

St. Catherine University

From the Selected Works of David Lesniaski

1993

Expanding Passport's Horizons with Third-Party Software: Newkey

David Lesniaski, *St. Catherine University*



Available at: <https://works.bepress.com/david-lesniaski/25/>

There are a number of good sources for dating recordings. One of the best sources is the Schwann Catalog or Schwann-1. The process is cumbersome if not tiring and requires moving backwards until a recording no longer appears. The issue that a recording first appeared is generally a good estimate of its date of release or publication. Bielefelder and Gramophon can be used in a similar fashion. Phonolog can also be helpful, but since it is designed as a looseleaf publication, only those libraries that keep the back files would have access to the needed information. Performers listed on a recording may be useful in dating a recording since there are sources that document personnel; this is useful for jazz.

The last 10 years have seen the publication of a number of very good discographies which chronicle the publishing history of some of the major recording companies. Also the notes issued with or on the recording are also useful, and often chronicle the performance recorded.

There are also problems associated with dating recordings. They include record company takeovers, releases, and re-masterings (analog to digital).

One of the most frequently asked questions in sound recording cataloging is "what do all of the dates mean?" The "P" date is the copyright date of recorded sound; the date usually is the date of first release and is useful in the absence of a formally stated date of publication. The "C" date indicates copyright protection of the work performed and/or the accompanying textual material (libretto, song texts, and program notes). A single "P" date can be taken as the date of copyright for the recording. When multiple "P" dates appear on a single recording, it is up to the cataloger to determine a) whether they indicate a reissue (as would be the case when a recording has only one work), in which case the latest date should be transcribed, or b) whether each date represents a copyright for a different part of the recording, and no date should be transcribed, because no date applies to the entire recording. An estimated date can then be determined (based on the latest date on the item). When no "P" date is present, a "C" date before 1971 should be transcribed as the date of copyright. For a "C" date later than 1970, the cataloger should infer an estimated date of publication.

Bibliography

Deutsch, Otto Erich. *Music Publisher's Numbers: A Selection of 40 Dates Lists 1710-1900*. London: Aslib, 1946.

Deutsch, Otto Erich. *Musikverlags Nummern: eine Auswahl von 40 datierten Listen, 1710-1900*. Berlin: Verlag Merseburger, 1961.

Hopkinson, Cecil. *A Dictionary of Parisian Music Publishers, 1700-1950*. London: C. Hopkinson, 1954.

Humphries, Charles and William C. Smith. *Music Publishing in the British Isles from the Beginning until the Middle of the 19th Century; a Dictionary of Engravers, Printers, Publishers, and Music Sellers*. 2nd ed. with suppl. New York: Barnes and Noble, 1970.

Krummel, Donald W. *Guide for Dating Early Published Music: a Manual of Bibliographic Practices*. Hackensack, NJ: J. Boonin, 1974.

Music Printing and Publishing. Edited by D. W. Krummel and Stanley Sadie. New York: W. W. Norton, 1990.

Neighbour, O. W. and Alan Tyson. *English Music Publishers' Plate Numbers in the First Half of the Nineteenth Century*. London: Faber and Faber, 1965.

Phonolog Reports. Los Angeles: Phonolog Publishing Division, 1948- (Earlier title: *Phonolog Reporter*)

Schwann-1 Record and Tape Guide. Boston: Schwann Record Catalogs, 1949-

Sonneck, Oscar. *Bibliography of Early Secular American Music, 18th Century*. Revised by William Treat Upton. New York: Da Capo Press, 1968.

Joan Swanekamp
Columbia University

EXPANDING PASSPORT'S HORIZONS WITH THIRD-PARTY SOFTWARE: NEWKEY

Introduction

It is possible to use third-party software (that is, non-OCCLC software) with Passport in order to make certain cataloging or other procedures more efficient. A set of programs known variously as TSR programs (terminate-and-stay-resident programs), macro programs, or "pop-up" programs is especially useful. Newkey, one such program, is one which, when activated, is resident in memory. It can record practically any number of

keystrokes and assign them to one key or a key sequence. This macro feature is similar to the "function key editor" in Passport. Newkey, however, allows macros to be assigned to any key or key sequence, not just to function keys. For example, it is possible to type *Bach, Johann Sebastian, |d 1685-1750* and assign that to the key sequence <jsb>. Since Newkey records keystrokes, it is possible to record a series of functions (like the <Esc I> command for the alternate character set) as well as text. Newkey also has a "cut-and-paste" feature similar to the Passport keys F6, F7, and F8. It is possible within Newkey to mark and copy text directly from the screen to a macro key or key combination. With Passport, only one copy/paste operation can be done at a time; it is not possible to store more than one segment of text. With Newkey, it is possible to store practically an unlimited number of "cuts" to practically any set of keys or key sequences.

Newkey also has features not found in Passport. It is possible to create different types of macros in addition to a simple recording of keystrokes or screen cuts. It is possible, for example, to create "fill-in-the-blank" macros, where a standard text will play back, pause to enable variable text to be entered, then continue playing back, completing the line of standard text. One example of this is the note

500 Program notes ... in container.

The text *500 Program notes* can be programmed as a macro, which will play back, pause for the variable text such as *by Sara Smith (20 p.)* to be entered, then continue with the *in container*.

These features of Newkey can significantly increase the efficiency of creating or editing bibliographic records in OCLC (or in local systems). For example, it is possible to do all the authority work at once (without toggling back and forth between the authority file and the bibliographic record) by pasting authority records into a series of macros, and then pasting those macros into the bibliographic record all at once. This can be an extremely valuable tool for use with the Microenhancer: it is possible to copy all authority and bibliographic data needed during one online session, save this information in Newkey, exit from OCLC, start the Microenhancer, and paste the information into the Microenhancer records from Newkey. It is possible to create macros for information which is repeated in different places in the bibliographic record (such as titles in the 505 and 740, for example). Many keystrokes can be saved by creating macros for commonly used field tags (such as *500, 700 12*, etc.) or subfields (like *|h sound recording* or

|o arr.). Using macros instead of keying in information reduces the number of keystrokes needed to create or modify a bibliographic record and also reduces the chances for typographical errors. Macros can be saved to a file, edited, and moved from one machine or application to another.

Newkey also can be used with other programs--from local bibliographic editors to WordPerfect to electronic mail--so it is not necessary to learn different programs or commands for each application.

This article provides a detailed description of the uses of Newkey for cataloging on OCLC. It includes many examples and techniques, and includes information on installing and configuring the program. Feel free to skim the article to get a feel for what this program can do. If you acquire Newkey, this article should also help you install, configure, and use Newkey with a minimum of trial and error. Though this article is very detailed, and contains some technical information, Newkey is not a complicated program to install or to use. It is no more difficult to use than Passport, and anyone with basic computer skills should be able to use Newkey with a minimum of effort.

Programs similar to Newkey (such as ProKey or Keyworks) have slightly different features but work in much the same way. Newkey, as of this writing, is the least expensive of such programs.

In the following article, examples are shown in italics. Specific keystrokes or keystroke sequences are enclosed in angle brackets < >. For example, the shift key is notated <Shift>, the function key F10 is <F10>, the keystroke sequence Control-6 is <Ctrl 6>, and so on.

Passport

OCLC is to be commended for making substantial improvements in system and software functionality. Three improvements in particular have helped make editing OCLC bibliographic records much easier: the improved function key editor, the cut/copy/paste feature, and the constant data feature. The ability to easily assign practically any series of keystrokes to function keys is very helpful; it in effect allows the substitution of a few keystrokes for a longer, more complicated, and more error-prone sequence. The ability to cut and paste enables the rapid and error-free transfer of information from one record to another, or from the authority file to a bibliographic record. The constant data feature also can save time and keystrokes by superimposing a

template of features common to a group of records over a workform or bibliographic record.

Unfortunately, each of these features also has some frustrating limitations. Although the function key editor allows the assignment of keystrokes to any function key, most of the preprogrammed functions ("reformat", "page up", and so on) shouldn't be overwritten. This leaves only 22 free keys available (plus the <+> and <Enter> keys on the numeric keypad). This may seem like enough...unless you've been spoiled, as I have been, with the option of having many more keys available. Also, the function keys are not mnemonic. If you were cataloging a lot of Bach, and wanted a macro for *Bach, Johann Sebastian*, |d 1685-1750, <jsb> is a lot easier to remember than, say, <Alt F10>. While it is possible to have multiple function key files, I feel it is inconvenient to toggle back and forth between several different function key files, especially since Newkey allows an almost unlimited number of macros.

Similarly, the cut/copy/paste feature, though useful, can handle only one block of text at a time. If you were doing analytics for a typical sound recording, you easily could be looking up a dozen of name or name-title added entries. If you wanted to cut and paste each one from the authority file to the bibliographic record, it would be necessary to toggle between the authority file and the bibliographic record for each authority record.

While the constant data feature also is very useful, it is necessary to create different constant data files for different workform formats, or for different media (compact discs vs. cassettes, for example) within the same workform format. The constant data record can't be modified to fit the peculiarities of each bibliographic record, so often it is the case that the constant data record has too much or too little information for the bibliographic record in question. Also, it is not possible to set up a template with just field tags, such as 028 01, 500, 505 0, and 700 12 since each field requires that bibliographic data be present.

Despite these limitations OCLC should be congratulated for improving Passport. However, I have found using Newkey significantly increases the efficiency of the cataloging process, and has benefits for other OCLC and non-OCLC applications.

Macro, TSR, and "Pop-up" Programs. Newkey

Newkey, like other terminate-and-stay-resident, macro, or "pop-up" programs, is resident in RAM and can be called up through just a few keystrokes. When it is active, it intercepts all keystrokes or input before the input reaches the main program which is running. Keystrokes which "mean something" to Newkey are "translated" into their macros; keystrokes which don't mean anything to it are sent on directly to the underlying application program. For example, if you "told" Newkey that the keystroke sequence *jsb* "meant" *Bach, Johann Sebastian*, |d 1685-1750, anytime you typed *jsb* the program would look in its dictionary, substitute *Bach ...* for *jsb*, and send *Bach...* on to the application program you had running (like Passport). Passport is unaware all this is going on; for all it knows, you typed *Bach...* rather than *jsb*. If you typed *jsv* instead, Newkey would realize that keystroke sequence had not been programmed to mean anything, and *jsv* would be sent on to Passport. When Newkey is not needed, a few keystrokes will turn it off, but it's still resident in memory, and can again be activated with a few keystrokes as before.

A "macro" is the substitution of a few keystrokes for a longer, more complicated sequence of keystrokes. If you use WordPerfect, you might have used the macro subroutine built into the program--for example, you could have your name and address programmed as a macro for letter writing. Many programs, especially sophisticated ones, have macro subroutines. Passport's function key editor is, in essence, a macro subroutine. It lets you assign a command (like <sl6>) or a sequence of commands (<Home><sl6><F11>) or practically any series of keystrokes to one of the function keys (F1-F12, or F1-F12 combined with the Shift, Control, and Alt keys). The way to do this in Passport is to invoke the function key editor (<Alt f>), choose the key to assign the macro to (like <Alt F1>), type the keystrokes you want assigned to the macro key, save the macro, and exit the macro subroutine. From that point forward, pressing <Alt F1> invokes the macro--that is, the keystrokes assigned to <Alt F1>. Notice that this replaces the deservedly maligned "function key editor" in the earlier versions of Passport, and brings this feature of Passport much closer to that of many other programs.

Newkey is compatible with both Passport and the Microenhancer (as well as with many other programs). Though I will be discussing features of Newkey specifically, my remarks in general apply to other similar programs (see Appendix 3 for a listing of these).

Advantages to Using Newkey and Similar Programs with Passport

Newkey has several features not found in Passport. In particular:

- 1) it is possible to assign a macro to any almost key or key combination--for example, `<Alt 5>`, `<Ctrl 9>`, `<Ctrl x>`, `<Alt F5>`;
- 2) it is possible to assign a macro to a mnemonic series of keystrokes, such as `<jsb>`; therefore, it is possible to create almost an unlimited number of permanent or temporary macros;
- 3) it is possible to copy practically anything displayed on the screen to a key or key combination, thus creating a macro by copying the information on the screen rather than keying it in. This means that it is possible to make almost an unlimited number of separate "cuts" from one file or source (like the authority file), save them, and then paste them in to another file (like a bibliographic record) at your convenience.
- 4) any macro--including macros created by copying text--can be saved for future use.

"Almost unlimited" suggests there are limitations: each macro is resident in memory, so the more macros, the more memory is used to store them. Also, after a while, it might be hard to think of "mnemonic" keystroke sequences. However, it is possible to create several different macro files. Though the program runs only one file at a time, it is very easy to toggle between one file and another. One particular advantage to this is that several people can use the same machine and program, yet everyone can have his or her own macro files. Files also can be copied to disk and used on different machines (assuming the program has been installed on each computer).

One other very nice feature of Newkey is that it can be used with programs other than Passport. The macros created for cataloging in Passport can be used in WordPerfect or other word processing or database programs. It is possible to have several types of macros for different programs or functions in one file, and keep Newkey and that file open when switching from one program to another. For example, I have one Newkey file which I use for cataloging, for word processing, and for e-mail. This file also contains macros which actually are batch files which enable me to switch quickly from one program to another. A word of caution: each application program used with Newkey interprets keystrokes as if they were created in that application program. If a Passport macro contains the keystrokes for

a subfield delimiter (`<Ctrl d>`), playing back that macro in WordPerfect will not display `<Ctrl d>` as a subfield delimiter but will display `^D` instead--just as typing `<Ctrl d>` in WordPerfect would.

Before I show examples of creating and using macros in Newkey with Passport, I thought I'd discuss why anyone would go to the trouble of learning and using such a program at all. You've probably had the experience of trying out a program which would simplify some task or which was an enhanced version of a previous program, and found it which took longer to learn than the time it saved. There are some tricks to using Newkey, but it has a self-guided tutorial which is very good. I have included many suggestions in this article which I hope will save some trial and error in configuring and using the program. Probably a few hours of experimenting will get you comfortable with most parts of the program you will want to use. In doing original cataloging I have found that I've reduced the number of keystrokes needed to create a record by at least 25%, and the ability to do all my authority work at once and paste in a series of authority records without toggling back and forth between the authority file and the bibliographic record makes that part of the process much faster and more accurate than using Passport alone. While using Newkey may not result in a great saving of time, it lessens the tedium of rekeying the same information time and time again, and therefore makes it easier to focus on the content of the bibliographic record rather than the mechanics of its creation.

There are several specific advantages to using Newkey:

1) **Saving keystrokes.** Original cataloging, for example, requires entering many of the same fields and information in each record, such as the 007 field, the compact disc note, or the keystroke sequence `700<space>12<space><space>` for each composer-title added entry. Programming these as 1- or 2-keystroke macros saves a lot of typing.

2) **Increasing accuracy.** Since it's very easy to copy information from one place to another, it often is much quicker to copy and paste than it is to key or rekey names, titles, and the like. While it may seem silly to cut/paste a short or simple name, cutting and pasting guarantees there will be no keystroke errors. We all know how easy it is to type something incorrectly, and how difficult it is to find those "simple" mistakes.

3) **Saving time.** Since it is very easy to do all the authority work at once, without toggling back to the bibliographic record (by searching and saving all names/titles from the authority file in macros, then

toggling to the bibliographic record and pasting in all the authority work), this saves the time involved in going back and forth from one file to the other for each name, title, or subject entry. Also, using 1- or 2-keystroke macros in place of longer pieces of information (like the 007 field, certain 500 fields, stock field tags and indicators, or the repetition of names in the 505) is much faster than typing all this information in. Increasing accuracy through the copy/paste facility of Newkey also means that less time is spent on correcting errors before or after the record is input.

Newkey supports the creation of several different types of macros: "simple" macros, macros created by copying information off the screen, "fill-in-the-blank" macros (which can be of both the fixed-length and variable-length variety), and combinations of the above.

Examples

1. Simple macros.

By "simple" I mean the sort of macro created by just typing and saving those keystrokes to a key or key sequence, like

```
500<space><space><space><space><space>
Compact disc.
711 12<space><space>
<Ctrl d>x<space>
```

These macros are very easy to create. For example, this is how to create the *Compact disc* note: Assuming Newkey is "on",

- invoke Newkey's macro function by pressing `<Alt =>`
- choose the macro's playback key--say, `<Alt d>`
- type the macro (above)
- press `<Alt ->` to stop recording the macro (The macro is saved until the `<Alt d>` combination is used for another macro, until another Newkey macro file is loaded, or until the computer is turned off or rebooted. See below for techniques on saving macros permanently.)

Then, anytime `<Alt d>` is pressed the note

```
500 Compact disc.
```

appears. The note requires 21 keystrokes; `<Alt d>` requires 2.

Just to show that practically any Passport function can be part of a Newkey macro, here's one for generating the labelling command *sl6* from anywhere in the record:

- invoke macro editor as described above
- choose key `<Ctrl F6>`
- type `<Home>sl6<F11>`

Anytime `<Ctrl F6>` is pressed the *sl6* label screen will appear.

Note that Passport uses `<Alt>` in combination with other keys for various Passport and OCLC functions. If Newkey is running, and if macros are programmed for an `<Alt>` combination (like the `<Alt d>` example above) this will override the Passport `<Alt>` key combinations. That's how the `<Alt d>` sequence above creates the "Compact disc" note rather than exiting to DOS, which is what `<Alt d>` does in Passport. However, it is easy to access the Passport keys--just turn Newkey off! The default keystrokes for turning Newkey off are `<Ctrl 6>`. Pressing this key combination will stop Newkey (but will not erase the macros you've created) and let you use the Passport key combinations. Pressing `<Ctrl 6>` when you're done with Passport will turn Newkey on again. Since it's easy to forget when Newkey is on and when it's off, it's safest not to program key combinations which could be troublesome in Passport when you don't want them--like `<Alt o>` (logoff).

2. Copy/paste macros.

In Passport, the copy/paste function is straightforward: position your cursor, press `<F8>`, outline the text, press `<F8>`, press `<shift F6>`, move to where you want to insert the text, press `<F7>`. The text you copy is assigned automatically to the `<F7>` key. You have no say about which key the information is copied to, and you cannot copy more than one piece of information at a time. With Newkey, the procedure is similar, but you can specify the key(s) you want the information copied to, and, since practically any key or key sequence can be used, you can copy many pieces of information before pasting them. The copy/paste function in Newkey is like the macro function; you are creating macros by copying information off the screen rather than typing information into a macro. The steps are straightforward:

- position the cursor (which can be done anytime before the text to be copied is marked)
- press `<Alt />`
- choose *U* from the menu. Text is copied to the `<Ctrl ins>` key sequence unless you specify another key. If you want another key,
- hit *F1*
- choose the key; if you want a series of keystrokes (like *jsb*) to serve as the macro, hit `<Return>`, type

in the name of the macro (in this case, *jsb*), then hit **<Return>**

- hit **<Home>** to begin outlining the text
- move the cursor; hit **<End>** to complete the marking.

Many of these steps can be combined into a macro within Newkey. For example, I have assigned the **<Alt />**, **U**, and **F1** commands to one macro.

3. "Fill-in-the-blank" macros: variable length

Sometimes there is "stock" text surrounding variable text. For example, in the note

500 Program notes <by xxx ([n] p. : ports.)> in container.

everything between **< >** will vary from item to item, but the remainder of the note usually is the same.

Newkey will allow you to create a "fill-in-the-blank" macro where intervening information can vary, but the information before and after is constant. To do this:

- **<Alt ==>**
- choose the macro key--say **<Alt p>**
- type 500<5spaces>Program notes<space>
- **<Ctrl />** (this starts the "fill-in-the-blank" function)
- **<Ctrl />** (this ends the "fill-in-the-blank" function)
- type in container.
- **<Alt ->** to end the macro.

When you play back this macro, it pauses after

500 Program notes

and lets you fill in the remaining information. When you are done, and are ready to complete the note, just hit **<Enter>** and the remainder of the note will appear on the screen.

4. "Fill-in-the-blank" macros: fixed length

Typing 700<space>12<space><space> (7 keystrokes) may not seem a big deal, but why type 7 keystrokes if you need only 2 or 3? You can create a macro that will enter

700<space>1

then pause for you to fill in the second indicator, and, after you enter the second indicator, will move to the beginning of the field text. Newkey lets you create a macro with constant data preceding and following a fixed number of characters--in this case, with only one space allotted for such a character. To create this macro,

- **<Alt ==>**
- choose the macro key--say **<Ctrl 7>**
- type 700<space>1
- **<Ctrl />** (this invokes the "fixed field" function)

- **<space>** (this is the one space you will want to fill in)
- **<Ctrl />** (this ends the "fixed field" function)
- **<space>** (this will move the cursor to the beginning of the field)
- **<Alt ->** (to end the macro)

5. Combination macros

Typing timings in the 300 or 505 fields requires many keystrokes--for example, in a contents note, typing (23:43) --

requires 14 keystrokes (including the shift keys necessary for the parentheses and the colon). By creating a combination macro consisting of a variable macro (for the minutes) and a fixed macro (for the seconds) this can be reduced to 7 keystrokes. Again, this may not seem like a big deal--unless you have a very long and complicated contents note. Keying in timings can get a little tedious (and susceptible to error) after typing in the 23rd timing. To create a macro which will make this process a little easier,

- **<Alt ==>**
- choose the macro key--say **<Alt t>**
- type (
- **<Ctrl />** (this invokes the "variable field" function, for the minutes)
- **<Ctrl />** (this ends the "variable field" function)
- type :
- **<Ctrl />** (this invokes the "fixed field" function, for the seconds)
- **<space><space>** (these two spaces are for the seconds)
- **<Ctrl />** (this ends the "fixed field" function)
- **)<space>--<space>** (this will position the cursor for the next item)
- **<Alt ->** (to end the macro)

To type timings using the above macro,

- **<Alt t>**
- enter the digit(s) for the minutes
- **<Enter>**
- enter the digits for the seconds
- the remainder of the macro is displayed on the screen, and the cursor is positioned for the next item.

6. Macros which Invoke Newkey Subroutines or which Execute Other Macros

It is possible to create a macro which executes a series of Newkey commands. As an example, I have put some of the steps for the copy/paste function into a macro.

Normally, after the cursor is positioned, invoking the cut/paste function requires 4 keystrokes: <Alt> <slash> u F1 (if you want to specify the key to assign the macro to). I've combined these into just 2 keystrokes: <Ctrl> u (mnemonic for "cut"). I won't go into details here, however, since doing this is a little more complicated than creating other macros.

The great flexibility of the program allows you to create various types of macros for different circumstances. It is not hard to create any of these macros or macro types. A macro has to be created only once; after your "stock" macros are created, most new macros are created by copying information off the screen.

Using Newkey is really quite straightforward. Though reading through the above examples might be tedious and seem complicated, the steps are easy to follow if you have the program available to work with. There are some tricks to using Newkey, and some problems to be aware of, including: 1) you can't always tell when Newkey is on; 2) you may get an "insufficient memory" message when running certain programs, especially the Microenhancer; 3) Newkey may not execute certain keys properly on your computer, especially the end-of-field marker <Ctrl> <Return>, <Esc>, and <Break> keys; 4) if you use Windows, Newkey has to be loaded in a certain order (before certain types of programs and after others); and 5) remember to save your macros before turning off your computer!

It is very easy to solve these problems (see Appendix 2) and I have never had any difficulty running Newkey on any machine (I have used it on 286, 386, OCLC, and non-OCLC computers) after configuring the program properly (again, see Appendix 2). It is compatible with Passport, with the Microenhancer, with WordPerfect (and is easier to use than the macro feature of the DOS WP51), and with PC-based catalog editors, so it has many uses beyond the possibilities shown here for creating original cataloging. It does not take long to learn: perhaps a morning, or an hour a day for a few days; different individuals can create their own files so different users can use the program on the same machine; and individual files can be copied to disk and used on different machines.

I should point out that some of you might feel more comfortable with other programs, such as Prokey. I cannot speak to them, for I have not used them, but the literature I have seen suggests that the general

advantages of Newkey apply to them as well. A brief list of such programs is included in Appendix 3.

I have found Newkey a very useful program in creating original cataloging and for eliminating some tedium in other applications as well (like typing my address in letters, logging into my e-mail account, and so on). While I have spent some time learning many of the details of Newkey in order to write this article, it isn't necessary to know everything the program can do in order to use it effectively. It takes very little practice to learn, and very little time for it to become a tool which can help eliminate unnecessary and repetitive typing, thereby freeing you to focus on the content of the work at hand instead of tedious and carpal-tunnel directed details.

Appendix 1: Sample Newkey Macros

This is a list of macros I have created for cataloging and other uses. I am not suggesting you create these specific macros, but this list may give you some idea of the variety of macros which can be used. These all are in one file; I have this list and their key assignments by my keyboard. I have omitted the key assignments below--while they usually mean something to me, they might not to you.

Special to music cataloging:

recordings fixed fields
 scores fixed fields
 007 for CDs
 007 for records
 007 for tapes
 033: London
 033: New York
 048
 subfields |b, c, x, etc., including
 |o arr., |h sound recording
 300 field for CDs, i.e.,
 300 1 sound disc (:) : digital, stereo ; ...
 300 field for records
 300 field for tapes
 500
 500 Sung in
 500 Compact disc.
 500 Program notes ... in container.
 505 0
 times (:) --
 times (:) ;
 511
 518 Recorded

650

Scores and parts.
Choruses, Secular
Choruses, Sacred

700 10

|4 prf

700 12

|f 1990

|f 1991

|f 1992[etc.]

710 20

740 01

Special symbols:

accent ´
umlaut ¨
flat sign

Passport functions:

cho af
sca pn
sca co
sca su
sca
hom, end

Labelling:

configure printer
sl6 (from anywhere on screen)
[f] (label for [folio])
Insert
[&1pt] (label for # of parts)
[&_pts]

File management, email, other:

ret;save (save a record in 1 step)
cancel record (1 step)
locfile.dat
locfile.idx
cut/paste macro for Newkey
setenv (for VT 100 terminal emulation)
email login (combines several steps)
email logout
my full address
my login name

Free keys (for storing on-the-fly macros):

Alt 2-0 e i j k m q v w y
Ctrl 1-2 a e g j l v w F10

Appendix 2: Further Hints for Using Newkey

When is Newkey On? You can't tell when Newkey is on except by hitting a macro, so the safest way to find out the status of Newkey is to hit an innocuous macro, or by hitting <Alt />, which brings up the Newkey menu without disturbing your other work. Even though Newkey can be turned off (by hitting <Ctrl 6>, and can be turned on by hitting the same key combination), the only way to remove the program from RAM is to reboot the computer.

Saving files: Save your macros! Whatever macros you create will disappear whenever you reboot your machine or turn it off unless you save them to a file. However, macros which have been saved will remain until they are deleted or overwritten, and the program gives a message that a certain key or key sequence has been used if you attempt to overwrite it.

Saving a macro file is quite simple. After you've created some macros, just press <Alt /> to bring up the Newkey menu. Hit <S>, and specify a filename (including the directory!) to which the macros should be saved. I prefer something straightforward, like

c:\newkey\cat

Of course, you can put this into a macro as well.

To open this file after Newkey is loaded, hit <Alt /> to go to the Newkey menu, hit <L>, and specify the filename.

If you want to add macros to an existing file, just have the file open as you create new macros. Save the file as described above, and the new macros will be added to the file.

Cutting and Pasting: There are various techniques for cutting/pasting blocks of text, or parts of multiple lines of text. Cutting one line of text, part of one line, or a block of many full lines is very easy to do. For one line, or part of a line, just position the cursor at the beginning of the text you want to cut, hit <Home>, and use the right arrow key to move to the end of the text you want; hit <End> to complete the procedure. To cut a block of several lines, position the cursor at the beginning of the text, press <Home>, and move the cursor with the down

arrow until all the text you want to cut is outlined. Press `<End>` to finish. Cutting and/ or pasting parts of several lines of text (rather than the whole line or complete contiguous lines) or a block of text that doesn't begin or end at the margins is possible, but requires some experimentation. Usually it is easier to copy part of one line and the entire following line or lines and delete the extraneous information later.

Installing Newkey: When you install Newkey, follow the directions which come with the program EXACTLY, or the program will not install properly.

Newkey is loaded when you switch to the directory you installed Newkey in and you type

`newkey <Enter>`

I find it easier to load Newkey through the following batch file (in the root directory). This will load Newkey from the C:> prompt.

- At the C:> prompt, type
- `copy con newkey.bat <Enter>`
- `cd\newkey`
- `newkey`
- `<Ctrl z>`

Then reboot your computer, type `<newkey>` from the C:> prompt, and Newkey will load. From now on, Newkey will load whenever you type `<newkey>` from the C:> prompt.

Newkey can have macro files of almost any size, but the default when Newkey is loaded is a 1,000 character maximum. This can be changed; I've found 10,000 quite sufficient. To allow for a 10,000 character macro file, type `newkey/10000` instead of `<newkey>` when you load the file. Or write a batch file:

- `copy con newkey.bat`
- `cd\newkey`
- `newkey/10000`
- `<Ctrl z>`

Alternatively, you can put newkey in your autoexec.bat file."

Newkey may not execute certain keys properly on your computer. I have found it dangerous to include the end-of-field marker `<Ctrl><Return>` in any macros, as including it in a macro may lock the keyboard. Newkey has a support, or sub-program, `Newkeysp`, which allows you to tell Newkey which keys to ignore. I included the `<Esc>`, `<Return>`, and `<Break>` keys in this, and have had no problems since. To call up this program:

- move to the Newkey directory (probably `c:\newkey`)
- type `newkeysp <Enter>`

- this brings up a menu; select *I* (keys to ignore)
- this brings up a second menu. Just select a line number and press the keys you want Newkey to ignore (`<Ctrl Break>`, `<Enter>`, and `<Esc>`). Note: the `<Ctrl Break>` will show up as `<UNK>` in this list.

This prevents Newkey from misinterpreting the ASCII codes assigned to these keys.

At the first menu you also have the option to set certain other parameters. To be on the safe side, set "slow typing" to *OFF*, "fast key" to *OFF*, "blank screen" to *OFF*. These parameters control the rate commands are sent from Newkey, rate of key repetition, and the screen saver feature. You probably won't need these features anyway, and disabling them from the start increases the probability that Newkey will not cause problems while running other programs. However, if you want to use these features, try them one at a time.

Memory Use and Memory Problems: You may get an "insufficient memory" message when running certain programs, especially the Microenhancer. It may be necessary to limit the amount of memory available for the macros, or it might be necessary to remove some other programs, particularly DOS programs which eat up a lot of RAM. Newkey requires about 60K of RAM. DOS versions 4.0 and higher have all sorts of "extra" programs which speed up the operation of your computer but also use quite a bit of RAM. However, DOS 5.0 or 6.0 on a 386 or 486 machine can be configured to run many of these programs in the upper or extended memory areas, thereby freeing up conventional memory.

Newkey and Windows: If you use Windows, Newkey has to be loaded in a certain order (before certain types of programs and after others--see the Windows manual). Do NOT use the screensaver feature of Newkey under Windows. I've also had problems loading (but not running) Newkey from various "shell" programs (like WP Office). I find it safest to load Newkey before most other programs, and just not use it (but leave it active in RAM) if it conflicts with other programs.

Appendix 3: A List of "macro" Programs

Newkey: This is a shareware program, and generally is not available from standard distributors. It is found in some shareware catalogs, or can be ordered from:

FAB Software
P.O. Box 336
Wayland, MA 01778
(508) 358-6357
compuserve: 75206,1366

Latest version: version 5.4 (1992), \$49.95.

Prokey: This program often is compared to Newkey in reviews. It comes in both a DOS and Windows version. It lists for \$99, and should be available from standard software distributors or from RoseSoft, Inc. (206) 454-7424.

Keywords: This should be available from standard distributors (latest version: 3.0, \$89) or from Alpha Software, (617) 229-2924.

I have seen references to RE-call, but have not been able to find a manufacturer or distributor.

Reviews of these and similar products often can be found under headings such as **utilities** or **operating system enhancements** in trade magazines such as *Byte* or *PC Computing*.

David Lesniaski
St. Olaf College

SMALL GROUP ACTIVITIES II

WORKFLOW AND STAFFING

The session was led by Sue Weiland, Wichita State University, and Lynn Gullickson, University of Wisconsin-Madison.

Discussion centered around two views of workflow and staffing: centralized and decentralized. Discussion topics included integration of special projects into workflow (including problems and opportunities of central and decentralized environment), and problems finishing (or picking up halfway finished) special projects.

One prevalent problem in a centralized environment was that the music cataloger is usually required to spend time cataloging other subject areas, and must also supervise special projects that may or may not relate to music. Music catalogers in a decentralized environment often have to juggle many special projects as well, but usually in music only. All expressed frustration with trying to finish special projects, usually due to funding or

a turnover in personnel (graduate students). Frustration with lacking enough money and staff to experiment with workflows was also expressed.

Laura Gayle Green
University of Missouri-Kansas City

OCLC'S FIRSTSEARCH

This session focused on FirstSearch from the reference librarian's viewpoint. Moderator Leslie Bennett (University of Oregon) outlined her institution's use of FirstSearch as part of the reference department's arsenal, and discussed the pros and cons of the system. Positive aspects include the user-friendliness of the database; the ability to search numerous ways, a boon to music searching; and the multitude of databases available through this service.

Negative aspects centered around the local hardware used to do the searching, as well as the hours that FirstSearch kept for the West Coast--things that will be resolved in future policy changes and library hardware purchases. Another fault noted was the tardiness--or lack--of documentation for the many new databases added to the FirstSearch family, leaving reference librarians unprepared to answer their users' questions.

This was followed by a discussion of various aspects of using FirstSearch, primary among which were questions of funding. Attendees asked where funding was obtained, and discussed the funding options--subscription versus package pricing--that OCLC offers for accessing FirstSearch.

Leslie Bennett
University of Oregon

AUTHORITY WORK

The session was led by Neil Hughes, Music Cataloger, University of Georgia Libraries. He offered as a springboard for discussion a paper on the need for authority control, originally prepared for "non-cataloger" librarians at the University of Georgia. To summarize the main points of the paper: Authority control is the mechanism by which the catalog fulfills its function of finding and gathering. Authority control is the process of selecting and establishing unique, standardized headings to be used as access points in the catalog, and of relating variations and other headings to them. It is