[*Translating and the computer 13: The Theory and Practice of Machine Translation – A Marriage of Convenience?*
Papers presented at a conference … 28-29 November 1991 (London: Aslib, 1992)]

DEALING WITH MULTIPLE LANGUAGES IN THE COMPUTER INDUSTRY

John Clews

SESAME Computer Projects

INTRODUCTION

This paper describes how scripts work, how languages work, and how they interact particularly through character set standards. It looks both at the development of such standards and their implementations in actual systems, particularly in the PC environment. In his related paper, John Parry describes some of the implications of character set handling and some problems of conversion and compatibility. In conclusion, particular trends in providing language facilities are highlighted, from producing localised versions of particular software packages to an increasing global provision of language features in the basic design of a computer.

SCRIPTS

Although there are several thousand languages worldwide, fortunately most of our uses of computers use written forms of language, which are much more standardised than spoken forms. The hundreds of languages in current commercial use are, or can be, rendered in at most two dozen scripts, and there are only three basic types. Even so, human ingenuity demands a lot more from the computer than is usually provided. Even more fortunately, there are only three basic script types which have several major scripts derived from them. Figure i indicates their geographical distribution and their cultural roots. As a rule, each script tended to follow specific religions and cultures, and also to develop more letters the further removed from the source script type.

Computers then ...

In 1984 I chaired a similar session to this[1] at Translating and the Computer number 6. There are a number of contrasts between the situation then and now. Firstly the range of facilities available. Most computers were big mainframe systems, serving multiple users in offices and laboratories, and were geared up for business use, which tended to mean number crunching and English language business text support. After all the almighty dollar meant that most business was conducted in English, and anyone else wanting to do business only wanted English letters, or "cut down" foreign languages with the accents missing. If you asked for foreign language support, salesman would tell you "Oh yes, we can provide an AZERTY keyboard for you, but accented characters: well, you can manage without those, can't you?" Even when accented characters were provided for specific languages, all sorts of anomalies

[1] Clews, John. World scripts (background paper). Translation and communications edited by Catriona Picken. London: Aslib, 1985 (Translating and the computer; 6), pp. 147-165.

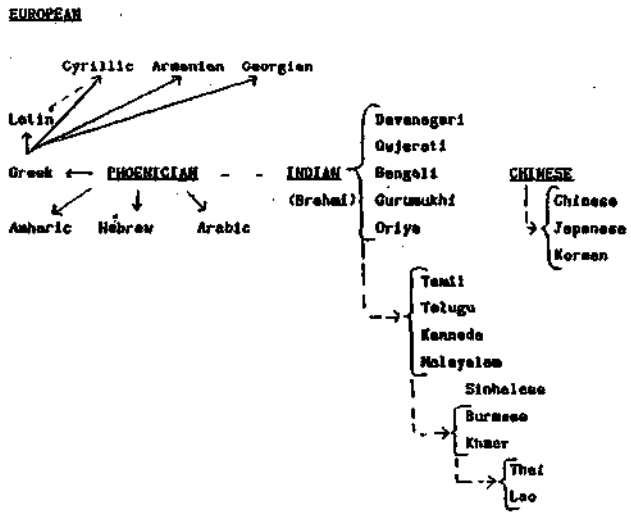


Figure 1 Distribution of major scripts

between typed text and printed text occurred, and computer dealers frequently found themselves unable to cope with users wanting such apparently exotic requirements as both French and German text together.

There was linguistic software available, but often this was only in big organisations where customised language systems might be developed on mainframe computers if the translation department had sufficient clout. At the other end of the spectrum, some small non-standard computers supported language friendly features, where software packages were developed for very small niche markets, particularly where educational computing was developing, e.g. using the BBC micro in the United Kingdom. Although these were tiny in their capacity, and were often frowned on and regarded as toys by computer professionals, much of the best language support was available on these computers in the 1980s. Some of it remains more highly developed than some standard PC software from a linguistic point of view.

By way of example, I would like to quote Brian Gould, one of the few British Members of Parliament who seems more geared up to using information technology rather than quill pens[2]. He claims to have been one of the very first MPs to own a word processor, back in the early eighties, and still remembers the feeling of excitement and liberation after mastering the intricacies of his Apple II... It cost him £2,500, for which he could now get half a dozen PC 286 machines with hard discs if he shopped around.

[2] Gould, Brian. Word processing and printers. Computanews, no 55, September/October 1991, p. 8.

He and his staff own, between them, two ancient Apples, two Amstrad PCWs and an Atari ST. Although these are regarded as toys by "serious" computer professionals, some of these are serious business machines, particularly in mainland Europe, and I suspect many here have, or would have had in their early days, used them because of the linguistic features available.

... and now

Fortunately, more language-sensitive software can now be expected as a standard feature from many hardware and software manufacturers. The IBM PC takeover worldwide and the mushrooming of compatible software has provided a buyer's market with more sophisticated software, demanded by increasingly more sophisticated users, at lower prices. The growth of alternative operating systems, and the development of Graphical User Interfaces provided by the Apple Macintosh and Microsoft Windows has meant far more can be done with text. These make things easier for users for whom the computer is a tool of their work, rather than an object of interest in its own right and solely the province of computer professionals. Many applications software packages and even operating systems now provide more "linguistic" features as standard. There is a great variety of price and performance – not always directly related.

However, this wealth of software becomes *un embarrass de richesses* and makes for a new problem of finding information on the appropriate software for your own needs. In fact one of the great problems is finding out information on language automation. One needs to shop around, but until now, relevant information has not always been easy to come across. The multilingual PC directory by Ian Tresman now provides a much needed information source to the maze of language software available. Although not the first directory, it is in a different league to its predecessors through being much more comprehensive, with a mass of useful – and readable – technical and product information on language automation. As well as useful information on languages, this directory gives a lot of extremely well collated information on PC hardware, software packages, code pages and operating systems like MS-DOS, as well as software packages, such as word processors, databases and spreadsheets.

Brian Gould also cites a particular problem with his older machines. Not only are they incompatible, they even all use different disc sizes. Despite the widespread use of similar computers, language related issues mean that incompatibility of machines is a problem, and not even having the same make and model of computer, or using the same or similar character set standards will solve all your problems of compatibility, as John Parry's paper indicates.

STANDARDS

The need for character set standards arises from the fact that computers deal in binary digits, usually arranged in groups of 8 bits, or a byte. To humans, a rose is a rose is a rose, but to computers these are basically strings of zeros and ones, and the same combinations must be agreed in advance if "a rose" is to appear on all other computers that wish to display or use the same text. The earliest character set standards generally used 7 bits, as one bit was frequently reserved for "parity checking" to identify frequent problems in data transmission.

7 bit character set standards

The American Standard Code for Information Interchange (ASCII) was developed by industry experts in the USA during 1977, and provided a standard coding for upper and lower case English letters, numbers and punctuation, as well as control characters such as carriage return and line feed (Figure 2).

Working together with overseas experts in the International Organization for Standardization (ISO), ASCII in effect became International Standard ISO 646, with 94 graphic characters available. It permitted variant codes for accented characters, and it was adapted and adopted as a national standard by many other countries, although people interchanging data often found annoying appearances of curly brackets when they wanted accented letters, and vice versa.

Another international standard, ISO 2022, allowed various character sets to be identified and invoked in the middle of a data stream, and an international register of 7-bit character sets was maintained to this end. A large repertoire of characters could be provided on suitable equipment though a rather messy procedure of swapping of character sets using escape sequences. This was only suitable in the days when relatively small amounts of data was exchanged, and users were prepared to wait for specialists in data centres to sort out their problems. This technique was particularly used in very large libraries for exchanging bibliographic data.

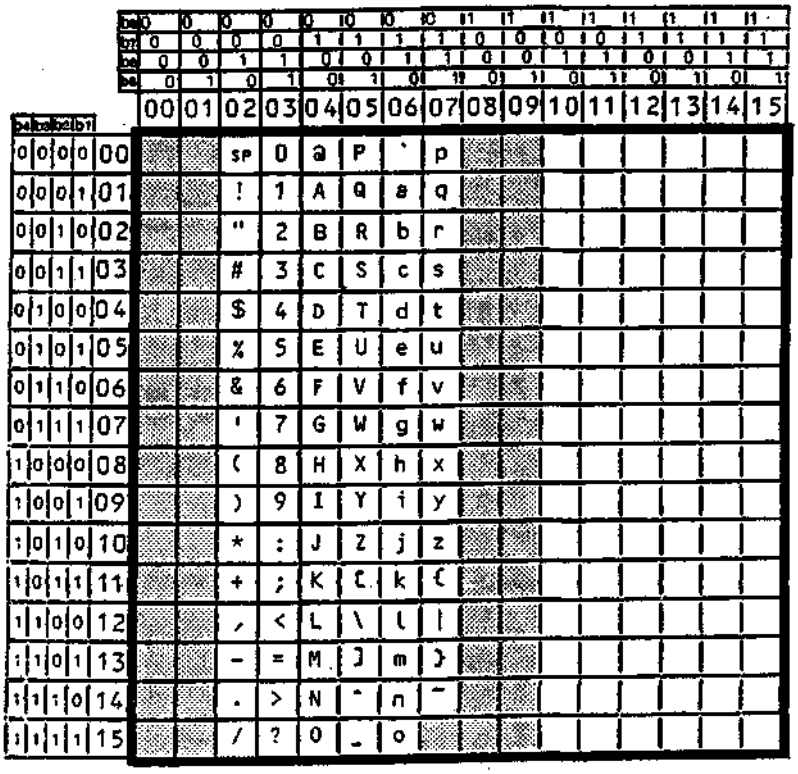
Multiple byte 7-bit character set standards

ISO 2022 and the closely related register of coded character sets also allowed 7+7 bits (94 x 94 characters) to be used for the Japanese national standard. The Chinese national standard and the Korean national standard both followed the general layout of the Japanese standard, and each duplicated many Chinese characters found in the other East Asian national standards. The ISO 2022 technique was also used by American librarians for their 7+7+7-bit character set (East Asian Character Code, or EACC) based on the Chinese Character Code for Information Interchange developed in Taiwan, which did not duplicate different Chinese characters.

8-bit character set standards

8-bit character sets mean a maximum of 256 characters. 8 bits are the current norm for character sets, which means that there is only room for either (a) English and Western European accented letters, or (b) English and one other script (such as English and Arabic). Although ASCII was universally used in the 7-bit world of the 1970s, and also formed part of many 8-bit character sets during the 1980s, standardisation broke down to some extent during this period, and several completely different 8-bit character sets have been called ASCII-8.

In the computer industry, the dominance of the IBM Personal Computer, and of MS-DOS, with its 8-bit character set (code page) which provided accented letters for most West European languages, provided a major standard and stimulus for other software to be developed, and the computer industry, including printer manufacturers, generally worked to this de facto standard. (However, the printer fonts supplied by Hewlett-Packard and Adobe companies differ considerably from the IBM PC character set).



b4 b3 b2 b1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0 0 0 0			sp	0	@	P	`	p								
0 0 0 1			!	1	A	Q	a	q								
0 0 1 0			"	2	B	R	b	r								
0 0 1 1			#	3	C	S	c	s								
0 1 0 0			$	4	D	T	d	t								
0 1 0 1			%	5	E	U	e	u								
0 1 1 0			&	6	F	V	f	v								
0 1 1 1			'	7	G	W	g	w								
1 0 0 0			(	8	H	X	h	x								
1 0 0 1			)	9	I	Y	i	y								
1 0 1 0			*	:	J	Z	j	z								
1 0 1 1			+	;	K	[	k	{								
1 1 0 0			,	<	L	\	l	|							
1 1 0 1			-	=	M	]	m	}								
1 1 1 0			.	>	N	^	n	~								
1 1 1 1			/	?	O	_	o									

Figure 2 The American Standard Code for Information Interchange (ASCII)

Librarians, with a need to catalogue books in a wide variety of languages, developed a different de facto character set standard in USMARC (Figure 3). This proved very useful within the library community, and had a limited influence on earlier multilingual ISO standards. However, librarians were slow to produce finished standards and they were left behind other developments, both in the computer industry and within standards bodies.

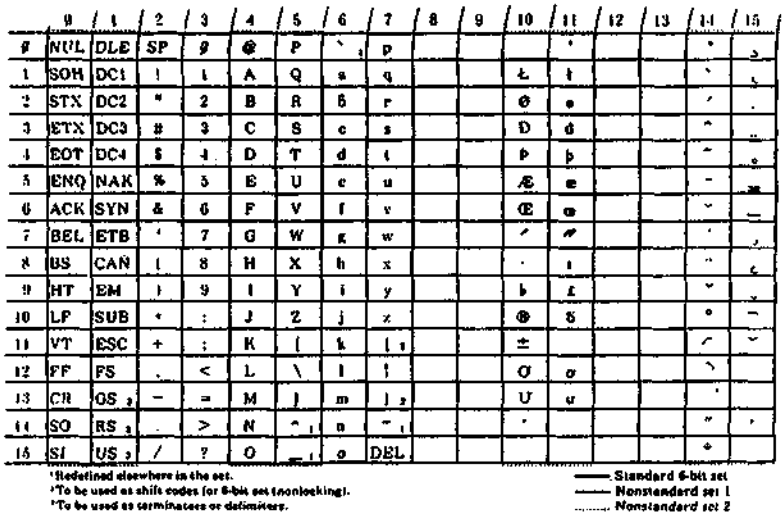


Figure 3 The USMARC character set

In ISO, many of the 8-bit standards being developed were also initially less successful than the IBM PC set, as they provided a smaller range of characters, reserving an additional 32 characters for control characters. Most users in fact preferred to have additional letters, and to achieve control functions though other means.

However, one of its later standards (ISO 8859 part 1, also known as Latin Alphabet no. 1 – shown in Figure 4) was particularly influential, and its repertoire was adopted by IBM (and Microsoft) as code page 850 (Multilingual) – the main Western European code page alternative to the original IBM PC character set (code page 437).

Standards in operating systems

Computer users in the real world don't have standards, they have real computers. Most of us use a clone of an IBM PC, so my following remarks focus on the world of PC-DOS and MS-DOS. Recent developments in MS-DOS form the second area I want to highlight as a landmark for 1991.

MS-DOS started life as an operating system bought-in to meet a contract for getting the first IBM PCs out, and has been the dominant operating system for the IBM PC and all its clones. Its character set, known as the IBM PC character set, and also known as code page

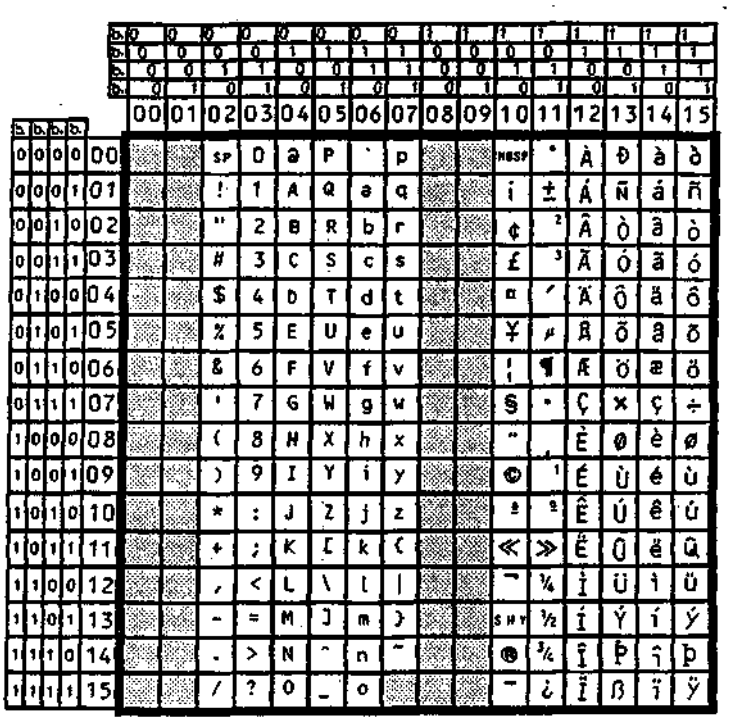


Figure 4 Latin Alphabet no. 1 (ISO 8859 part 1)

437, became a de facto industry standard, although it was never adopted as a national or international standard. Different printer companies adopted completely different character sets, which became yet further industry standards, with the PostScript and Hewlett-Packard printer character sets being the most commonly used one.

MS-DOS version 2.1, released in early 1984, was the first to provide national language versions, although there could be a great deal of incompatibility between different versions if users wanted to exchange textual data between systems.

MS-DOS 3.3 – introduced in May 1987, and now bundled as the standard version provided with many PCs sold worldwide – was the first to provide code page support. Alternative code pages (i.e. character sets), keyboard layouts, and national date, time and currency conventions could be relatively easily changed by users. Most of these were limited to Western European language variants, including Code Page 850 (Multilingual) which provides the full repertoire of ISO 8859 part 1. IBM and/or Microsoft offices in other countries did sell and support other language versions such as Cyrillic, Hebrew and Arabic, particularly from version 4.0 onwards. One annoying feature of MS-DOS was that different language versions were sometimes mutually incompatible. A further annoying feature was that different language versions could not be obtained outside the "market" country, e.g. Greek MS-DOS not available outside Greece.

MS-DOS 5.0, introduced in June this year, is a multilingual landmark in that it already allows several additional code pages as standard, including East European (roman and Cyrillic script) code pages, again corresponding to the repertoires, and in some cases the actual coding, of the appropriate international and national standards. Some anomalies of previous versions have now been removed: for instance, data compiled using Japanese MS-DOS is now reported to be at last compatible with MS-DOS version 5.

Non-roman character set standards

Probably because they were developed later than the 8-bit extended roman character sets, non-roman national and international character set standards were much closer to the internal code pages used by MS-DOS computers, in particular following the Cyrillic, Greek, Hebrew and Arabic parts of ISO 8859. This was because, like ASCII years before, the computer industry made a determined effort to get together with national and international standards bodies to ensure compatibility. ECMA (the European Computer Manufacturers Association) has had a major coordinating influence on 8-bit standard development within ISO and within the industry.

The Greek national standards body, ELOT, developed its 8-bit character set standard (Figure 5) in conjunction with ISO when ISO 6937 and ISO 8859 were first being developed. The more recent national character set standards for Cyrillic (Figure 5) were developed in partial consultation with ECMA which was invited to send a delegation to Moscow as far back as 1988 to make recommendations on character set coding for Russian and extended Cyrillic sets, as well as for Armenian and Georgian.

Hebrew has had a variety of character set standards produced for it, both for character set standards for information interchange and for code pages on PCs. However, although the range of supplementary Hebrew characters, including special punctuation and vowel diacritics, may vary, the basic 22 Hebrew consonants have been identically coded since the Israeli standard SI 960 was produced by overlaying lower case ASCII letters by the 22 Hebrew consonants. Some older Israeli software still uses only upper case English letters and Hebrew characters. Computers handling Hebrew also have to cope with the right-to-left screen and

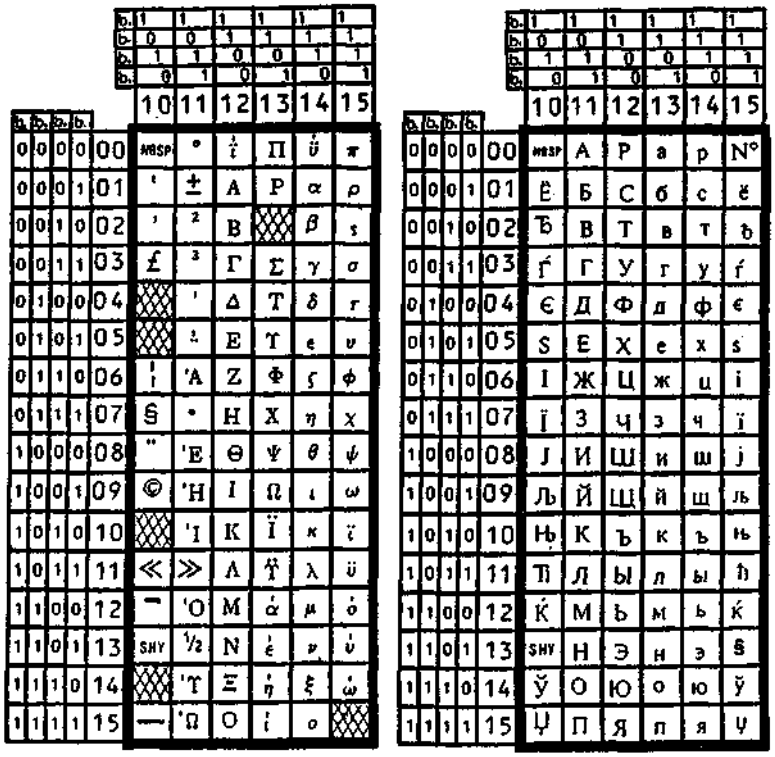


Figure 5 Greek and Cyrillic parts of ISO 8859

printing direction, and be able to mix this satisfactorily with left-to-right English text (see Figure 6).

Arabic character set coding has also been very standardised, following the 1982 standardisation of what was then called CODAR-U FD, and which shortly after became the Pan-Arabic standard ASMO 449. Here again, the code design was achieved by overlaying ASCII letters by Arabic letters (see Figure 6).

The situation is much more anarchic for Indian scripts, as the computer industry, the national standards body took different lines for many years. The Indian ISCII adaptation of American ASCII has gone through three or four unrelated versions, and there has been no attempt to get a comparable standard or even registration at the international level.

Japanese, Chinese and Korean, all of which use a very large repertoire of Chinese characters, need two bytes for representing a large number of characters, although the current national standards of China, Japan and Korea each only use less than 14% of the available 16-bit code space. This inefficient use of code space was one of the reasons for the development of the UNICODE industry standard.

Non-roman script operating systems

Fortunately, the remaining European Greek, Cyrillic, Armenian and Georgian scripts require no more additional processing than does roman script. However, in Asian scripts there are several extra things which operating systems need to provide.

Arabic MS-DOS provides several alternative code pages, including some to allow output in ASMO 449 coding, and others designed to allow either Arabic vowel diacritics or European characters, as well as to work with box drawing characters. Arabic MS-DOS also provides far more sophisticated installation and code page switching than other versions of MS-DOS, and also provides contextual analysis to allow letters to have different shapes depending upon their position within words.

Hebrew needs are similar to those of Arabic, requiring right-to-left display, but not the full contextual analysis required from Arabic operating systems.

In India, a standardised hardware configuration (the GIST terminal) has been used to provide an Indian script operating system. This has to allow for non-spacing vowel diacritics above, below, or around consonants, and also to allow multiple scripts to be handled.

Local operating systems have been developed in other countries too. Although many of them not have been developed by Microsoft or IBM, they will emulate them, with varying degrees of success. CC-DOS for Chinese, and MC-DOS for Mongolian, Cyrillic and Chinese, are examples. These may also be required to cope with vertical writing directions and different forms of the same character too. Chinese, Japanese and Korean operating systems also need to group bytes in pairs to allow a maximum of 256 x 256 characters, although many use a subset of 94 x 94 or less. A few operating systems group bytes in threes to ensure access to the fullest possible range of Chinese, Japanese and Korean characters.

LOCALISATION

During the 1980s, many large computer companies maintained offices in other parts of the world, to localise each new software package, and each new release of it, as it was issued. This led to a wide variation in versions of the software, incompatibilities and increased costs.

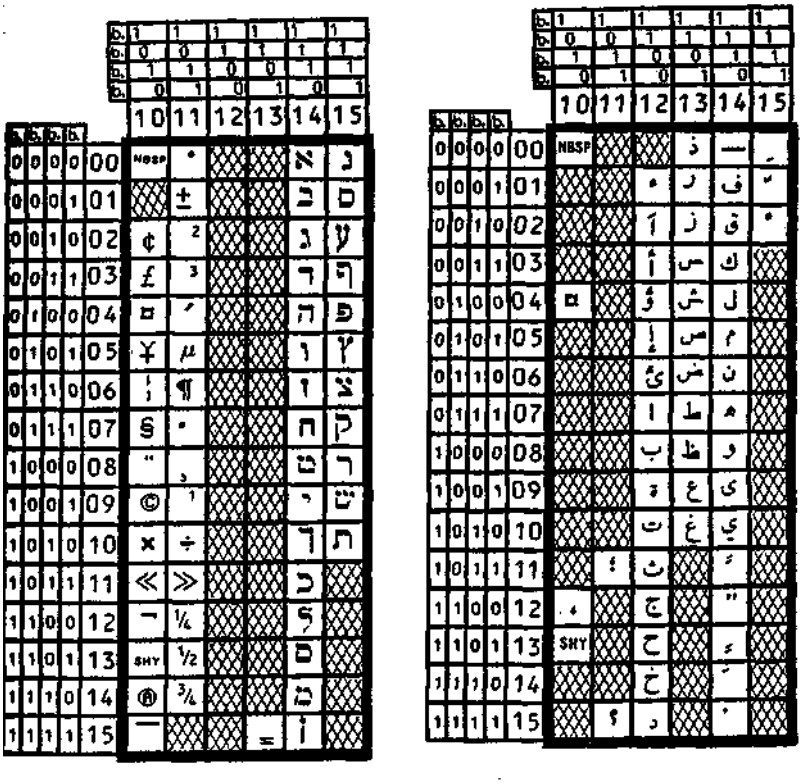


Figure 6 Hebrew and Arabic parts of ISO 8859

If this localisation is built into the software at the operating system level, most application programs (word processing, spreadsheet, database etc.) will run in Arabic, Greek and so on, the underlying character set means that it will work in English (ASCII) and the other language or script. The obvious benefit to computer companies is that there are fewer versions of software to support, and much less staff needed to service them. However, if the application software is ill-behaved – i.e. it does things to the computer besides just stay strictly within the confines of the operating system – there may well be problems. Localisation doesn't solve all the problems.

Globalisation

For the next generation of computers, internationalisation (or globalisation) is replacing localisation, so that only one version of the software needs to be supported, thus further reducing their staff costs. This cannot be achieved by using the current 8-bit character sets (with English and one other script) but only through the use of at least a 16-bit character set.

ISO 10646: the development of the first global character set

The international standards organisation ISO spent some years developing such a character set standard (ISO 10646) which would encode all possible characters with room for further expansion. However, the structure adopted was, in retrospect, both extravagant and archaic, requiring four bytes (32 bits) to represent each character, and did not allow large parts of the code to be used in order to allow maximum compatibility with existing standards. The earliest draft also duplicated many Chinese characters by incorporating the existing Chinese, Japanese and Korean standards as they stood. The attendance of different experts at each working group meeting led to several changes of structure in the draft standard. There were attempts to impose a more straightforward 16-bit structure known as UNICODE onto the standard, but these were rebuffed, in order to assure backward compatibility with older equipment and existing standards.

UNICODE

Frustrated by this, representatives of several major American computer companies based in California, formed the UNICODE consortium. Its aim was to develop this simple multilingual 16 bit code as an industry standard. Its 16 bits only allowed 256 x 256 characters (65,536) with no expansion possible, so all possible Chinese characters could not be represented by this code. Like 7-bit ASCII and the 8-bit IBM PC character set before it, UNICODE was designed to provide a universal way of representing most of the characters required by users now. Led initially by staff from Apple and Xerox, who already had multiple script operating systems available in their computers, the UNICODE Consortium soon attracted other companies into developing and using a competing industry standard. New equipment based on top-end 386 and 486 chips is already being developed by some companies.

ISO 10646(U)

Many computer professionals were concerned by the probability that two standards would be self defeating: some manufacturers would conform to one code and some to the other, and yet others would be tempted to develop their own. In the space of a year, a concerted attempt

has been made to harmonise both sets. The recently issued third draft of ISO 10646 is now very straightforward, with all the strengths of both its predecessors, and none of their weaknesses, and with some additional useful features which appeared in neither before.

The draft international standard ISO 10646 now allows either two-byte or four-byte working, and has UNICODE as its Basic Multilingual Plane (Figure 7) including a "unified" Chinese, Japanese and Korean zone, with no characters duplicated, and with a wide range of alphabets for all other modern scripts, together with additional zones and additional facilities to allow improved interworking with existing computers and networks.

CONCLUSION

In closing, I would like to go over the main issues again. As we have seen, character sets and operating systems are intended to solve problems, but they can raise others. We need to force developers, and to force ourselves, to look at operating systems and software packages from a language viewpoint. There are increasing improvements in software, and some increases in linguistic awareness shown by, for example, developments in MS-DOS. There are also developments like UNICODE just around the comer, where a brighter future is promised – but going back to examples of MPs and new technology – we've been promised brighter futures before! The computer industry can be as fickle as politicians in that respect!

We need information now. I suggest that you make the most of this conference. Look at the exhibition, talk to colleagues, talk to us. The computer industry would be nowhere without all of us together saying what we need.

AUTHOR

John Clews, SESAME Computer Projects, 8 Avenue Road, Harrogate, HG2 7PG, UK

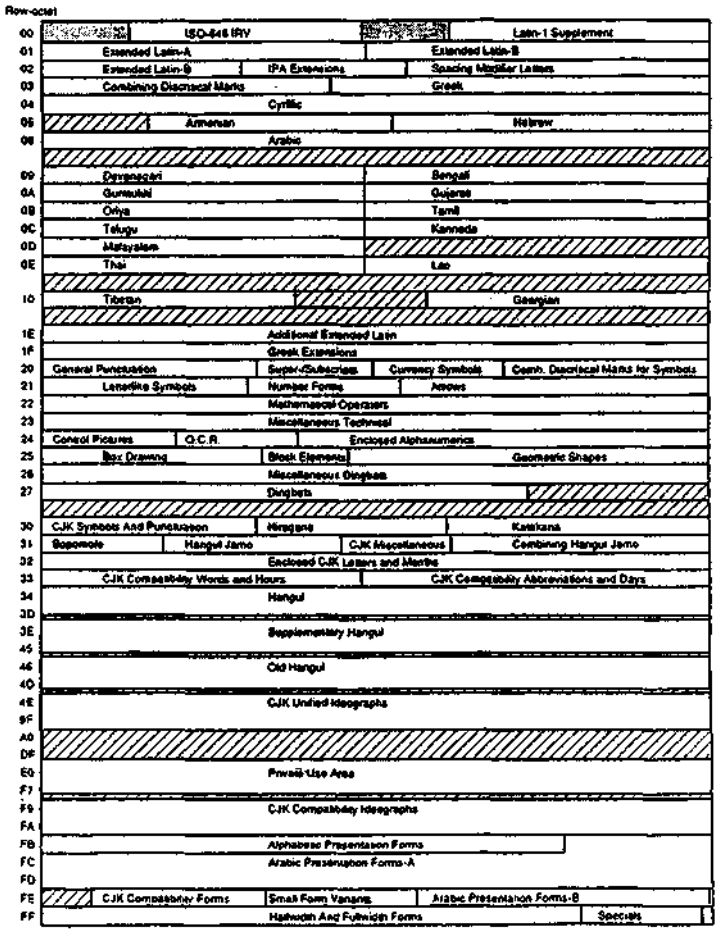


Figure 7 ISO 10646/UNICODE Basic Multilingual Plane