failed_2.pdf”… “failed_n.pdf” into a single file was undertaken.

Requisitions were noted to be highly heterogenous. Requisitions from the in-hospital outpatient phlebotomy station were generally hand-written original from the ordering physician. These are barcoded in our laboratory accessioning with a label containing the patient PHN in Code 128 symbology. Those from in-hospital clinics are also the original but may contain one or more barcodes associated with the work-flow of the clinic. These were also re-barcoded by our laboratory with the PHN. Requisitions accompanying referred-in specimens were provided as: (1) A photocopy of the original (2) a single-page printed request for a single patient (without a photocopy of the original requisition) stating the patient’s identifying information, off-site sample collection information (e.g. ordering and cc physicans, collection time, receive-time, shipment date, accession number, test requests) (3) As in situation (2) but with a stapled appended photocopy of the original requisition (4) A single page containing printed requests for multiple patients and the accompanying off-site sample/demographic information but without copies of the original requisitions.

Our label-printing process for accessioning specimens using SunQuest v 6.4 (Tucson, AZ) is currently set to print container ID barcode labels for all collected specimens from a given patient, but only one barcode referring to the patient’s PHN. For this reason, single-page requisitions referring to a single patient (original, photocopy or printed) are barcoded with a single PHN label. Multipage requisitions for a single patient are barcoded with the PHN on their first page only. Single-page requisitions referring to multiple patients are barcoded with multiple labels for the PHNs of all patients to which the requisition refers.

In our situation, we set Reqscan to search for Code 39 and Code 128 only, as these are the symbologies used by our laboratory and some onsite clinics. Numerous other barcode symbologies may appear on requisitions sent to us from other hospitals, clinics or private laboratories, but these are set to be ignored since the barcode content may refer to patient identifiers or sample accession numbers of other hospitals or their LIS.

A 1-week period from Sunday October 5 to Saturday October 11, 2014 was selected to examine barcode retrieval success rates. In that week, 4,653 individual sheets were scanned at 300 DPI. Reqscan found 4,528 (Code 39 and Code 128) barcodes and corresponding PDF files were created. There were 1,025 pages on which no barcode was found. These are identified as “failed_1.pdf,” “failed_2.pdf” etc. These files were visually inspected for barcode labels that should have been identified under ideal circumstances. Thirty-five of the 1,025 pages did contain barcode information that Reqscan should have picked up. This meant that a total of 4,528 of 4,563 available barcodes were successfully identified over 4,653 pages making a success rate of 97.3% for conversion to a searchable PDF document. Notably; however, only about 1/3 of missed barcodes were those applied in-house. Barcodes that were not successfully found were typically rotated to angles between about 35° and 55°. We have not observed examples of a barcode being decoded incorrectly, nor would we expect to since barcodes generally contain internal checksums.

Though we do not currently employ two-dimensional barcodes for patient identification labels, a decision outside the authors’ control, we did want to assess the success of Reqscan on QR codes at various sizes and orientations. Using the Linux command-line utilities qrencode and the open-source utility LaTeX, QR codes were generated as 1,200 DPI png images and incorporated into a PDF document at a base size of 20 mm × 20 mm and rotated to angles of 0–340° at increments of 20° (for 18 equally sized barcodes at various rotations). Each barcode was encoded with the phrase “The quick brown fox at *x* degrees,” where *x* was the extent of rotation. Duplicate images were produced but scaled down in size to 60% (12 mm × 12 mm), 40% (8 mm × 8 mm) and 30% (6 mm × 6 mm) relative to their original size. This four-page document containing 72 barcodes and various size and rotation combinations was printed at 600 DPI and run through Reqscan in triplicate. Barcode identification success was 100% at 600 DPI scan resolution, 97.6% at 500 DPI, 94.9% at 400 DPI and 78.2% at 300 DPI. Not surprisingly, unrecognized barcodes were uniformly among the smallest (6 mm × 6 mm at 500 and 400 DPI scanning resolution and both 6 mm × 6 mm and 8 mm × 8 mm at 300 DPI scanning resolution).

DISCUSSION

Set Up and User Modifications

The setup of this software requires some knowledge of installing and updating a Linux system. It also requires the user to be able to use the Linux software repository of their Linux distribution (there are many), and use the appropriate command-line tools to install the necessary additional open-source tools ("dependencies") required. These tools are: Python 2.7, Tkinter, the Python-Zbar bindings, Python Imaging Library, scanimage, dmtx-utils, and ImageMagick. The SANE interface comes preinstalled with Ubuntu Linux.

It is natural for the user to want to customize the output to suit their needs. For example, they may wish to delete files of a certain name-format, shorten the names to the first *n* characters, suppress undesirable symbols arising in barcodes from outside sources, or concatenate the one-page PDF files of pages on which no barcode information was retrieved. These kinds of file manipulations are part of basic knowledge of Linux shell scripting and could be undertaken by any determined individual with introductory programming experience such as that afforded by an introductory undergraduate course in computer science.

Another natural file-manipulation a user might wish to undertake is to concatenate a barcoded page that is followed by a series of unbarcoded pages that all refer to the same patient. This would suit a situation where multipage requisitions are common. Of course, this approach assumes that pages remain in the order they were received. Although we considered doing this, we found that it too often led to the concatenation of documents from two separate patients. However, this procedure can be performed programmatically by sorting the current directory's file list by the PDF's timestamp (i.e. time of creation) and concatenating a page successfully named by its barcode content with the successive *n* files named "failed_1.pdf," "failed_2.pdf," "failed_n.pdf," until another successfully named file is found. Concatenation can be accomplished with the commands of the PDFtk tool.

The user could also use the shell scripts and PDFtk to concatenate all PDFs for which identical barcode content was found, as these are named according to the barcode information followed by an integer that is incremented as many times as necessary. This could likewise be easily achieved.

Finally, it would be a trivial task to make the day's PDFs write-protected by scheduling a so-called "cron-job" wherein the administrator ("root") could schedule a command to alter the write-privileges of new folders at a fixed time daily.

Because there are so many possibilities of customizations that could be undertaken with shell-scripts, we have not implemented them, realizing that sites should decide on their needs and undertake these simple shell scripting tasks themselves. However, if this seems daunting, the easiest way to make Reqscan function "out of the box" is to make sure a clear, near-horizontal barcode is applied to each page of all single-page and multi-page requisitions. If care is taken in application and scanning resolution is high enough, near 100% scanning success rate is expected, and the small number of failures could be manually renamed. It has been our observation that one-dimensional barcodes can be consistently found at 300 DPI scanning resolution but that two-dimensional barcodes require 600 DPI.

We elect to scan images at 300 DPI resolution rescale images to 25% of their original length and width which results in scanned pages which are between ≈ 50 and 150 kB and have dimensions of 636 and 825 pts and 75 DPI. This has not impeded our interpretation of a scanned requisition, but we could certainly re-evaluate this and store at a higher resolution if needed.

It is important to note that we use this software to take care of a clerical problem and not an accreditation-mandated document storage requirement. At present, Reqscan is not compliant with the provincial requirements for requisition storage. For this reason, the original requisition is still retained. The provincial requirement is TIFF only storage, 400 DPI resolution and storage on write-only media. Notwithstanding, for the simple purpose it was designed, we have been satisfied with Reqscan as the process of retrieval is essentially instantaneous.

In terms of "sunk-cost" in setup, development and implementation-related customization, Reqscan was written on a budget of $1500 by the first author. It was tested by a premedical student volunteer daily for 30 days and shell-script customizations requiring about 10 h of invested time were undertaken by the second author.

General Comments

Systematic review of the causes of laboratory errors demonstrates that most can be traced to the preanalytical phase[3,4] which includes errors in patient identification, the order-entry process, the specimen type, handling, processing, and storage. With respect to errors related to the interpretation of a paper requisition, in a study of 660 medical institutions in multiple countries, it was demonstrated that approximately 5% of outpatient laboratory requisitions were associated with at least one order-entry error and that 10% of institutions had errors with at least 18% of requisitions.[5] After those related to the physician name, not surprisingly, these errors were traced to the laboratory ordering a test that was not on the requisition or not ordering a test that *was*. Error rates for send-out tests are substantially lower (about 2%) but as high as 5% in some laboratories.[6]

While software of the type we provide here is not designed to prevent preanalytic errors *per se*, it does facilitate rapid requisition retrieval for confirmation and correction of any errors or omissions at the order-entry level in a *post-hoc* manner. One context in which the corresponding author has found Reqscan particularly useful relates to samples collected for tumor localization. For example, St. Paul's Hospital performs aldosterone analysis[7] for a province-wide Primary Aldosteronism screening program. Part of this service involves tumor localization procedures in which multiple specimens are collection from the left and right adrenal veins and the inferior vena cava. Errors in specimen identification could lead to the surgical removal of the incorrect adrenal gland. Because 12 specimens from three anatomical locations, collected in close temporal proximity arrive for accessioning and order-entry at our laboratory all at once, there is always a risk for specimen mix-up. It is now very easy to electronically retrieve the scanned requisition from radiology and confirm that labeling is correct. Additionally, those performing billing procedures for the laboratory have also found Reqscan useful for correcting or confirming billings to the insurer that have been questioned or rejected.

We recognize that there are advantages and disadvantages to the development of in-house software.[8] The ways that we have managed to keep costs down on this project were: (1) To limit the scope of the project to a specific task that would be done well (2) to avoid developing features that would require professional consultants (3) to build the project around the work-flow rather than the converse (3) and to undertake modest customizations using the programming expertise available on site. Keeping the project simple and dedicated as made Reqscan a good solution for the one problem it addresses.

The benefits of dedicated commercial scanning technology for rapid retrieval paper work associated with internal and referred-in anatomical pathology consult cases has been described, and although financial benefits were modest, uptake and utility was high.[9] The availability of scanned requisitions is considered a standard part of the laboratory automation solution[10] and in ideal circumstances would be integrated within or seamlessly link to the LIS.[11] From the perspective of avoidance of order-entry error, however, electronic physician order-entry has been found to be superior.[12-15] Unfortunately, this is not an option for us at present. Obviously, commercial solutions would offer the manner of convenience and refinement of LIS integration that is out of scope for this small-scale project. Nevertheless, user satisfaction has been observed to be high and document retrieval is essentially instantaneous. The weekday human time-commitment for scanning is approximately 45 min for the ≈ 700 requisitions scanned each day. Approximately half of this time is consumed with removing staples from requisitions sent from other hospitals and institutions. Since implementation, we have had no downtime due to scanner breakdown and maintenance of the scanner is simply cleaning of the reflective surfaces read by the scanner optics.

The rate at which ZBar successfully identifies the barcode is not 100% but in real-world environments, barcodes are often haphazardly applied, partially obscure or accidentally marred with marker or pen. In the absence of these problems, we have not observed situations where the barcode could not be identified. Two-dimensional QR barcodes, by nature of their built-in error correction, are much less vulnerable to information loss from distortion and marring and would be a good choice for an application like this where the documentation may have suffered some abuse before arrival.

There is no reason that Reqscan's use should be limited to paper requisitions. Any barcoded paper document could be stored for easy retrieval using Reqscan.

CONCLUSION

We present an open source solution for requisition scanning that finds the barcode(s) on a document and saves the document as a PDF in a folder specified by the date and named according to the barcodes identified.

ACKNOWLEDGMENTS

We would like to acknowledge Mr. Ken Liao and Mr. Calvin Lee for their enthusiasm for this project and assistance in its implementation.

REFERENCES

1. Brown J. Zbar bar code reader; 2012. Available from: http://www.zbar.sourceforge.net. Last accessed 2014 Nov 19.
2. Steward S. PDF toolkit; 2014. Available from: https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit.Last accessed 2014 Nov 19.
3. Bonini P, Plebani M, Ceriotti F, Rubboli F. Errors in laboratory medicine. Clin Chem 2002;48:691-8.
4. Plebani M. The detection and prevention of errors in laboratory medicine. Ann Clin Biochem 2010;47:101-10.
5. Valenstein P, Meier F. Outpatient order accuracy. A College of American Pathologists Q-Probes study of requisition order entry accuracy in 660 institutions. Arch Pathol Lab Med 1999;123:1145-50.
6. Valenstein PN, Walsh MK, Stankovic AK. Accuracy of send-out test ordering: A College of American Pathologists Q-Probes study of ordering accuracy in 97 clinical laboratories. Arch Pathol Lab Med 2008;132:206-10.
7. Van Der Gugten JG, Dubland J, Liu HF, Wang A, Joseph C, Holmes DT.

Determination of serum aldosterone by liquid chromatography and tandem mass spectrometry: A liquid-liquid extraction method for the ABSCIEX API-5000 mass spectrometry system. J Clin Pathol 2012;65:457-62.

8. Sinard JH, Gershkovich P. Custom software development for use in a clinical laboratory. J Pathol Inform 2012;3:44.
9. Schmidt RA, Simmons K, Grimm EE, Middlebrooks M, Changchien R. Integration of scanned document management with the anatomic pathology laboratory information system: Analysis of benefits. Am J Clin Pathol 2006;126:678-83.
10. Hawker CD. Laboratory automation: Total and subtotal. Clin Lab Med 2007;27:749-70, vi.
11. Sepulveda JL, Young DS. The ideal laboratory information system. Arch Pathol Lab Med 2013;137:1129-40.
12. Henricks WH, Duca DJ, Skilton BE, Ross GR. Electronically generated requisitions and error reduction in anatomic pathology. Arch Pathol Lab Med 2002;126:793.
13. Westbrook JI, Georgiou A, Dimos A, Germanos T. Computerised pathology test order entry reduces laboratory turnaround times and influences tests ordered by hospital clinicians: A controlled before and after study. J Clin Pathol 2006;59:533-6.
14. Pantanowitz L, Henricks WH, Beckwith BA. Medical laboratory informatics. Clin Lab Med 2007;27:823-43, vii.
15. Baron JM, Dighe AS. Computerized provider order entry in the clinical laboratory. J Pathol Inform 2011;2:35.