[*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)]

COMPUTER CHARACTER SETS: THEIR EVOLUTION AND IMPACT

John Parry

Europeanization Computer Products

INTRODUCTION

This paper resulted from a joint presentation made by Mr John Clews of the Sesame Project and myself. Mr Clews covered the evolution and standards of the different character sets used on computers in the past and present times. He culminated in a description of the UNICODE character set, which should be the one that is universally adopted. This paper should be read in conjunction with that of Mr Clews.

BASIC CHARACTER SETS

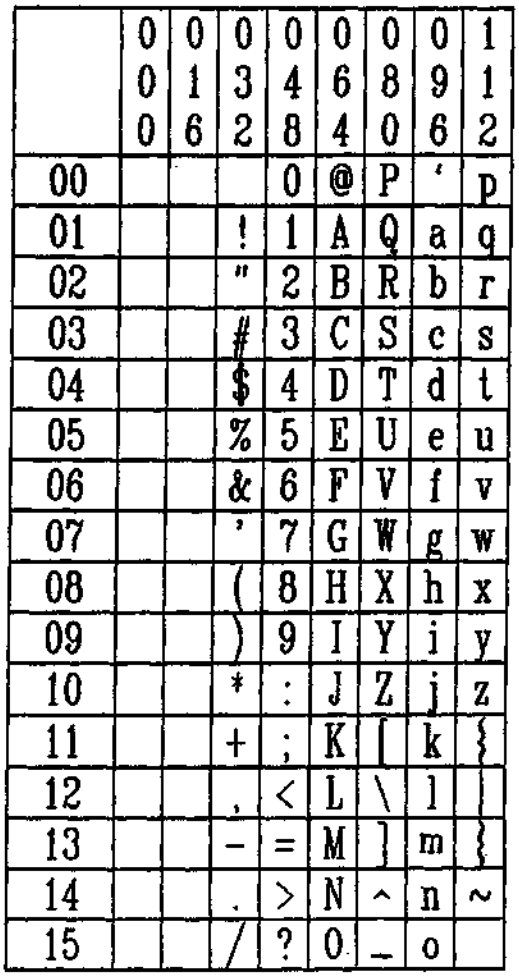
The topic of this paper is the use of character sets by computers. Not just the text itself but what the text means when processing data. The subject is large and can be involved. In this limited space I can only make you think about the implications and, hopefully, inspire you to ask some searching questions of your software vendor. I will cover the basics of character handling and manipulation within computers, i.e. casing, comparing, and sorting.

As Mr Clews outlined, all computer character sets evolved from a simple 7 bit system. Although the basic unit of storage within computers is the 8 bit byte, 1 bit was used to test the integrity of storage. This left 7 bits to represent data. 7 bits allows the representation of 128 different characters so the simple ASCII-7 character set was formed.

Looking at the chart of the ASCII-7 character set (Figure 1) you will see that each character is represented by a number. The standard notation used is to find a letter, get the number from the top of the column in which it is contained and add it to the number of the row in which it is contained, e.g. the character W is represented by the decimal number 87. To uppercase a letter, first obtain its numeric value, then subtract 32. This will give the numeric value of the uppercase letter. Lowercasing is the opposite. 32 is a magic number to computers. It is easily added or subtracted. In early computers characters were compared by their numerical value. So B would come after A, b would come after A, but b would come after C. When comparing strings of characters the same rules were applied comparing characters from left to right.

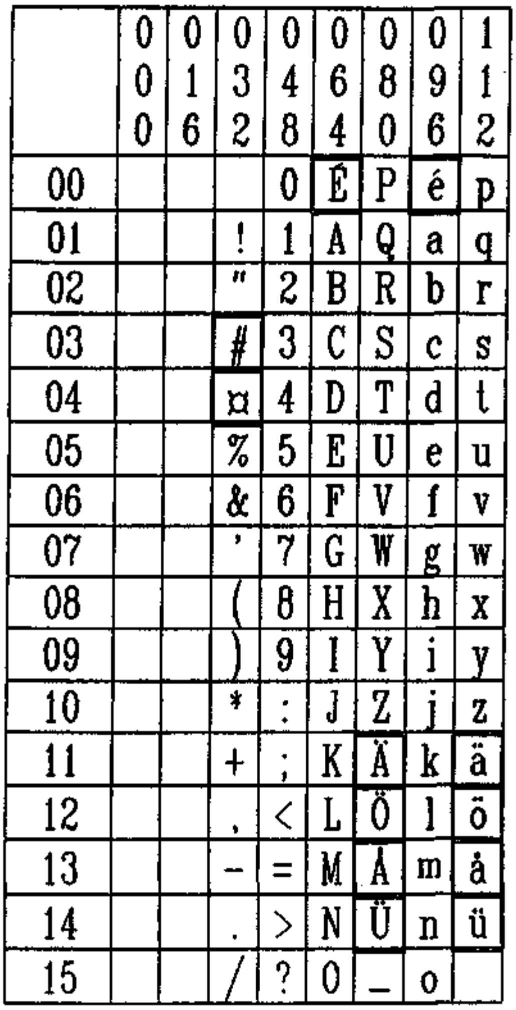
Europe

Although there was only a limited character set there was still a need to use computers in Europe. This problem was solved by substituting the necessary characters for little used symbols. The symbols # $ @ [\ ] ^ ' { | } ~ had substitutions made by the UK, Germany, France, Spain, Italy, Norway, Denmark, Sweden and Finland, Holland, Belgium, Switzerland, and Japan. Figure 2 shows the set with the necessary substitutions to support Sweden and Finland.



	000	016	032	048	064	080	096	112
00				0	@	P	‘	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			$	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			’	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
10			*	:	J	Z	j	z
11			+	;	K	[	k	{
12			,	<	L	\	l	|
13			–	=	M	]	m	}
14			.	>	N	^	n	~
15			/	?	O	_	o	

Figure 1 ASCII-7 character set



	000	016	032	048	064	080	096	112
00				0	É	P	é	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			¤	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
10			*	:	J	Z	j	z
11			+	;	K	Ä	k	ä
12			,	<	L	Ö	l	ö
13			–	=	M	Å	m	å
14			.	>	N	Ü	n	ü
15			/	?	O	_	o	

Figure 2 ASCII-7 character set for Sweden and Finland

When computers were made more reliable there was no need to use the eight bit to ensure integrity. This allowed 256 characters in the character set. Although there are variations of the 8 bit set the most popular and most used is the Latin Alphabet No 1 (Figure 3).

Using the traditional methods of casing, comparison and sorting we have not gained a lot, although most of the Western European characters are represented. The problems are:

this is a set of characters, not a set of letters
this is a set of characters, not an alphabet
this character set does not represent a language.

Possible solutions

Consider casing. Different languages have different casing rules. In French if you uppercase a letter the accent is lost. Of interest is that in French-Canadian the accents are retained. When comparing letters you need to know which letters come before or after others. This weighting of letter will vary for different languages even for the same letters. The rules for sorting or searching through a list of strings of letters (words) will vary for each language. For example, when using a dictionary you are searching and comparing strings of letters based on knowledge and experience. Using an English dictionary you will browse backwards and forwards until you find the word that you are searching for. Applying English knowledge will not necessarily help you find the required word in a German dictionary.

Expansion of characters. Some languages have one character to express two letters. In German the single character Ü represents the letters U and E. When comparing strings containing this character consider the expanded letters to obtain the correct comparison.

Contraction of characters. Conversely, some languages use two characters to represent one letter. For example, in Spanish the single letter ch is represented by the two characters c and h. Again consider contractions when comparing and searching text.

Considering accents. There can be variations in some languages when sorting text containing accents. The rules for comparing strings of letters in French are quite simple. First uppercase the strings and compare the weights of the letter from left to right in turn. If a difference is found then position the strings. Second, compare the strings from left to right ignoring accents. If a difference is found then position the strings. Thirdly, if the strings still appear to be identical, compare them from right to left taking into account the weights of the letters with their accents. If a difference is found then position the strings. The following four words are to be compared:

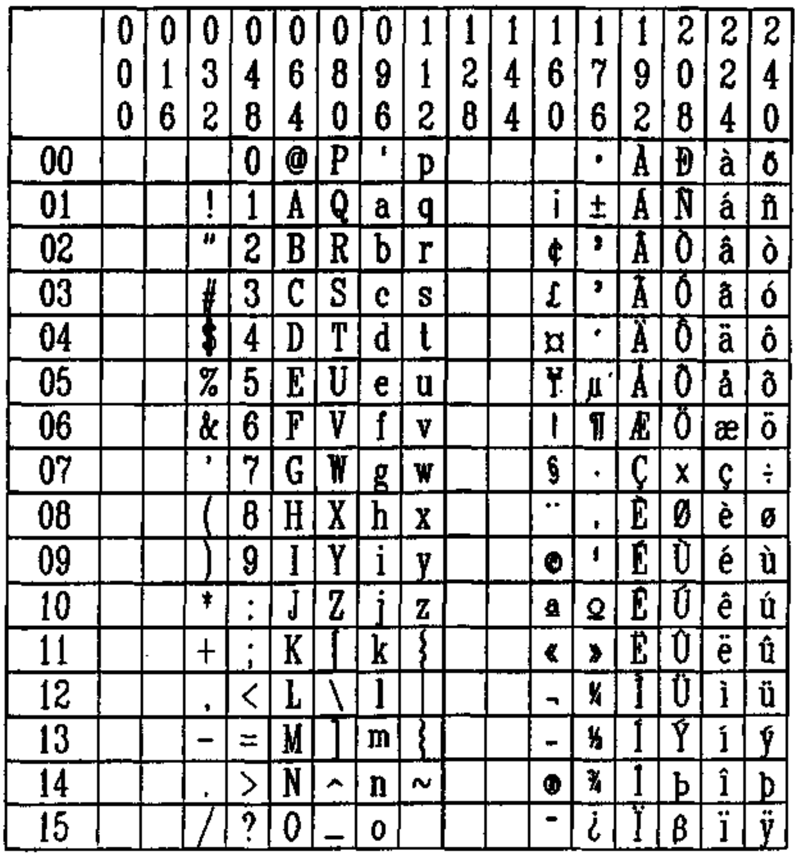
share	cote
coast	côte
aspect	côté
classed	coté

Using English sorting rules the following sequencing would be achieved:

cote
coté
côte
côté

But using French comparison rules they would be sequenced:

cote
côte
coté
côté



	000	016	032	048	064	080	096	112	128	144	160	176	192	208	224	240
00				0	@	P	'	p				°	À	Ð	à	ð
01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
06			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
07			'	7	G	W	g	w			§	·	Ç	×	ç	÷
08			(	8	H	X	h	x			¨	¸	È	Ø	è	ø
09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
10			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
11			+	;	K	[	k	{			«	»	Ë	Û	ë	û
12			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü
13			-	=	M	]	m	}			-	½	Í	Ý	í	ý
14			.	>	N	^	n	~			®	¾	Î	Þ	î	þ
15			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

Figure 3 ASCII-8 character set

FULLER CHARACTER SETS

The second part of this paper covers, briefly, what you could do after you have decided to convert to a fuller character set. These are not instructions as to how you should convert your system, but rather an indication of the type of questions that you should ask your software vendor.

Localised software

Localised software allows you to use a particular product according to the national conventions and common language usage associated with a particular local area. This means, for example, that a product localised for Greece would allow you to use a Greek character set, express dates and times in Greek, and also compare and sort text according to the Greek alphabet.

You should ask whether the product is a localised version or whether it is truly international but is tailored to the Greek market. An international product will allow you to use any character set but in a local manner. To take the Greek example further. A Greek version of an international product will allow you to produce text in any language but will have its dates and times set for the Greek local variant.

Before committing yourself to a localised product you should consider whether it will affect your working practice. If you or your staff are used to having data presented in an English format will the localised version cause you problems? If so ask your vendor how the product can be customised to suit you and how it can be changed gradually to a local version to suit staff requirements and training.

Mixing software

Consider how installing a localised version of a product could affect other pieces of software on your system. Having a localised version of your spreadsheet may be very useful but if it does not match your database product then any benefit could be lost or serious misinterpretation of data could occur.

Conversion of a complete system

If you have decided to convert your whole system to use a common character set along with its associated conventions then be careful. The conversion is not simply a case of convening an occurrence of a character to another character. This could be true of text and raw data.

What about program? Programs are stored as binary codes that may appear as characters. These should not be converted. But with programs there will be literal strings of text and these will need to be converted. With knowledge of the structure of the programs this is possible.

Finally, there are the databases themselves. The databases will have indexes that were set up according to a particular collating sequence. If the conversion of your system means that the collating sequences will change then the indexes will need to be rebuilt. Most databases provide a tool for unloading the data, then reloading it, such that the indexes will be remade, but check that yours can do it with an alternative collating sequence.

You should be able to have a system converted. If a vendor says that they can convert 95% of your data, then press them about the other 5%. It could be the area that contains your tax records.

Display type peripherals

Before you finally commit to conversion you should consider how you are going to display the new data. Will your existing printers and VDUs be able to display the new character set? If not, can you make them do so with interfacing software?

Before you converted you knew what the limitations were. Now you and your staff will think that what they see what is there, not that some data cannot be displayed because of a limitation with the equipment.

Remote access vs Remote process

Now that you have your converted system you will wish to access other similarly minded systems. Beware. What you see is not necessarily what is there!

If you use your English system to access remote German data then the results will be returned to you according to English searching rules. This is remote access and cannot be relied upon to return accurate data. But if your English system asks a German system to retrieve and return some data to it then the results can be relied on because they have been retrieved according to the rules that stored them. This is remote process.

Do we need to convert?

To live in an isolated world with specific character sets will leave us behind. The move towards UNICODE is a great step. To be able to incorporate the knowledge of languages will be another great step but one that I feel may not be that close.

Something that may help or hinder us is Open Systems Interconnection (OSI) or more commonly Open Systems. Open Systems is a fast growing initiative that aims to make the interconnection of computers easier. Open systems is defining methods for computers to communicate irrespective of the manufacture, operating systems, software, or even character set! Like the definition and acceptance of a common character set, Open Systems is taking some time to make itself felt. But just like the common character set it is not that far off.

AUTHOR

John Parry, Europeanization Computer Products Ltd, The Woad, Sherington Road, Newport Pagnell MK16 8NL, UK