9-5-1995

Guinea Pig with a PC: OR, BCL3 Gap Reports In ASCII and What They Can Mean to You

S. Ray Granade
Ouachita Baptist University, granade@obu.edu

GUINEA PIG WITH A PC:
OR, BCL3 GAP REPORTS IN ASCII AND WHAT THEY CAN MEAN TO YOU

by: S. Ray Granade

9/5/1995

Everything that follows must be understood in light of three major facets of my experience, training, and assumptions. Without this background, at least some of this article will be less (if not in-) comprehensible, so bear with me.

First, I'm trained as an historian, have taught history for 25 years (still do occasionally), research some, and publish when I can. I try to keep my hand in as an academician, for I identify myself as a member of the academy. My formal training ended as historians were beginning to use machines to engage in cliometrics—the statistical study of history—and none of my professors were so inclined. The closest to that designation was a British historian who used Hollerith cards in taking notes and sorted them with a card-sorter. My long-term interest in collective biography has fueled interest in databases and machine-readable files and their potential for and use in research. All of this pre-dates my conversion of over a decade ago from classroom teacher to library Director and subsequent training in librarianship.

Second, temperamentally I am a "`tekkie' wannabe" who first started fooling with computers in 1981. Early and continuing interest in things mechanical, an original determination to be an engineer, and an eagerness to tinker have all meant that equipment has an attraction for me in which awe holds no place. I learned BASIC and the rudiments of computer operations early, then got a computer and have played with it and its successors as research/writing/productivity tools ever since. Our staff is small enough that when we automated we added no systems person, which I by default became.

Third, I migrated from a manual typewriter to word processing, starting with **Wordstar** and progressing to a mainframe monster called **Select** and finally an academician's dream**—Nota Bene**, a multi-lingual textbase database, word processor, and bibliographic utility combined. It will do everything but serve breakfast and I am its devoted slave.

The project to produce a bibliographic database I could use on my PC had its roots in several frustrations. First, we have a small staff: one person each in AV, Reference/Circulation, Periodicals, Government Documents, and Special Collections; two in Technical Services; and the Director and a secretary. The Director has maintained collection development responsibility from the time the staff totaled one. Second, in any given year we receive a large number of gift books which need to be evaluated. In 1993 we received two gift collections of about 4,000 titles each and two smaller ones of about 2,000 titles each, in addition to our regular flow. Third, while the library is the University's and must reflect its curriculum, a professional staff cannot abrogate all responsibility for collection development. Some departments take their collection development responsibilities very seriously, others do not; faculty negligence cannot be cited in defense of a collection which ignores any discipline's needs. Fourth, Books for College Libraries' core collection listing should provide a wonderful bibliographic "floor" upon which departmental offerings stand. In practice, I find its current (paper) format difficult to use as a systematic collection development tool, especially when faced with large numbers and little time.

Believing that standards are useful both in a utilitarian sense and as propaganda with administrations, I tried for a decade to match our holdings with BCL1-3. Student workers checked individual volumes against first our card catalog, then the OPAC that replaced it. Time consumed and what I considered an unreasonable error rate obviated that as a viable approach without negating the potential of using BCL systematically as a collection development tool. Work with a faculty committee and our administration made this approach part of our "Centennial Goals" in 1986, which pledged us to have that foundation by the year 2000.

The project also had its roots in the library's gradual automation. Ouachita was an AMIGOS charter member in 1976. Audio-Visuals automated in 1981 with a PC and an in-house program. A 1984 grant provided an IBM-AT and printer for the office. In 1987, Ouachita and its sister state-supported institution, Henderson State University, installed an integrated library system under the auspices of the Joint Educational Consortium. Ouachita began electronic ordering via modem connection the next year. Successful use, added to increased efficiency and quick feedback, convinced us that electronic ordering was the only viable approach, especially on large orders.

My interest in the solid foundation in book holdings offered by BLC and in automation converged in several sources' announcement offering tape matches of BCL3 with an individual library's holdings. This seemed the answer to my dilemma. A machine could supply a listing of the BCL items our library lacked. Some vendors offered the tape match for a fee; others offered it free in return for a specified level of orders. Now money was the issue. An unexpected grant and a portion of annual endowment income which we set aside for BCL purchases coincided to make the project feasible.

The change from interest to reality meant further thought, especially on output—the format in which I would get the results. I wanted something I could keep on a hard drive as a database to check when catalogs/flyers/lists came in, then port over to make electronic ordering easy.

Conversations with AMIGOS prompted me to ask whether output might be available on diskette in ASCII format. No one had asked that question before. Eventually AMIGOS assured me that the output could be produced in ASCII and on high-density 3.5" diskettes. We agreed on a cutoff date and the next month, AMIGOS sent me a box of 44 diskettes, each except the last virtually filled with an average of about 540 "paper pages" or 600 "screen pages." The match produced a file of 41,904 records: 23,688 pages (26,676 screen pages), as opposed to the 2,336 pages in the original!

One issue I should have considered was the method of the match. I knew that there were multiple records for the same basic item in the database, and that each had its own OCLC number. It did not occur to me that the match might be made by OCLC number, which would automatically exclude a match of the same basic item with a different record. I didn't even think to ask if a different kind of match would be possible—preferably by author and title.

The result of my lack of foresight on this issue meant that the output would include items which we held, but held in a different edition. The listing would therefore be much longer than necessary and the database would contain extraneous items. The result was not an unmitigated disaster; given the use to which the database would be put, that problem was better than the one of not having all the items we lacked (i.e., less would definitely NOT be more!).

A second unresolved issue was exactly what the final result should look like. Even had I known, I'm not sure I could have either gotten or paid for it; subsequent discussions indicated that AMIGOS could not produce it. The result of not resolving this second issue appeared as soon as I loaded the information from the first diskette into memory and took a good look.

For my purposes, the records were initially incredibly cluttered. The output in ASCII was merely paper output in another format. Each "page" had a header—the AMIGOS screamer with date, time, page number, report number and title—and spacing and hard return for the "page break" as a footer. Page breaks rarely lined up with records, so records were divided between pages. In addition, the record was full MARC, including the fixed fields.

Working through the first several pages produced an initial answer to the format issue, a decision based on a number of considerations. First, good identification demanded OCLC number and ISBN where possible, author, title, publication information (including edition), and number of volumes. Second, the ability to check earlier editions or printings required titles of previous editions. Third, knowledge of whether the item had been included in previous BCL editions could prove helpful. Finally, the file's size meant that space would be at a premium and required a fine balance between space and information (though initially I erred on the side of information).

The third issue now became how to get from what I had to what I wanted.

The first diskette was my test case. I knew I had to delete extraneous material, but I had to find the final format I wanted and how to change what I had and get what I wanted with the tools at hand (i.e., **Nota Bene**).

What emerged was a bibliographic unit divided from the next by two carriage returns. For databases, that's perfect; for space conservation in a file to be searched by a program with a good search capability, 1 separator will do. Wanting the file to look familiar meant author, title, and bibliographic information (the 100, 245, and 260 fields) snugged against the left-hand margin but with the lesser information set off somewhat from the rest by indention. The OCLC number and ISBN came from the 001 and 020 fields, again indented to allow quick observation of author and title. The 500 and 504 fields offered any earlier edition that had been reprinted but not revised, and the 300 signaled multiple volumes. I kept 590 field just out of curiosity for what'd shown up in BCL2. Trying to keep each portion of the entry to a single line (both for ease of access and space considerations) meant keeping translators but not the original language, the editor but not annotator or author of introductions and/or prefaces. The edition went from its own line to join the bibliographic information from the 260 field, just as did the number of volumes (if multiple) from the 300. Everything else was expendable.

Getting the end result I wanted meant deleting much of the file. Workspace was the first consideration. MS-DOS only gives you 640K of memory with which to work, and from that must come the space to run whatever application you use. Since the files were about 1.4 meg and since about 340K goes to run NB, that meant I had less than a quarter of the space I needed. Like most good word processors, NB has a memory manager that utilizes an overflow file system, shifting chunks of large files between memory and the overflow drive as it stores currently unneeded chunks. Whatever has been designated as the overflow drive gets the excess. Initially my default overflow drive was the hard drive (C:). The first stiffy I loaded onto my hard drive and worked from that, but access time was horrendous (though not as bad as working from the floppy drive would have been) and the hard drive kept whirring. My machine has 4 meg of memory and NB is able to handle a ramdrive for overflow. After the first file, I set up a 2-meg ramdrive as the default overflow. That speeded up access immensely.

The next consideration was how to delete all the stuff I didn't want. The first file convinced me that there was a quicker way than going line by line, though doing so had certain advantages as I got familiar with both the file and the process. I discovered discrepancies, for example—items on the first diskette that I KNEW we had because I'd used them here. On the first run-through I checked my memory against the catalog and discovered that we indeed had items that the file claimed we didn't. More on that later.

Global changes and wildcards were the answer to the speedy-deletion problem, but only with certain hard-earned and well-learned caveats.

Most word processors offer global changes in one of two modes—item-by-item or invisible. The former is time-consuming, the latter dangerous. The computer does EXACTLY what you tell it. Global invisible changes may do things you don't want done if you're not careful, especially depending on whether the search engine is case-sensitive or case-insensitive (Press also selects "press" wherever it finds those letters, so that impressionist became imPionist in case-insensitive).

Wildcards come in alpha, numeric, separator (punctuation), and character (anything that takes up a space—a true wildcard). Character wildcards are best for removing chunks of text, but their peculiarities have to be taken into account. Since the carriage returns that mark the end of each line take up a space, a character wildcard will remove them, playing havoc with the file.

The general rules I developed with the global-wildcard deletions are pretty simple, though not entirely foolproof. The first was a systematic procedure designed to retain the information I wanted intact, and each information line inviolate. The first step was to delete all headers/footers first, for a file-size reduction of about .5 meg. The second was to work through the file from top to bottom of each entry sequentially. The third was to delete lines incrementally from left to right. The fourth was to delete whole lines before pieces of lines. The fifth was to replace globally certain repetitious words with abbreviations and globally delete certain unwanted phrases. All "New York"s became "NY"s and "with an introduction"s were deleted, for example. The last step was to make global changes to set up the format I wanted—deleting field markers (though I later started keeping them as convenient ways to delete partial lines, especially in the 600 and 700 fields) and putting in tabs on certain lines.

The second general rule was to ensure redundancy. That meant a series of back-ups at specified points so that I never lost much, or so that I could easily undo mistakes. First, I loaded each diskette onto the hard drive and worked from that, keeping my original intact. Then, I backed up after the header/footer/"page break" removal and maintained that as a separate file so that I could always start from that point if I needed to rather than starting over with the original. These two versions gave me the necessary means to check up on oddities, like missing data in the 020 (ISBN) or 001 (OCLC#) fields (which sometimes showed up incomplete).

The next step in ensuring redundancy was to back up my working file after each major deletion. I eschewed automatic backup to avoid the possibility of compounding an error in deletions by over-writing a good version with one with deletions that corrupted the file. Finally, as I worked my way through the file, I backed up after about every five pages.

Having a global/wildcard strategy of deleting unwanted information from the file did not obviate the necessity of working through each one if I wanted to maintain minimum file size. Several hours of global deletes produces a file of about .5 meg. A complete walk-through brings it down to about .14 meg in about a day's time.

All of this, though it can be done with a word processor, is time-consuming. After getting the process down pretty well I could do a diskette in about 6-8 hours—too long, considering the number of diskettes. At that point I called on the expertise of my son, Stephen, who wrote a program in C to process the file and write a new one with only the fields specified, in the format specified. In roughly ten minutes, my 386-33 can take the original from the stiffy and produce a .22 meg file on the hard drive. Just out of curiosity I tried it with a complete file on the hard drive, so the program was reading from and writing to the hard drive. The process took less than five minutes. Global changes/replacements of the unwanted words/abbreviations brought the file size down below .2 meg in about another 15 minutes. I figured I could handle the difference in file size for the time savings.

As with most undertakings, the proof of the pudding is in the eating. I've used the resulting match file twice so far, both times with out-of-print catalogs. Both were, in my estimation, highly successful and indicated that I'd made a wise investment. Let me use the first as an example.

In late January we got a catalog from Q.M. Dabney for out-of-print music books. In about two hours I had run through the catalog, checking author and title. Sixty-three showed up. Those 63 were then checked against our holdings; 35 emerged as "no shows." We ordered them by telephone the same day and secured 23. Had we followed our normal pattern, we would have spent weeks checking various places and been lucky to get any. My observation has been that out-of-print catalog items vanish pretty quickly. Actually, truthfulness demands the acknowledgement that we'd have probably tossed the catalog after a month or so because we hadn't had time to get to it. [Subsequent experiences—summer '94—Austin Bookshop catalog—15 showed up, we had 4 and got all but 3 (8). Zobel catalog—539 books bcl3; 208 showed up in LINUS; 65 were in gap report.]

One problem with the match, and one reason I found Stephen's program a good compromise between speed and file size, is revealed in the first two figures in this search. While 35 beats the 28 erroneous reports, a 45% error rate isn't great. Is it worth it? In my estimation it is because of the ability to do quick searches and turn out-of-print catalogs around in time to get more of the books we need at reasonable prices and because of the ability after further checking to port over chunks of the report into electronic ordering, or print up a request to send in paper format without the necessity of keying the information into a file the first time.

My experience has generated recommendations for both AMIGOS and potential users of Gap reports in ASCII format. First, Helen and I talked about providing matches specified by author and title (preferably) or at least by title. Her response is that such a capability does not seem to be in the immediate future. I would recommend to potential users that they seek out someone who could provide such a match at a competitive price. If a library has a local system which supports tape creation of MARC records in the necessary format, an offer like that from Midwest might be very attractive should they make the output available in ASCII on diskette.

Second, I recommended that AMIGOS allow the client to specify what fields should be deleted from the file, and the format of the output. Helen responded that at this time AMIGOS can and will suppress headers and footers; the fields issue depends on the flexibility of their programming and that is currently being explored. Suppressing headers and footers will reduce file space requirements significantly. Since the ASCII output required a small per-diskette cost, some savings would result from header/footer suppression. With a program like Stephen's, this second recommendation is less significant in terms of processing time and space considerations than would otherwise be true. Even with that, there'll always be a need to "clean up" personally for a perfect file; the machine can't do it all.

Global changes:
University Press-UP; University-U; Press-P; New York-NY; N.Y.-NY; N. Y.-NY; N.J.-NJ; N. J.-NJ;

Global deletes:
All delta-a-x; ill.; illus.; port.; cm. Changes through experience—don't change deltax; use it as separator to delete in 600s; wildcard for all 600s & 700—6X0 & 7X0 Also global authors, esp in lit.

Decide what's worth a global ci & what isn't.

Subsequent use & refinement.

; abbreviations were therefore fine, even mandatory
I had a reasonably good idea of what I wanted, but not exactly how to get there.

PROCESS FLOW CHARTS

1. Out-of-print book catalogs:

    A. Check item against BCL3 database.

    B. Item there?
        i. If yes, mark.
        ii. If no, ignore and go on to next item.

    C. Check marked item against LINUS.

    D. Item there?
        i. If yes, note call number, delete from database.
        ii. If no, consider for order.

2. Gift items (single items or collection):

- A. Check item against BCL3 database.

- B. Item there?
  - i. If yes, check LINUS.
  - ii. Item there?
    - a. If yes, compare edition, condition.
      - (1) Item better than current holding?
        - (A) If yes, replace.
        - (B) If no, discard.
      - (2) Delete from database.
    - b. If no
      - (1) Send to technical services.
      - (2) Delete from database.
  - iii. If no, check LINUS.
  - iv. Item there?
    - a. If yes, compare edition, condition.
      - (1) Item better than current holding?
        - (A) If yes, replace.
        - (B) If no, discard.
    - b. if no, consider for addition to collection.

3. Sale catalogs—pre-1986 items only

- A. Check item against BCL3 database.

- B. Item there?
  - i. If yes, mark.
  - ii. If no, ignore and go on to next item.

- C. Check marked item against LINUS.

- D. Item there?
  - i. If yes, note call number, delete from database.
  - ii. If no, consider for order.