Hebrew Word Processing

Aaron Wolfe Kuperman
Brooklyn Public Library
Brooklyn, NY

Introduction

Personal computers have brought about substantial changes in the way American librarians work. Any text that needs to be typewritten can be processed faster and more efficiently with a computerized word processor. Computer-readable files of documents created elsewhere, including cataloging records, can be copied automatically for local editing without retyping. Newsletters and bibliographies produced through computerized "desktop publishing" approach the appearance of expensive printing, but require less effort than using a typewriter. In all areas of public and technical services, libraries are among the leading beneficiaries of the revolution in small, personal computers.

Anyone trying to work with Hebrew text on a personal computer tends to be envious of colleagues whose computing is limited to the Roman alphabet. At this point in the development of personal computers, a user can walk into a store and walk out with a computer that will run virtually any software "off the shelf." When one hits an "a," one can generally assume that an "a" will appear on the screen and will eventually be printed when the printer receives a command to print the "a." Virtually all systems can produce output that any other computerized system can utilize.

When a usually user-friendly computer is asked to attempt Hebrew word processing, it suddenly becomes an anti-Semite (literally; Arabic word processing is even harder than Hebrew, and many Asian scripts are even more difficult. *See:* Becker, Joseph D. "Multilingual word processing," *Scientific American*, July 1984, p. 96–107). No "off the shelf" computer will willingly display Hebrew characters or write right-to-left (backwards as the *goyim* say). Printers can print Hebrew only as well as they can print other non-standard characters (e.g., italics), which is to say that if one wants fast, cheap, high-quality printing, one will have to settle for two out of three. There is currently no "standard" system for recording Hebrew characters in machine (computer)-readable format, meaning that files produced for one system are probably unintelligible on any other system.

Entering the world of Hebrew computing is like taking a time machine to what English word processing was like several decades ago. Except for "hackers" who enjoy doing things themselves, or someone who considers backwardness to be lovably "quaint," Hebrew word processing is still extremely frustrating.

Hebrew Coding

A computer can only tell if the state of something is *yes* or *no*, or if a switch is either *on* or *off*. With one switch, there are two possible outcomes, *on* or *off*. This *on* or *off* is the smallest unit of information, and is commonly called a *bit*. With two switches, or bits, there are four possible combinations (both on, both off, only first one on, only second one on). With seven bits, there are 128 possible combinations, and with eight bits, there are 256 possible combinations. A generation ago, a standard system for representing the American "character set" was developed using seven bits. The 128 possible codes represented upper and lower-case letters, all ten numerals, assorted punctuation marks, and several codes necessary for controlling a printer. These 128 standard "ASCII" codes are uniform throughout United States computers, and with minor modifications, are standard throughout countries using the Roman alphabet.

For a variety of reasons, computers prefer data in units of eight bits, commonly referred to as *bytes*. A byte can represent up to 256 different things, in our case, characters. Because of the 128-character ASCII set, the first 128 possibilities are "spoken for." While many personal computers use IBM-style coding for the "upper" ASCII characters (i.e., characters 128 to 255), which facilitates foreign accents, scientific coding and certain graphics characters, everyone understands that there are no guarantees that one's computer will recognize any code other than the standard "lower" ASCII characters (i.e., characters 0 to 127).

ASCII 97 is always a lower-case "a," and any computer or word processing program will display and print this letter when asked to display or print ASCII 97. When receiving a command to display ASCII 97, the computer system looks to its character generator, which is part of the unchangeable (but circumventable) ROM—Read Only Memory, to find out how to display an "a." This type of standardization is totally lacking, at present, for Hebrew.

Three methods are currently available to represent Hebrew text in computer-readable format. One possibility makes use of ASCII characters 128 to 257. The second possibility utilizes the ASCII symbols for lower-case Roman letters to represent Hebrew, thereby rendering it impossible to produce a document including Hebrew and a full, i.e., upper and lower-case, English character set. A third possibility uses two bytes (characters) for each letter, the first byte indicating which alphabet is in use, and the second byte indicating which letter.

The result is total incompatibility. Not only must each system give special instructions to the computer and printer as to what to display or print when it receives any of the ASCII codes used for Hebrew (i.e., to ignore the character generator in ROM and print a Hebrew character for a given ASCII code), but the various systems for Hebrew are totally incompatible. One system may use ASCII 97 to represent an *alef*, a second system may use ASCII 152, while a third system may represent an *alef* with two characters, the first to indicate Hebrew, and the second, perhaps an ASCII 97, to indicate the *alef*.

Hebrew Character Sets

Prospective buyers of Hebrew word processing systems have to do a lot more shopping around, a lot more research, and often settle for a more expensive system that does less, than someone wanting an English-only system. Whereas top-of-the-line ("bells and whistles") word processing programs cost under $300, Hebrew systems usually cost about $500 and are inferior in terms of features other than the ability to do Hebrew. Furthermore, whereas one can normally buy virtually any computer, and then find the appropriate version of desired software to run on the system selected, when shopping for a Hebrew word processing system, one has to carefully coordinate the purchasing of all hardware (the computer and printer) and software (the word processing program).

The usual way a word processing system displays characters is: when told to display an ASCII code, it looks at the character generator to see what to put on the screen. As a fairly standard option, most computers use a "graphics card" (an additional hardware element) that makes it possible for software to control the image on the screen on a dot-by-dot basis, thereby facilitating production of better and more characters than can be done through the character generator. Since the standard character generators are not programmed to display Hebrew characters, in order to display Hebrew text, one must either replace or circumvent the character generator.

Many systems come with what is commonly referred to as a "Hebrew chip," which is a modification to the hardware that produces Hebrew characters in response to various commands. This chip either supplements or replaces the standard character generator. Since it is hardware, it tends to be fairly expensive, and cannot be easily inserted and removed; however, it is always in the computer and one can usually use it with whatever software is running. For example, if the regular software wants to display "ñ" for ASCII 164, the chip might change that to a *khaf*. If such is the case, depending on the chip, whenever ASCII 164 is called up, even in a program not designed for Hebrew, a *khaf* will appear. In some cases, this means that the computer can insert Hebrew characters into any program; in other cases, it means that any program not designed for Hebrew characters will "bomb" (not work) when run with the chip in the computer.

The second alternative is to have the Hebrew characters produced by software. This requires having a graphics card. Whenever a character is requested, the program, in effect, detours away from the character generator and looks to the program for the specifications of what to put on the screen. The Hebrew program must be running whenever Hebrew characters are desired for screen display; absent the Hebrew program, only the regular character set will be displayed. Avoiding purchase of a specialized Hebrew chip saves money, though in general, a program that utilizes graphics to produce text is slower than one using the character generator or a special "chip."

A third possibility is becoming available. The leading producer of graphics cards has a new model that in effect creates a programmable character generator with the capability of dealing with a dozen fonts simultaneously (offering a maximum of 3072 characters instead of the 256 that is the current norm). Since this will probably become a "standard" piece of hardware (unlike a specially manufactured Hebrew "chip"), it will be more compatible than the chip, with other software. This new card will combine the speed of relying on the character generator with the flexibility of graphics-based screen display. In theory, one programs the card's alternative character generator to produce Hebrew, and it remains programmed until it is reprogrammed. This card is still very new, but *if* it works as advertised, most word processing programs will probably be modified to exploit its features.

Several additional considerations remain for anyone shopping for a Hebrew word processor. The simplest and least expensive systems produce Hebrew consonants and upper-case Roman letters. Many systems produce a complete Roman alphabet and Hebrew, but some do not offer vowels or special symbols useful for writing other languages that use the Hebrew alphabet, such as Yiddish. Some systems offer not only Hebrew and Roman characters, but other alphabets as well.

A person writing business letters almost exclusively in Hebrew, may have no need for vowels, and while he might need to insert an occasional word in English, he would not need the lower-case English letters. Most bilingual documents require a full English character set (including lower-case Roman letters), but not Hebrew vowels. Vowels and other diacritics are required, however, for many documents produced for religious applications, almost any text discussing Hebrew grammar, and virtually everything produced for children. Many libraries, scholars and community groups may require additional scripts if they need to produce documents in Arabic and Russian.

Bi-Directional Wordwrap

Inserting Hebrew text, even as transliterated "gibberish" (e.g., reversible Romanization) into a system not designed for Hebrew isn't difficult. Assuming one has a printer that will print said gibberish as Hebrew, it is quite reasonable to input transliterated text to be subsequently printed as Hebrew. Getting the computer to type "backwards" is not a major problem (in insert mode one can simulate backward typing by backspacing after each keystroke). *Wordwrap* is a different story.

On a typewriter, a bell generally rings at the end of each line, and one physically returns the carriage to the beginning of the next line. On a computer, that is not usually the case. One keeps typing to the end of the paragraph, and the computer automatically fits as much text as will fit within the prescribed margins on each line, and then goes to the beginning of the next line. When typing a left-to-right document which includes an occasional Hebrew word, this works reasonably well. If one is trying to insert several Hebrew words into a predominantly English document, however, or is writing a document that is truly bilingual (alternating between Hebrew and English with substantial segments in both languages), the standard wordwrap systems will not work.

Assume that two words of Hebrew are to be inserted into a predominantly English document, and the line comes to an end between them. Wordwrap logic designed for English will put the first word at the end of the first line, and the second word at the beginning of a second line. Since the two-word Hebrew phrase is written "backwards" (from the perspective of the English-oriented wordwrap program), the program will end up putting the second word of the Hebrew phrase at the end of the first line, and the first word at the beginning of the second line.

Some programs have a wordwrap feature that can work with Hebrew or English. This can backfire. Using the above example, the program will accurately place the first word of the Hebrew phrase at the end of the first line, but upon seeing that the next word is Hebrew, it will place it at the right margin of the next line (which from a Hebrew perspective is the beginning of the next line). The rest of the document is in English, however, and it might try to print that English text starting at the left margin of the same line. The absurd result is that the first word of a Hebrew phrase will be placed at the end of the first line (i.e., the right margin where it should be), and reading left-to-right, the next English sentence will be found on the following line, followed by the second word of the Hebrew phrase from the preceding sentence.

To meet a wide range of needs, a Hebrew-English wordwrap feature requires considerable flexibility. It must be powerful enough to adjust to a document that is predominantly right-to-left, predominantly left-to-right, or mixed. Some systems do this, but some do not. Using wordwrap, one of the most basic word processing features, is possible in Hebrew-English word processing, but it requires great care by the user in selecting and using the system.

Different Computer Systems

There are several "lines" of personal computers currently available. The most popular in offices is the line of IBM-compatibles. This includes personal computers produced by IBM, and a wide variety of computers designed to run software produced for the IBM (or to be technical, to use software written to be compatible with the MS-DOS operating system common to both IBM and the compatibles). To a certain extent, this is the de facto standard computer, and while it has many deficiencies, its leading role means that a wide variety of software is available for it.

Its leading competitor is the Macintosh computer produced by Apple Corporation. While the "Mac" can produce ASCII output just as well as an IBM-style computer, it is a very different computer. Interestingly enough for Hebrew word processing, the Mac is designed to work with a variety of fonts. This feature was apparently designed to allow display of different Roman fonts, but regardless of motivation, it is easy to exploit for Hebrew. Using suitable software, a Mac can display Hebrew with substantially less difficulty than an IBM. Several systems now exist to do Hebrew word processing on the Macintosh.

Several other computers, such as the Commodore and the regular Apple II family of computers, are generally used for relatively simple educational applications and games; however, there are viable systems for Hebrew word processing for these relatively inexpensive computer systems, though there are fewer software packages available for them.

Printing Hebrew

Getting a computerized system to print Hebrew characters is relatively easy. The degree of difficulty is no greater than that involved in making a computerized system print italics. While fast *and* cheap high-quality results are unavailable, one can get letter-quality results either fast *or* cheap, and *almost* letter-quality results both fast and cheap.

There are two commonly used ways to print a document. One is with a formed character system, similar to that of a typewriter. The results are, by definition, "letter quality" (i.e., equivalent to a good typewriter's output). The most common version of this method for a personal computer is a "daisy wheel." Like a typewriter, this prints by having a backwards image of the desired letter hit the paper through an inked ribbon. The disadvantage of a formed character system is that one is limited to a fixed number of characters by the daisy wheel or typewriter.

The second way is to have the computer, or a computerized element in the printer, make a grouping of dots that resembles a letter. In its cheapest form, a simple dot matrix printer, this method is fast and the output "looks" as though it was printed by a computer. Its most expensive form, a laser printer, produces results that are at least as good as "letter quality."

Most "daisy wheel" systems can handle about 100 characters. Thus one wheel can print a standard ASCII character set, and maybe a few extra characters. If additional characters, such as italics are wanted, extra wheels are needed. There are daisy wheels with both English and Hebrew characters, but unless one is content with vowelless Hebrew and only upper-case English, Hebrew-English word processing requires two daisy wheels. Occasionally, there are references to printers with two daisy wheels, or a daisy wheel with 200 characters (i.e., two complete character sets), but these are not widely available at this time.

Printing more than a single character set (such as full Roman and full Hebrew character sets) means using extra wheels. In practice, this requires sitting by the printer while it is printing, and when the text indicates the need for a different wheel (i.e., to change from a standard Roman typeface to Hebrew or italics), the user manually changes the wheel. Many word processing programs thoughtfully support such changes by beeping when it is time to change wheels, and indicating on the screen which wheel to insert; however, having to sit next to the printer changing daisy wheels by hand is rather tedious and totally negates many of the advantages of a computerized word processing system.

The alternative is a system that—instead of printing whole letters on the paper—places dots. A good system can place 90,000 dots per square inch, which is at least as good as a typewriter or daisy wheel (i.e., "letter-quality"), and approaches the results one gets from conventional typesetting. A major advantage of a dot matrix or laser system is that one can integrate a variety of fonts (such as Hebrew or italics) without difficulty.

A cheap dot matrix printer costs a few hundred dollars, is very fast, and produces rather tacky looking output. It is generally considered unacceptable for a business letter or any other application for which a high level of readability and neatness are desired. A laser printer, which costs several thousand dollars, produces results equal to or better than a formed character printer, and works faster than any other type of printer. A good dot matrix printer costs about a thousand dollars and produces results that approach those of a laser printer almost as fast as a cheap dot matrix, but at a speed well below that of laser.

Many dot matrix printers support user-designed fonts. If the system does not produce the desired fonts, the user can design his own, dot by dot. Various software systems facilitate designing one's own fonts, though this may not be simple. Some software systems designed to enhance inexpensive dot matrix printers will cause them to produce a wide range of characters similar in quality to that of the best dot matrix printers, but at speeds that are unreasonably slow, in part since in order to improve the quality of a dot matrix printer's output, a system must place more dots on the paper, and printing additional dots takes additional time.

Even if a computer cannot display Hebrew characters, a printer can independently print Hebrew characters. A text might call for the printer to print "a" in font three, and if font three is Hebrew, and ASCII 97 (lower-case "a") in that font is *alef*, then the printer will print *alef* even though the screen shows an "a." Producing Hebrew output without a Hebrew word processing system is difficult since one must enter the Hebrew text in Roman letters.

Both dot matrix printers and formed character printers use the same ASCII codes. Since there are no standard codes for Hebrew, in order to print Hebrew, the word processing program has to either produce ouput designed for the specific printer in use, or address each dot in controlling the output (a job usually done independently by the printer). This problem is not insurmountable, but it demands more care and expense than is required of someone working with standard Roman alphabet text.

A third printing option exists, but hasn't been exploited widely, especially for Hebrew. One can have a fast inexpensive printer, probably a dot matrix, for rough drafts and personal non-business correspondence, and then arrange to have someone else produce letter-quality (or typeset) output for important documents. Such arrangements with commercial typing services, typesetters, and better-quality photocopy shops exist for English word processing output, but would at present be awkward for Hebrew since every word processing system uses different codes. This might become a popular alternative as the number of people doing Hebrew word processing increases, and if the codes become standardized.

Printing Hebrew is relatively easy, at least compared with the problems of producing and editing Hebrew text. Many computerized printing systems were designed with the goal of facilitating unusual fonts. Since such systems print without human involvement, and, in fact, this is one of their major advantages over manual systems, the direction of the text is totally irrelevant. The only serious problem is the lack of standardization in coding Hebrew. This necessitates using a printer programmed for a given word processing system, or requires the word processing system (or even worse, the user) to, in effect, program the printer to print the bilingual output of the word processor.

Summary and Conclusions

Some of the complexities of Hebrew word processing are inherent in the differences between Hebrew and English, such as the problem of producing a document containing text oriented in two different directions. This sort of problem will be solved as more people get experience in using and designing Hebrew word processing systems. Other problems are somewhat political, such as weighing the interests of the businessman or government official who does not use vowels in correspondence versus the interests of scholars and teachers for whom the full range of Hebrew diacritical marks is essential.

Lack of standardized coding for Hebrew is perhaps the most serious problem. With standardized coding, files from one system would be transferable to any other. One could buy a screen display system from one vendor, a text processing system from another, and a printer from yet a third. This flexibility exists for English word processing systems and is highly beneficial. Excluding the kludged ("quick and dirty") system of putting Hebrew characters in place of lowercase English (thereby rendering it impossible to produce respectable English correspondence), two coding systems are being widely used.

One coding method uses the "higher level ASCII characters," that is to say, ASCII 128 through ASCII 255. A major disadvantage of this method is that not all computer-related equipment can process higher ASCII characters, since some equipment is designed to work only with the seven-bit lower ASCII characters. As stated above, the "upper" ASCII characters are not standardized. Hebrew is competing with a variety of other non-Roman scripts for use of the upper characters, including Arabic, Greek, Cyrillic (Russian), and the various South Asian (Indian) languages; when all forms of all these alphabets are considered (including both upper and lower-case forms of Roman letters, initial, medial and final forms of Arabic letters, etc.), it is unlikely that a generally accepted international standard governing the upper ASCII characters would include Hebrew. A convention (other than an official international standard) for expressing Hebrew with "upper" ASCII characters for Hebrew word processing could be developed, however, and it would be possible to make such a convention compatible with the two-byte "standard" that will probably develop for defining Hebrew characters.

The only serious problem is the lack of standardization in coding Hebrew.

The other system uses, in effect, two characters for every letter, one to indicate the alphabet being used, and the second to indicate the letter. This offers great flexibility, since 256 different character sets can be indicated (through the first character), plus 256 characters within each character set. The drawback of this method is that every character requires two bytes instead of one, meaning that it will require twice as long for the computer to process, and take up twice as much storage space. If a standard system for indicating fonts were to include a device for indicating the Hebrew character set, then Hebrew word processing would become almost as easy as English word processing. Given implementation of the proposed character set being developed for Hebrew by the Research Libraries Group in cooperation with the Library of Congress, and the structure of the *Hercules Graphics Card Plus RamFont* system, this alternative may be the way of the future.

Appendix

Sources of Product Information

This list is by its nature incomplete. New products are constantly being introduced, and as is the case with personal computing in general, all existing products are regularly being improved. Differences in availability exist even between New York and California, and all the more so between North America, Europe and Israel.

Hebrew word processing programs

This list features the better-known products for MS-DOS (IBM-compatible) and Macintosh computers that are available in the New York area in the spring of 1987; the list in no way attempts to be comprehensive or critical. With programs enumerated below, one can display, edit, and print Hebrew characters. All programs are for MS-DOS unless stated otherwise.

Achbar, Davka Corporation, 845 N. Michigan Ave., Suite 843, Chicago, IL 60611, (800) 621–8277. Formerly called "Mouse Write," this is a Hebrew-English word processing program for the Macintosh. The company markets a variety of Hebrew-English word processing programs, as well as other types of Jewish-oriented software for a variety of computers.

Alef-Bet, Quad Inc., 23601 Draco Way, Canoga Park, CA 91307, produced by Grynberg Engineering in Israel, is a word processing system that uses a special "chip" rather than software. This program is widely used in Israeli government offices.

Computer Linguist, P.O. Box 70742, Eugene, OR 97401. Formerly known as *Pangloss*.

Intext, Intex Software Systems International, Ltd., 488 Madison Ave., New York, NY 10022. Software-based program for Hebrew-English word processing, that also has the capacity to do Arabic and Russian scripts.

MacInHebrew, c/o Joseph Weinstein, MIT Hillel, 312 Memorial Drive, Cambridge, MA 02139. This is a non-commercial "shareware" system for Hebrew-English word processing on Macintosh computers.

Mince, Davka Corporation (address above), is a word processing program for MS-DOS computers that can be either software or chip-based.

Multi-Lingual Scholar, Gamma Productions, 710 Wilshire Blvd., Suite 609, Santa Monica, CA 90401. Software for multi-lingual word processing in Hebrew, Roman, Cyrillic, Arabic, Greek, and other scripts. Supports all Hebrew vowels, including special characters used in Yiddish.

Kuperman (*Continued*)

Wordmill, Bigger Byte, Inc., 1 S. Central Ave., Valley Stream, NY 11580. This is an Israeli word processing program based on a Hebrew "chip."

Font programs that include Hebrew

These programs work with dot matrix or laser printers to produce a greater range of fonts and/or higher quality fonts than would otherwise be available. They supplement but do not replace a word processing program. The programs listed below include Hebrew fonts.

Fancy Font, Soft Craft, Inc., 16 N. Carroll Street, Suite 500, Madison, WI 53703.

Fontrix, Data Transforms, 616 Washington Street, Denver, CO 80203.

Lettrix, Hammerlab, 5700 Arlington Ave., Riverdale, NY 10471.

User groups

User groups often provide honest and independent information about using computers. The two listed below specialize in Hebrew and Judaica applications of personal computers, and both publish newsletters containing valuable and current information.

Hebrew Users Group, Berkeley Hillel Foundation, 2736 Bancroft Way, Berkeley, CA 94704.

Computer Hebrew Users Group of New York, c/o Michael Rand, 21 Bennett Ave., New York, NY 10033.

Aaron Wolfe Kuperman is a Librarian working for the Brooklyn Public Library in Brooklyn, New York. The paper is based on a presentation at a workshop on Hebrew computing held at the Jewish Theological Seminary on Jan. 19, 1987 under the sponsorship of the New York Metropolitan Area chapter of the Association of Jewish Libraries.

The Library Automation Scene in Israel—1986

Elhanan Adler
University of Haifa

Introduction

The major development in the Israeli library automation scene during 1986 was the large number of libraries initiating automated services. In virtually all cases, the libraries chose one of two commercially marketed Israeli library systems: ALEPH or Sifriah-83.

ALEPH—Libraries

In 1985, I reported on the selection of ALEPH by the University Grants Committee as the basis for the university-wide network and on the changes in the network concept towards a highly decentralized system (Adler, 1985). During 1986, the decentralized VAX computer version of ALEPH was installed in several Israeli universities.

The initial intent was to run-in the ALEPH VAX version at the Technion-Israel Institute of Technology while maintaining the Hebrew University's older CDC version until the ALEPH VAX version was completed and fully tested. This plan was unfortunately not followed, as the Hebrew University found it necessary (primarily for financial reasons) to drop its leased CDC sooner than planned. Testing at the Technion went on simultaneously with conversion of the various Jerusalem-based libraries (each with its own particular problems which had to be solved in the process) and, as a result, the run-in procedure did not progress as smoothly as it might have.

The Hebrew University itself is now running ALEPH on several VAX computers at its various campuses. The catalogs of the Library of Social Sciences and Humanities at the Mount Scopus campus and the Library of Sciences at the Giv'at-Ram campus have been completely converted, and these libraries function totally online, without a card catalog. Other University libraries (Law, Education, Archaeology, and Agriculture) are in the process of catalog conversion; however, public access terminals are not yet in use. The Jewish National and University Library (JNUL) closed its card catalog in 1985, and all materials cataloged since then are accessible via computer only. There are no plans for mass conversion of the old JNUL card catalog in the foreseeable future. Each Hebrew University library catalog is a separate, independent database with its own authority files—making a campus-wide search rather difficult.

The Technion was the first major library system outside the Hebrew University to use ALEPH VAX, and it has served as the test site for various ALEPH modules. Despite the fact that the Technion has a highly decentralized system of 24 libraries, it is maintaining a centralized automated system with a single database of all its libraries' holdings. As of April 1987, the central library and five departmental libraries are online, including circulation. Other libraries await conversion and a larger computer; meanwhile, they are using a campus-wide COM microfiche catalog, in addition to maintaining their card catalogs. All Technion cataloging since 1981—covering a very high percentage of "live" material in technology—is in the database.

The Ben Gurion University of the Negev and Tel-Aviv University libraries have also moved their files from Jerusalem to local VAX computers. Ben Gurion plans to have public access terminals available in Fall 1987; meanwhile, current cataloging and retrospective conversion are proceeding. Circulation via ALEPH is limited to those items already in the database; a parallel manual system is still in use for unconverted materials. Tel-Aviv University activity consists also primarily of current cataloging and conversion at this stage. Tel-Aviv will have a multiple database system, reflecting its system of several large, highly autonomous libraries.

The Weizmann Institute of Science Library is committed to ALEPH also, even though its central library is currently using the microcomputer-based "Sifriah-83" system (see below).