SEAS

SHARE European Association
Incorporated in The Netherlands
Founded in 1961

SEAS
National Character
Task Force

White Paper on national character, language and keyboard problems, September 1985

Copies of this White Paper may be obtained from

SEAS Headquarters
University of Nijmegen
Toernooiveld
6525 ED Nijmegen
The Netherlands

Phone +31 80 558833 ext. 3357

Enquiries and suggestions regarding the content of the White Paper should be directed to the above address and marked

'Attention SEAS Office Automation Project'

SEAS
National Character
Task Force

White Paper on national character, language and keyboard problems, September 1985

Abstract

The SHARE European Association National Language Task Force (SEAS NCTF) was inaugurated in 1982 to investigate and report on the problems encountered using IBM software and hardware in non-English-speaking countries with a written language based on the Latin alphabet. This White Paper is the result of NCTF work in the period 1982-85 and covers the areas of traditional dataprocessing, word processing and personal computing.

Introductory sections deal with the history of the problem area, followed by sections describing the actual problems and their required solutions as regards 'national characters', 'national languages' and 'keyboards'. The final section outlines the general characteristics of future implementations of software and hardware which are required by the SEAS user community. These implementations are expected to give equal value for money for both English-speaking and non-English-speaking IBM customers, together with significantly improved product performance, quality and ease of use.

Current Members of the Task Force

Anders Berglund, CERN, Switzerland
Bernard Chombart, Promodès, France
Günther Krysmanski, Germany
Jerry Andersen, IBM Kingston, USA
Klaus Daube, Oerlikon Bührle RZ AG, Switzerland
Peter Gardner, Kommunedata, Denmark (chairman)

Earlier Members of the Task Force

Hubert Ickes, IBM Kingston, USA
Knud Nielsen, Denmark (chairman)
Michael Täschner, Germany

Management Summary

European users of IBM hardware and software products find increasingly that these products do not meet their requirements in the areas of:

- National character support
- National language support
- Keyboards
- Inter-product compatibility

IBM hardware and software products are primarily designed for the English language and the 26 letter latin alphabet. When installed in non-English-speaking countries most IBM products suffer a degradation in function and/or user-friendlyness and/or performance. The user often has to be satisfied with some sort of compromise or has to put up with continual irritation and inconvenience. In other words, European users experience poorer product quality and get less value for their money.

The White Paper describes in detail the problems encountered in Europe and asks IBM to give top priority to the solving of these problems on an integrated and comprehensive (architectural) basis.

The White Paper has been designed as a source document for the IBM product organization at all levels.

The contents will also be of interest to the various national and international standards organizations since they reflect the current state of affairs in the largest single computer user community in Europe.

Table of Contents

1 Introduction

This White Paper has been prepared by a 'National Character Task Force' under the auspices of the SHARE European Association, SEAS. The White Paper addresses the problems experienced by non-English-speaking countries when using the hardware and software products currently available from IBM.

These products are primarily designed for the English language and the 26-letter Latin alphabet. When installed in non-English-speaking countries most IBM products suffer a degradation in function and/or user friendliness and/or performance. The user often has to be satisfied with some sort of compromise or has to put up with continual irritation and inconvenience.

SEAS believes that the time has come for IBM to provide the non-English-speaking user with systems of the same quality as is attainable in the English-speaking user community.

Section 2 of the White Paper describes the history of the problem area. It explains why the situation is becoming critical now, despite the fact that most of the problems to be described have existed for at least the past 10 to 15 years.

The prime objective of the White Paper is to be found in section 3. This section provides IBM with a basic source document describing problems as experienced by users and stating the required solutions. There is no attempt to suggest details of implementation, that is to say details of how the solutions should be achieved. However the general characteristics of the implementation as a whole are of great importance to the user community. Section 4 sets out SEAS' requirements for the implementation.

1.1 History of the National Character Task Force.

The events which led to the establishment of the Task Force had their origins in a paper given by SEAS Executive Board member Michael Martin at the 1980 Spring Meeting. The paper was entitled 'User Experiences with Communications' and in it Michael Martin outlined many of the problems facing a typical Danish installation using IBM hardware and software products. By far the majority of problems were caused by product design which made it difficult, if not impossible, to use (or 'implement') the Danish alphabet and the Danish language.

The problem area was also discussed in presentations at the SEAS meeeting in Antwerpen in 1981 given by B. Mölenhof (W.Germany), B. Chombart (France) and by IBM representatives Loizides and Schwinge.

Michael Martin's paper merely brought together a number of problem areas which had been plaguing the non-English-speaking user community for many years. The problems described were to be found to a greater or lesser extent in all European countries.

The points raised in this paper were incorporated into a resolution from the SEAS OS project (088-0680, 03.06.80) which received a sympathetic response from IBM - but a response which did not imply that IBM were really aware of the seriousness of the situation in Europe. Since Martins' paper had stated that European installations got in effect less value for money than their American counterparts, the SEAS board decided to raise the national character problem as one of several 'major areas of concern' at their next regular meeting with the management of IBM Europe.

Executive Board member Frédéric Roux (France) presented the problems on January 8th 1982 and proposed the formation of a task force consisting of 3 SEAS members from different European countries plus an IBM specialist.

IBM agreed to participate in a task force and the inaugural meeting of the SEAS 'National Character Task Force' took place in Noordwijkerhout in the Netherlands on May 4th 1982. The Task Force at that time consisted of SEAS members

Knud Nielsen	(Denmark, Chairman)
Bernard Chombart	(France)
Michael Täschner	(West Germany)
and IBM specialist	
Hubert Ickes	(Kingston, USA)

The Task Force held a 2 day meeting in Paris on August 9-11 1982, by which time Hubert Ickes had been replaced by Jerry Andersen (also from IBM Kingston) and Michael Täschner by Günther Krysmanski (also from West Germany). The contents of the White Paper were agreed on and draft versions for several sections were produced shortly afterwards.

The departure of Knud Nielsen from SEAS resulted in no new work being done until early 1984. The Task Force, now under the chairmanship of Peter Gardner (Denmark), held a 2 day meeting in Paris on 25-26 January 1984. Further work was done here on the structure of the White Paper but again there were difficulties of a practical nature - perhaps understandable given the 15-year history of the problem and its all-embracing aspects.

The next series of working meetings was held in the Netherlands in Veldhoven, in conjunction with the SEAS Spring Meeting 15-19 April 1985. The task force had by this time acquired two new members, Klaus Daube and Anders Berglund, both from Switzerland.

At Veldhoven it was agreed that the subject matter be 'frozen' so that a final version of the White Paper could be presented at the SEAS Anniversary Meeting in Zürich in September 1985.

1.2 Objectives.

The objectives of this White Paper are twofold:

Firstly to make IBM aware of the magnitude of the problems encountered in non-English-speaking countries (section 3).

Second to make IBM aware of the need for solutions at the system architecture level together with the need for stepwise, user-controllable and low-cost conversion procedures when the time comes to implement these solutions (section 4).

1.3 Scope.

The Task Force has found it necessary to limit the scope of section 3 in two ways:

The problem descriptions will be limited to problems arising in countries using the Latin alphabet as the basis for their written language. This means that additional problems arising in, for example, Arabic, Cyrillic, Greek, Hebrew, Kanji and Katakana user communities will not be described.

The problem descriptions will be further limited to those arising through the use of locally standardised character sets where the attributes of a character are limited to its graphic value, the code point in the EBCDIC table and, for alphabetic characters, the case (upper or lower). We are therefore addressing problems in the traditional Data Processing (DP) and Word Processing (WP) environments together with problems in the new Personal Computer (PC) environment.

Both limitations are felt to be reasonable at the time of writing and have been necessary in order to keep the work of the Task Force to a manageable level.

The first limitation illustrates the essentially global nature of the problem area and requires further comment. We have no illustrations in section 3 of problems specific to, for example, the Greek user community - unfortunately, since Greece is in Europe. We do, however, require the use of the Greek alphabet everywhere in Europe - for both scientific and commercial work.

Similarly we can state that a solution which does not allow use of the Cyrillic, Greek etc. alphabet will increasingly come to be seen as a hindrance. The same arguments obviously apply to Arabic and Japanese. These aspects are not taken up specifically in section 3, but they resurface in section 4.

The second limitation which is purely technical in nature puts additional problems

which may occur in the fields of, for example, computer graphics, publishing and image processing, outside the scope of the White Paper. It is hoped that IBM will by inference recognize the same (or new) types of problem in these fields and take action accordingly.

The scope of section 3 is thus a 'level of ambition' for the problem descriptions in this section. In contrast, the scope of section 4 has wider boundaries. Here the Task Force feels able to outline a set of requirements which are generally applicable to any geographical or technical area.

Whilst international standards for coded character sets are refered to in several places in section 3 it should be realised that there is no organized attempt in this White Paper to relate problem descriptions or required solutions to existing international or national standards. We are in effect criticising IBM for its lack of standards rather than for its deviation from existing standards which may in themselves be restricted or of dubious usefulness in regard to the problem areas described. (Witness for example the many registered national character sets where important delimiter characters such as { } [| / \ are replaced by national use characters...)

Finally on the subject of scope we would note that it has been necessary to 'freeze' the subject matter at a certain point in time in order to concentrate on publishing the White Paper. This document should thus be assessed as a reasonably well-formulated attempt at presenting a very complex problem area, but **not** as an exhaustive thesis.

1.4 Intended Audience.

The intended audience for this White Paper is primarily the IBM organization at all levels, both inside the USA and in Europe.

We also hope that the White Paper will be an eye-opener for the many DP-professionals who have 'learned to live' with character, language and keyboard problems. Now is the time to react - to bring all the old problems up to the surface and to work for a common solution.

The White Paper may also be of more than passing interest to the various standards organizations. The problems described reflect a sad state of affairs after many years of standardisation effort. We are aware that these conditions are by no means confined to the IBM product world, though we will not attempt comparisons with other suppliers here. What must not be overlooked is the commanding position IBM holds in both hardware and software markets. A standard which is unacceptable to IBM or only partly acceptable (for whatever reasons - technical, financial or strategic) has little chance of wide acceptance. As a result, IBM customers are often unable to reap the benefits of existing standards, whilst the standards organizations must see their work neglected in practice by the largest user community. This state of affairs

has certainly dominated in the past. The NCTF hopes that both sides will be able to cooperate more productively in the future .

As mentioned in the Introduction, most of the problems described in section 3 have been with us for a long time and it is hard to see why so little has been done to solve them over the years. In fact the problems have had a tendency to multiply rather than disappear. We can only assume that both IBM and its European user community have had a blind spot here - or have been harassed by other problems of a graver nature.

Whilst it is fairly easy to understand the conservative attitude of IBM product development organizations in the USA with respect to national language and character problems, it is difficult to understand the inaction or lack of leverage of the IBM organizations in each European country. These organizations will, between them, recognise each and every problem in section 3. Most of them will have had first hand experience of these problems in their service bureau and consulting work.

2 Changes in Computing Environment

The Task Force believes that an understanding of the changing information processing environment is essential to put coded character set requirements and problems in perspective. The following discussion of the changing environment is offered in that spirit.

2.1 History

Both the cost and size of computers have decreased by several orders of magnitude over the past quarter century. These reductions have lead to equally dramatic changes in the types of applications processed on computers, and in the nature of the end-users of those computers. The result has been a significant evolution of the coded character set requirements levied against hardware and software products.

Twenty five years ago, the cost of a computer was high enough to be prohibitive for all but the largest businesses and government agencies. Applications were limited to those that could be justified on such an expensive tool. Payroll and accounting were typical, as were various scientific programs that performed particularly complex and lengthy calculations. Computer users were data processing professionals who accepted that communication with a computer had to be on its terms and in its language.

Input was usually from punched cards or paper tape, and data was output on a line printer or card punch. Character sets typically contained 30 or 40 characters, and 5 and 6-bit codes were common. European requirements were satisfied by setting aside a few 'national use' code points, so that graphic characters could be assigned based on each country's need.

As requirements grew, and technology improved, character sets stabilized at 94 characters. IBM standardized on the 8-bit-Extended Binary Coded Decimal Code (EBCDIC), with increasing European requirements being met through expansion of the 'National Use' code-points to the current 13.

The environment described above continued to change markedly through the ensuing years. Input from remote terminals became virtually universal. Keyboards grew in size, and techniques were adopted that allowed 3 or more different graphic characters to be selected from a single key. Terminals with the ability to handle larger, and multiple, character sets became common. Matrix printers also overcame character set size limitations, and new character codes were adopted to meet various application needs.

As a direct result of the dramatic reduction in price, computers have become

relatively common in small businesses and even private homes, with today's end user as likely to be a teller, secretary, student, or accountant as he or she is to be a DP professional. While computers still process payrolls, they are increasingly being used to draft letters, play games, modify recipes, and prepare income tax returns.

This dramatic evolution in the information processing environment has continued to put steady pressure on coded character set support. While character sets are significantly larger, the expansion has fallen short of the demand. European customers are increasingly requesting that products allow them to use all the legitimate letters of their native language. Text and Word Processing customers find that they need additional characters on their keyboards and in their character sets in order to correctly spell various words and names. It is clear that the 94 character set can no longer meet the requirements of many countries whose languages require large repertoires of accented letters.

In addition to problems of character set size, multinational and multilingual requirements are increasing. Multi-national companies are finding that they have an increasing requirement to be able to correctly transmit data across national borders. Customers in multi-lingual countries, such as Belgium and Switzerland, are increasingly insistent about the need to supply coded character set support that spans the requirements of two or more languages.

These problems are aggravated by difficulties created by differences between various coded character sets, as well as software design that incorrectly treats the differences between coded character sets of the various countries. Users of products spanning two subsystems are often frustrated to find that data entered from one product cannot be retrieved by another. Others find that characters that they require are effectively unusable with some programs, even though available at the terminal.

In sum, the coded character set support that was designed to meet the requirements of 10 and 15 years ago is now needlessly diverse, as well as increasingly non-responsive to modern needs. Coded character support needs to be updated to bring it into line with today's end-users and their needs.

2.2 Future Evolution

The trends observed in section 2.1 can be expected to continue in the forseeable future. As the price of computers continues to drop, small business and personal computers will continue to proliferate. New application areas will be discovered, and casual and non-professional end-users will become an even more important factor. Ease-of-use will be stressed, with subsequent pressure on coded character set enhancements. Two trends in particular appear especially significant in regard to character set support.

First, the Task Force believes that the future will see continued growth in the area of Text/Word Processing. This is important because it underscores the requirement for enhancement of European character sets so as to include all of the characters required for each language. Combined with increasing requirements for international communication of data, the net result is continued pressure to supply character sets that not only meet the requirements of one country, but of all countries sharing a common basic alphabet.

Second, the Task Force believes that product distinctions between data processing and word processing are artificial and misguided. The trend is expected to be in the direction of merging these two areas into one 'information processing' application area. Terminals, printers, and software may then be anticipated to interwork across both DP and WP, resulting in the requirement for one work-station on a user's desk, not two. Although the trend can be verified in many newer products, coded character set support continues to be divided into three families, one for DP, one for WP and a third for PC. We believe this hinders and delays the inevitable merger.

Finally on the subject of future evolution, we would like to emphasize that solutions are already long overdue. The piecemeal solutions currently available compound the entire problem area. New ad-hoc solutions in specific areas are constantly being devised at installation, company and national levels - resulting in an overall situation which is difficult if not impossible to describe, let alone understand. Section 4 emphasises the need for a speedy statement of intent from IBM which will point the way to long-term solutions and allow best possible construction of interim solutions.

The user community is also increasingly aware of the fact that other suppliers are already in the market with operational solutions to many of the problems which exist in the IBM-world.

3 Problem descriptions

This section is divided into three major parts:

Section 3.1 describes hardware problems encountered when trying to enter or reproduce national use characters, together with other hardware problems encountered through usage of national characters. These problems are called '**national character problems**'.

Section 3.2 describes software problems encountered either because a national character set is in use or because of the requirement that software products must be able to 'converse' in the language or languages of the countries in which they are executing. These problems are called **'national language problems'**.

Section 3.3 is devoted to the specific problem area of **keyboards**. The keyboard of a terminal is by now the primary input device in most installations. It is the primary man-machine-interface in all installations and it is when using the keyboard that many of the problems described in sections 3.1 and 3.2 first become apparent.

Some problems may be on the border-line of these categories - or may be a mixture of them. In these cases the seemingly best category has been chosen and this rather loose approach underlines the basic policy of section 3. Our objective is description rather than analysis. The technically expert reader will quickly recognize that several problems have the same fundamental cause, but he should be on his guard in any critisism of the White Paper for banality, naivety or lack of technical insight in this respect.

By describing what the IBM customer in Europe identifies as **his** problems we are simply passing on 'the facts as they are' to the IBM technical organization. It is up to this organization to perform the analysis and formulate the necessary strategies for solving these problems within the constraints described in section 4.

We feel obliged to state at this point that many European installations (often members of SEAS) possess experts in the problem areas described. These people have a knowledge level at least as high as, and in some cases higher than, the IBM organizations in their respective countries. It is therefore not through lack of expertise that the problem descriptions in this section have been kept at a fairly simple technical level - but as policy.

We have attempted to adhere to the following pattern for each description:

- A non-commital title
- The problem description
- Examples where appropriate
- The required solution.

We have also felt obliged to grade the problems roughly according to the scale:

A Result incorrect or not as intended,

B Impossible to do but the intent is obvious and reasonable,

C Impossible to do but would like to be able to do,

since the problems described range from direct errors to facilities which are at present missing but are urgently required.

3.1 National character problems

This section describes problems in connection with hardware, chiefly manual input devices and screen and print output devices. Problems resulting from the diversity of character sets have also been placed here, since the character set is essentially a function of the hardware capabilities of the available input devices. For example the de-facto character set of a typical commercial (DP) installation is identical with the character set installed in its 3270 network.

3.1.1 Dual 3270 character sets (grade A)

Several countries have had their 3270 national character sets split into two almost identical versions (the 'almost' is the problem). The 'old' version dating from the first introduction of the 3270 hardware is know known as the 'alternate' character set. The 'new' version dates from the introduction of the 3278. Some of the countries affected and the graphics/code points changed are shown in the following small section from GA27-2837 (figure 10-32).

National use number I/O Hex Code Controller Language Device	1 4A	2 5A	3 6A	4 79	5 5B	6 7B	7 7C	8 5F	9 A1	10 C0	11 D0	12 E0	13 4F	14 7F
Austrian/German	Ä	Ü	ö	`	$	#	§	^	ß	ä	ü	Ö	!	"
Austrian/German (alternate)	ö	ü	ß		Ü	Ä	Ö	¬					|	ä
Danish/Norwegian	#	¤	ø	`	Å	Æ	Ø	^	ü	æ	å	\	!	"
Danish/Norwegian (alternate)	â	å	|		Å	Æ	Ø	¬					|	æ
Finnish/Swedish	§	¤	ö	é	Å	Ä	Ö	^	ü	ä	å	É	!	"
Finnish/Swedish (alternate)	ö	å	|		Å	Ä	Ö	¬					|	ä

The 'new' character set was developed/introduced without any form of customer or user-group consultation. The result is that old users (3277 converted to 3278) have had a tendency to stay with the alternate character set - the new version having no obvious advantages of any kind, and in some cases several disadvantages. Note here

that there was already a very widespread use of 3277's at the time the 3278 was introduced. Customers (and their customers in in turn) had no incentive to replace the terminal equipment in nets with several hundred terminals, often purchased and not rented. Over the years the 3277 has of course been phased out - but the alternate character set is still with us.

New customers (post 3277) have acquired the new character set, in most cases without having been informed of the alternative.

IBM has been very quiet on the whole subject of the change, and there is no conversion/comparison documentation available to our knowledge.

The differences between the old and new character sets are essentially:

- the re-positioning of some or all national characters in the code table.
- the disappearence of 'vertical bar' and 'not sign' which are replaced by 'exclamation point' and 'circumflex accent'.

Note that both special symbols are heavily used in PL/I, a popular programming language in Europe. Also the vertical bar is an indispensable component for producing flow-charts, block-diagrams and low-resolution graphics in general.

To add to the confusion, the 3278 keyboard is only available with key-tops corresponding to the new character set. This means that 'exclamation point' must be hit for 'vertical bar' etc. - a constant source of irritation especially for end-users. It also means that some of the keys have no effect when hit - more irritation.

There are other peculiarities. For example in Denmark the u-umlaut from neighbouring Germany was a new graphic in the new character set. Danish users have always required this character and in fact it had been available on '3277-compatible' equipment for some years in both upper and lower case versions. The IBM offer is a single case u-umlaut (unknown whether upper or lower case) in a code point different from either of the two which were already widely in use, and incidently also different the German/Austrian code points for u-umlaut in both the 'alternate' and new versions for these countries (see table above).

The existence of dual 3270 character sets in some countries further complicates the entire national character problem area, causing confusion for all parties involved - not least IBM themselves. The ultimate in confusion arises when 'alternate' and 'new' TP-networks are tied together in an SNA/MSNF configuration.

The requirement is that this sort of unilateral uncoordinated change never be repeated in the future.

3.1.2 Character sets are incomplete (grade B)

Multinational companies and agencies find themselves unable to enter and display correspondence which is written in different languages, in the DP environment.

To take a simple example, it is only in the UK and Italy that standard DP equipment will be able to reproduce the sentence

The exchange rate is £1 = $0.75

Only these two countries have the £ symbol and several countries do not even have the $ (and incidently the assignment of these two graphics to code points in the UK and Italy is such that there is no possibility of the sentence being correctly reproduced after transmission from one country to the other).

The requirement is that each country have access to a unique and complete code table and that all codes be enterable and displayable.

3.1.3 Diacritical marks in latin alphabet based languages (grade B)

Currently IBM products are only available with the national use characters of selected European countries. Teletext (the successor of Telex), however, supports all the European languages with their special symbols and diacritical marks. With the increased linking of computers for electronic mail and linking of computers to the Tele(te)x network the lack of support from IBM for many European languages is seen as an upcoming severe problem area.

It is impossible to translate a text from Videotex or Teletext correctly to EBCDIC. Also with the multilingual codepage there are missing accented characters.

Turkey belongs to NATO and is strongly related to the European Common Market. So it is necessary to write Turkish in Europe also on DP equipment. Also writing Polish is necessary for international institutions and organizations. The multilingual codepage does not support all the needed characters - and with the concept of "fully formed" characters in one codepage it is not possible to define all of the accented characters which are needed.

The Turkish alphabet uses the special characters

Ç ç Ğ ğ Ş ş Ö ö Ü ü İ ı

The accents are not only for correct spelling and pronounciation, they also can change the meaning or sense of a word completely!

Olumlu bir haber	a good message
ölümlü bir haber	a mortal message

Havalar ılık geçecek	the weather will stay warm
havalar ilik geçecek	the weather will go through a button hole
çam yeşildir	the firtree is green
cam yeşildir	the glass is green.

There are the following special letters in the Polish alphabet. Not all of them can be printed or displayed on IBM equipment:

ą Ć ć ę Ł ł ń Ó ó Ś ś Ź ź Ż ż

Ona jest mężatką	She is married
Łódź jest przemysłowym miastem	Lodz is an industrial city.
To jest żmudna praca!	What a troublesome work!

Also the Czech, Slovenic and Serbocroatic languages use additional charcters in their alphabets:

č Č č Č ď Ď ň Ň ř Ř š Š ť Ť ž Ž

The French language uses some accented letters, which are especially vowels, plus the c with cedilla:

à â ç é è ê ë î ï ô ù û ü

For instance the French sentence "Les bûches brûlent tête-bêche dans l'âtre de la cheminée: c'est Noël" is not a strange sentence, but a very typical case.

A concept of flying accents would save many codepoints. It also would allow the writing of many latin transliterations of non-latin-based scripts. Since there are output capabilities for these things (see this paper), an adequate solution for storing and input of these texts is needed.

3.1.4 Inconsistent code points across DP/WP/PC (grade A)

It is not possible to transfer documents from a WP-environment (e.g. IBM 5520) to a DP-environment without loosing information.

In the WP-environment there are a lot of code points for graphics which have no corresponding graphic in DP-environment. For example there is no µ in the DP-environment. National characters which are available in both environments are coded differently. For example the ü has code point X'D0' in DP-environment in

Germany but X'DC' in WP-environment.

Similar problems also occur when transfering out of the PC-environment.

The requirement is that no such diversity of code tables exist.

3.1.5 Codepoints for same graphic differ internationally (grade A)

If letters are sent from one country to another, lets say from Germany to Denmark, they may be written in a language not spoken in either of these countries (e.g. English). For the text of the letter there will be no problem. To sign the document the national characters of the sending country are needed and for addressing those of the receiving country are needed.

In the case of one of the members of the SEAS National Character Task Force, Günther, problems arise. ü is a German national character, but it is also a Danish 'neighbouring country' character. In Denmark this name appears on the screen or the printer as Gånther.

ü is coded as X'A1' in Denmark and as X'D0' in Germany. X'D0' is displayed as å in Denmark.

It is not a good solution to translate the code points at the borders of the countries because there are files, which may contain both text and non-text information. There is no way to decide which part of the file has to be translated and which part has to be left untranslated.

The requirement is that the same code points be used for the same graphics in all countries.

3.1.6 Graphic values for same code point differ internationally (grade A)

This is the reverse situation to the one mentioned before. A letter sent from France to Germany by electronic mail or tape cannot be read, because the French national characters are displayed as German national characters.

The è is coded in France as X'D0'. This code point is displayed on a German 3278 terminal as an ü. Notice that ü both is a German and a French national character.

The requirement is that the same as for 3.1.5

3.1.7 Hardware limitations (grade A)

The requirement to be able to use a specific national character set may impose hardware limitations on the customer which seriously reduce the value for money invested in a piece of equipment.

For example users of a so called 'alternate' character set (see 3.1.1) will find that IBM 3274 cluster controllers cannot be configured for 7 colours, graphics or the APL character set. This means that it is necessary to acquire extra controllers if any of these facilities are to be used.

It is also impossible to install different "languages" (keyboard support for different languages) in one 3274 controller.

An IBM 7436 letter-quality printer configured to use an alternate character set cannot print the vertical bar '|' which is included in the character set. It prints as a minus '-' which makes nonsense of any kind of block chart or form.

The IBM 6670 laser printer originally allowed only 2 concurrent fonts in Denmark, whilst American users were allowed 4 concurrent fonts.

The IBM Quietwriter printer as supplied is unable to reproduce the Danish/Norwegian Ø ø. This is probably a side effect of the fact that these two graphics are (incredibly!) missing from the IBM PC character set. (They can be implemented on the PC by installing IBM's 'National Supplement, Scandinavia' which in turn causes problemens for third party PC-software.)

Obviously the requirement is that no such limitations exist. They are completely artificial and stem not from technical problems but from either IBM policy or lack of knowledge of the European market.

3.1.8 Neighbouring countries (grade A)

Each country having a neighbouring country with a different written language (and this means most countries) has a requirement to be able to data process some or all of the national-use characters of its neighbours. In most cases there will be a relatively large number of people from neighbouring countries resident for shorter or longer periods in the country in question and DP installations have to be able to store and reproduce names which may include neighbouring national-use characters. At present there is either no support for this, or very rudimentary support.

For example in Denmark it is mandatory to be able to reproduce all national-use characters from Norway, Sweden and Germany. At present there is no problem with Norway, which has the same character set, but the Swedish and German Ä, ä, Ö, ö are non-existent. The German Ü, ü are either non-existent (alternate character set) or present only in a single case version (at a code point which differs from all other implementations of Ü, ü).

The Danish prime-minister at the time of writing is Poul Schlüter. Here is his name as reproduced at one installation in Denmark:

UPPER CASE	lower case	Equipment
SCHLÜTER	Schlüter	Non-IBM, 3270 compatible
SCHL≤TER	Schl-ter	IBM 3270 alternate
SCHLüTER	Schlüter	IBM 3270 standard

Note that only the non-IBM equipment can do the job correctly. There are many ad-hoc 'standards' in different installations which attempt to solve the problem. The latest development in Denmark is a series of RPQs introduced by banking and finance installations to provide some sort of homogenity - at least in their world where financial transactions and customer names and addresses are exchanged on a network basis. Needless to say, this is just one more example of an ad-hoc solution which further confuses the total picture. Next after problems concerning a country's own national-use characters, the 'neighbouring countries' problem is probably the worst one for most DP-installations.

The requirement is that no such problem should exist.

3.1.9 Entry and editing of "scientific" text (grade B)

Greek characters and mathematic symbols are not part of the normal IBM character set. It is thus today impossible to type and edit a text containing greek characters and mathematic symbols with these characters visible on the screen.

The IBM strategic in-house-publishing product DCF "solves" the problem of printing such text by allowing the change of font command. However, typing

```
.bf greek
p
.bf special
+
.bf roman
p
.bf special
A
.bf roman
K
.bf special
+
.bf greek
S
.bf special
+
```

to obtain $\pi^+ p \rightarrow K^+ \Sigma^+$ is extremely clumsy compared to many (non-IBM) systems

having these symbols as part of their normal character set. This results in

- decreased productivity in typing the original text
- increased number of cycles for printing/correcting drafts

and thus in increased costs.

Looking at the market offerings of non-IBM equipment and the program offerings developed by the Research Division of IBM it is clear that the market segment having these requirements is large enough to be taken very seriously.

It should, however, be clear that this requirement should not be confused with the formatting of mathematical formulae in a text processor. Here we speak about typing and seeing the actual characters and not how they should be laid-out on the finished document.

3.1.10 Entering and editing text using a foreign alphabet, or special diacritical marks used in transliterations (grade B)

With the invention of electronic printing techniques (laser printers, dot matrix printers, ink jet printers, ...) the demands of the end user for flexibility in the use of foreign alphabets (e.g. cyrillic) mixed in with roman text, special diacritical marks etc. are met in many devices.

What is not met, and the reasons for it appear ridiculous to the end user, is the possibility to enter and edit such texts with the characters visible in a recognizable shape on the terminal used. To use DCF again as an example:

.bf cyrillic
ABV

is not really recognized as **АБВ** resulting in a much more error-prone text entry.

The customer base for the requirement of foreign alphabets includes a number of large export oriented manufacturing firms and banks. Requirement of the special diacritical marks is very large in computerized library catalogues - it is very sad to read the introduction to the new British Library catalogue, comparing it to the old manually typeset catalogue!

3.1.11 User defined code points (grade B)

There is no space left in the code pages for user assigned graphics.

Some installations require graphics like "diameter" ø or "horizontal" ⊓. But where to place them in the code table?

A solution should provide for some codepoints not to be used by general software (printer drivers, compilers, etc.) but assignable by the user (installation).

3.1.12 System use of code points (grade C)

Besides the controls with codes X'00' to X'3F' some products require additional codepoints for internal use.

Fortran compilers (G1, H, H-extended) for example use X'FF' as an internal string delimiter. Thus it is impossible to define this code within a string to be transmitted to a file. Also within a FORMAT statement this codepoint cannot be used.

The use of this code point for this purpose is not bad but it should be known and clearly stated.

3.1.13 ASCII to EBCDIC Translation (grade B)

For reading and writing ASCII tapes on an MVS system there is only one translate table in the system. This table is based on the "old" EBCDIC assignment or "character set 103 of code page 256". An installation has no choice of installing several of these tables to support the various EBCDIC assignments. The final solution, however, should be that only one standard table is needed.

Using only a translate table rather than a real translate process does not allow the use of ANSI control sequencies (e.g. to switch to different character set, set tab stops etc.). In a mixed environment with both IBM- and "ASCII"-machines a more flexible generalized conversion mechanism is needed. With growing networks the number of "mixed environments" is also increasing.

The PC behaves in many aspects as an ASCII device, but the use of the control characters there is highly dependant on the device (screen, printer, ...). There is also no way to make the special graphics of the upper part of the code table availble to the EBCDIC environment, since the 8-bit code cannot be represented in two 7-bit code planes with a switch (SO and SI). This de-facto standard is not supported by any non-PC or host based software known to us. So the micro to mainframe link is still restricted to the US-ASCII graphics.

HEXA-DECIMAL WERT	0	1	2	3	4	5	6	7	8	9	A	B	D	C	E	F
0	LEER (NULL)	►	LEER (SPACE)	0	@	P	‘	p	Ç	É	á				∝	≡
1	☺	◄	!	1	A	Q	a	q	ü	æ	í				β	±
2	☻	↕	"	2	B	R	b	r	é	Æ	ó				γ	≥
3	♥	‼	#	3	C	S	c	s	â	ô	ú				π	≤
4	♦	¶	$	4	D	T	d	t	ä	ö	ñ				Σ	∫
5	♣	§	%	5	E	U	e	u	à	ò	Ñ				σ	∫
6	♠	▬	&	6	F	V	f	v	å	û	ª				μ	÷
7	•	↨	'	7	G	W	g	w	ç	ù	º				τ	≈
8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿				Φ	°
9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐				Θ	•
A	◙	→	*	:	J	Z	j	z	è	Ü	¬				Ω	•
B	♂	←	+	;	K	[	k	{	ï	¢	½				δ	√
C	♀	∟	,	<	L	\	l	¦	î	£	¼				∞	η
D	♪	↔	—	=	M	]	m	}	ì	¥	¡				∅	²
E	♫	▲	.	>	N	^	n	~	Ä	Pts	«				∈	■
F	☼	▼	/	?	O	_	o	△	Å	ƒ	»				∩	LEER "FF"

Code table of PC

3.1.14 Error graphic (grade B)

There is a strong need for an "error graphic". Currently undefined codes (no graphic assigned to that code) are presented on most devices as blank or minus '-'. This is completely misleading.

Assume for instance an Assembly language program (still in use!) specifying an area by DS C' ' (define storage) rather than DC C' ' (define constant). The DS defines (by action of the linkage editor) an area of X'00'. During program developement nothing happens because the stuff is printed only. But sent to a 327x this will be interpreted as control (buffer address)!

The 'error' character would be useful for displaying any 'text' that has codepoints with no graphic assigned on the outputdevice.

It may also be advantageous to be able to specify the 'error-fill' character or characters to be used in a particular output.

3.1.15 Currency Symbols (grade A)

The $, ¢ and £ code points and graphics lead an interesting life outside the U.S.A. To take just the $ as an example:

- The code points vary from country to country. X'4A' for Hebrew speakers, X'E0' in Japan, X'5A' in Brazil and X'5B' as 'standard'.
- In the English language area the UK has special problems. The £ occupies the same code point as the US $ (X'5B') whilst the $ occupies the same code point as the US ¢ (X'4A'). Since these are almost the only 3270 character set differences between the US and the UK, there is a tendency for suppliers to provide US instead of UK equipment. Thus £ frequently prints as $ and $ as ¢. Obviously an undesirable state of affairs.
- In Denmark, Norway, Sweden, Finland and Spain the $ is not available, its code point being occupied by Å or Pts.
- In Austria and Germany the $ is available to standard character set users, but not to those using the alternate set.
- The X'5B' 'dollar' position has been historically overworked as a delimiter character and is in fact still the default for JES2 - with unpleasing results in many countries.

The requirement is that if discrete currency symbols are to be implemented in the future, then they must occupy internationally fixed positions in the code table.

The NCTF cannot hazard a guess as to how many discrete currency symbols are neccessary. Presumably we require the $, ¢, £ and ¥ for international work. Note that many countries do not use discrete symbols, i.e. the currency symbol is compounded from upper and lower case alphabetics (thus Swedish kroner are 'kr' inside the country and 'SKr' or even 'SEK' outside).

3.1.16 Text and code

Text is a flow of characters. Their meaning is highly dependent on the context. Also the used graphic has different meanings in different environments. AM is an abbreviation both for Amplitude Modulation and Ante Meridiam! For coded characters it's even worse. Once in the computer storage the intention of the human is lost - if the meaning is not determinable by the context. With the current practice of many code tables used in different devices, the meaning is lost totally.

In the ANSI world (the mini computer world) a common practice is the code extension technique. Within the text the meaning of the code is defined. To switch to another meaning of the code a control sequence is used. The only thing needed is an authority defining the graphics with the associated control sequence... And this authority already exists. For example the ISO character set used in Videotex (identical to ISO 6937 part 2 with the exception of 2 flying accents not present there) is identified by **ESC) b** for use as G1-set.

There should be a distinction between input, storage and output of graphics (this term is used for punctuations, latin characters, semigraphics etc.). The current practice leaves out the intention of the text. Text with national-use-characters in it becomes meaningless, if this text was entered using different devices. To edit this White Paper we had to set up conventions on "national characters". A tape written in Denmark, read into a system in Switzerland cannot be interpreted without an accompanying manual description!

The input of special characters may be done with two keystrokes on an international keyboard or with special keys on a national keyboard. But the code generated and transferred to storage must reflect the complete meaning (e.g. base graphic and accent or visual interpretation of diphthong etc.).

On output the capabilities of a device must decide what to do with the internal representation of a graphic:

- place the accent onto the base character by overprint
- leave the accent out, if there is only a limited character set (which would be a bad option for Turkish)
- substitute the two byte sequence (e.g. print ae instead of ä)
- select a combined graphic (e.g. for 3800 printer)

3.2 National language problems

Once entered into the system, the various national use characters combine with the Latin alphabet to form text in a national language. This section describes the problems encountered in processing national language text. Also included are problems encountered in getting software products to 'converse' in the desired national language. These problems are of special concern for products which reach the end-user directly - for example 'information center' products.

3.2.1 Messages (grade B)

Messages are not generally available in the national language of the country in which a given software product is being used. In most cases there is no easy, well-documented method for the customer to translate these messages. In some cases the text length available for a possible translation is too small.

The term 'messages' in this context is to be understood as

- Automatic error messages
- Automatic informational and action messages
- 'HELP' information explicitly requested by the user.

The following examples are taken from IMS messages. Although not classed as an end-user product IMS (like CICS) provides the end-user of a DP application system with a bewildering variety of messages of a highly technical nature which are difficult enough to understand for English-speaking users. The first step is to be able to see these messages in one's own language. Here are some literal translations :

```
DFS053 TERMINAL RESTARTED - PLEASE REFORMAT SCREEN
DFS053 Votre terminal vient d'être re-démarré.
       Veuillez ré-initialiser votre contenu d'écran
       (French)
DFS053 Terminal wieder gestartet -
       bitte Bildschirminhalt neu aufbauen
       (German)
DFS053 Terminalen er genstartet -
       vær venlig, at genformatere skærmen
       (Danish)

DFS064 NO SUCH TRANSACTION CODE
DFS064 Le code-transaction n'existe pas
       (French)
```

```
DFS249 NO INPUT MESSAGE CREATED
DFS249 Keine Eingabe erhalten
       (German)

DFS058 EXIT COMMAND COMPLETED
DFS058 Exit kommando genomfört
       (Swedish)
```

The IMS system has no formal procedure for translating these messages. Some messages are in separate load modules, others are contained in program modules and the actual text content of a given message may be run-time variable.

As already mentioned, the content of messages and help information may be so full of DP jargon as to be unintelligable even when literally translated. So the translation process is required not only to achieve the desired national language but also to

- correct spelling from US to UK English
- provide dual case text
- provide more lucid and jargon-free explanations
- provide application-orientated text

Here are some more examples to illustrate this, all in English in order to show that the problem is not confined to non-English-speaking countries :

```
JOB CANCELED
This job has been cancelled.
```

(Dual case, spelling corrected, same content but more user-friendly)

```
DFS064 ?
DFS249 ?
```

(In reality both messages, as shown in the introductory example, simply state that IMS cannot understand or accept the input. The installation has replaced both texts with a '?').

```
DFS058 PRESS PA2 TO GET LOGON SCREEN
```

(The user is told what to do next. This translation could only be achieved by using a fairly complicated exit routine, since the first word of the original message, 'EXIT' in the introductory example, is run-time variable)

Note that in some installations there has been a tendency to remove messages completely, in order to protect the end user from sudden bursts of 'foreign language'. Needless to say, this approach has abvious disadvantages when errors have to be diagnosed!

The requirement is that all messages be capable of customer-translation according to the individual customer's needs. The message may be required to be shortened or (considerably) lengthened by translation. The verbal content (meaning) of a message must be unique for a given message number (processing situation) and must be held separate from any run-time variable information. The method of translation must be well-documented and easy to carry out in practice - for example editing of a file or re-assembling of modules containing the text.

Note also that the customer may only wish to translate a frequently used subset of all possible messages for a given product (For IMS, under 5% of possible messages ever reach the application user's screen).

Thought should also be given to the customer's maintenance problem when installing new versions of the product. the requirement is that previous 'translations' should be useable without the need for complete retranslation. This means that new versions should specifically document

- Altered meanings to existing messages
- New messages

in relation to the previous version.

3.2.2 Reserved words (grade B)

It is not generally possible for the customer to specify alternative reserved words or keywords in his own national language. This is of special concern in products which reach the end-user directly as for example Information Centre type products.

For instance :

```
CREATE VIEW MIJNZICHT (WKNR,NAAM,AFD,JOBK)
  AS SELECT WERKNEMERNR,FAMILIENAAM,AFDELING,JOBKODE
  FROM DSN8.TEMPL
WHERE JOBKODE = 52
   OR JOBKODE = 54
   OR JOBKODE = 56
SELECT *
  FROM MIJNZICHT
```

may be required to be written as :

```
MAAK ZICHT  MIJNZICHT (WKNR,NAAM,AFD,JOBK)
  ALS KIES  WERKNEMERNR,FAMILIENAAM,AFDELING,JOBKODE
  VAN  DSN8.TEMPL
WAAR  JOBKODE = 52
  OF JOBKODE = 54
  OF JOBKODE = 56
KIES   *
  VAN  MIJNZICHT
```

which is now completely in Dutch. The requirement is that such specification of 'synonyms' be possible and that the specification process be well-documented and easy to do in practice - for example by re-assembling a module containing the text or by having the text in a separate file which is accessed during the product's initialisation phase.

3.2.3 Fixed text (grade A/B)

It is not generally possible for the customer to specify fixed text in his own national language - and it is mandatory that all user-directed output from an execution be entirely in the desired national language.

By 'fixed text' is meant the type of (usually short) text which has a tendency to be hard-wired into a product, often in several different modules. Examples are

- 'PAGE'
- 'DATE'
- 'CONT.' or 'CONTINUED'
- The names of the days
- The names of the months
- The names of colours
- Text accompanying graphs, charts and tables
- etc. etc.

The requirement is that fixed text be specifiable by the customer as for example in 3.2.2.

3.2.4 Menus and Prompts (grade B)

This is presented as a separate problem area since although analagous to the previous subsections, the technical aspects are different.

It is not always possible for the customer to specify the fixed text fields of a menu or a prompt in his own national language. Even when this can be done, there may be

problems with fields which require fixed input in English.
For example

```
Copy complete
Copy another (Y/N)?
```

Must be translatable to

```
La copie est terminée
Encore une copie (O/N)?
```

and the product must be able to accept the French **Oui** instead of **Yes**.

The requirement is that such translation be possible, that it be easily performed and the method well-documented. As with 3.2.1, the customer must be helped in the maintenance situation.

3.2.5 Dates (grade A)

It is usually impossible to specify the form of the date which is to appear automatically on user-directed output. It is mandatory that dates supplied automatically by a product be in a nationally or locally understood format.

To cut an otherwise lengthy description of the problem to a minimum, it can be stated that the automatic date format should be entirely at the customer's control, especially since the format may be application-specific (even the ISO standard may not be acceptable!)
For example the American date format for February 4th 1985, supplied automatically by 'product X' would typically be

02-04-85

which is understood as 2nd April 1985 in most of Europe. The requirement is that the customer be able to specify

02/04-85	USA
1985-02-04	ISO standard
04.02.85	Europe
4 2 1985	Europe
4-FEB-85	Europe
4 FEV 1985	France
4 février 1985	France
Den 04/02 1985	Denmark
LUNEDI 040285	Italy

or whatever.

More problems will be solved by allowing freedom in this respect than by assuming national standards. The examples above are not all national standards but they are reasonable and nationally or locally understandable in each case.

The 'fixed text' problem of day and month names has been covered earlier.

Note, by the way, that a global IBM 'utility solution' to the date formatting problem would be the best long-term goal. The automatic date format would then reduce to a simple special case. The global solution would of course be 'callable' from any programming environment and would include the necessary new developments required to distinguish dates before and after year 2000.

3.2.6 Thousand and decimals (grade A)

It is generally not possible to specify the delimiting character for editing thousands and decimals when the output is produced under the control of a standard product, or when input is interpreted under the control of a standard product. For example

10,000 yards	is ten thousand yards in the USA.
10,000 meters	is only ten meters in Europe.
1.000 hours	is one hour in the USA, but nearly a month and a half in Europe.
113.500 guilders	may leave the recipient in doubt as to whether he is rich or not.

It is required that the customer be able to specify the character to be used to delimit thousands and decimals when the output and/or input is under the control of a standard product. This must be 'parameter option' and not a 'standard for Europe' since usage varies from country to country. For example Sweden uses the decimal point whilst Denmark, Germany, Austria and Switzerland use the comma.

Note that the input situation raises interesting problems, since the comma has become generally accepted as an inter-field delimiter for 'stream' input. The implication is thus that the inter-field delimiter(s) also be user-specifiable. Currently it may be impossible to enter a comma in a numeric field on certain terminal equipment.

3.2.7 User-specified names (grade A)

National characters are not always eligable for use in user-specified names.

For example MVS JCL only allows use of the 3 'traditional' national use characters, with graphics corresponding to code points X'5B,7B,7C', in names on JCL-statements. This means that users of the standard Austrian/German 3270 code cannot include the national characters Ä, Ö, Ü in their JCL (they could with the original 'alternate' code) :

Country	JCL	Result
U.S.A.	//DD1 DD DSN=ANDERSEN.#69K,...	OK
Denmark	//DD1 DD DSN=KMD.ÅRHUS,...	OK
Germany	//DD1 DD DSN=OERLIKON.BÜHRLE,...	JCL error

It has also been suggested that lower case should be accepted, together with some special symbols such as the useful '_' for binding words together.

The requirement is that all software must at mimimum accept valid national characters and that the present over-restrictive rules in some products be relaxed wherever posssible.

3.2.8 Lengths of names (grade C)

It has been suggested that the 8 character limit on the lengths of some variable names may be too short to allow reasonable mnemotechnic use in some countries.

- member names in partitioned data sets
- elements of data set names
- keywords
- etc.

It would appear to be a reasonable requirement, especially in end-user directed products, that names could be longer than 8 characters (16, 32 ?) or in fact variable-length up to a fairly high limit.

3.2.9 Delimiters (grade A)

This is a typical 'grey area' for many countries. The problems which arise can be roughly grouped into two areas :

a) Use of the three 'traditional' national use code points X'5B,7B,7C' (with US graphics $, #, @) as syntax elements or delimiters.

   A typical example here is the very popular Statistical Analysis System (SAS) which makes heavy use of all three of these US graphics in its input/output syntax. This leads to programs which are confusing to read and in which the use of national characters in variable names is prohibited.

   Although not an IBM product, SAS has nevertheless been recommended by IBM as a useful info-center product.

b) Use of other special symbols such as [ ] { } \ .

The EBCDIC code points for these graphics are X'AD,BD, C0,D0,E0'. Looking at the 3270 codes, however, we find [ ] assigned to X'4A' and X'5A' for Belgian, Portuguese, Spanish and Swiss users. All 5 code points, X'4A,5A,C0,D0,E0', are also candidates for national characters. It is therefore certain that a mainframe software product using these symbols will run into some sort of trouble in Europe.

Using a national ISO standard coded character set on a PC, one or more of these graphics may be replaced at the standard ISO code point by a national character. Examples of software products which are adversely affected by this are PASCAL compilers and the IBM-marketed MULTIPLAN.

The requirement is that no such 'dualling' of the graphics for a given code point (and vice versa) exist. A national character graphic must never carry accidental syntactical significance by virtue of its code point.

3.2.10 Case conversion (grade A)

Some products may still employ 'folding' to convert lower case entered at terminals to upper case for further processing. This has obvious repurcussions for national use characters which have code points between X'3F' and X'80' and are therefore unchanged by folding (OR-ing with X'40').

Lower case to upper case translation only respects the 26 letters of the english alphabet. For example, 'é' stays as an 'é' instead of an 'E' with no accent. Furthermore, if there is a hardware folding facility on some 3278 screens, it concerns its whole contents, and seems to ignore some special graphics e.g. these associated with the dead key if the PRPQ is installed.

Another consequence is the translation of national characters to blanks or punctuation marks when automatically folded to upper case in many software products.

Further there are usages that can differ from one country to another. For example, 'European French' people do not convert accented letters into upper case accented letters, as the Canadians do. Thus the 'European French' upper-case version of the sentence

"Les bûches brûlent tête-bêche dans l'âtre de la cheminée: c'est Noël" is
"LES BUCHES BRULENT TETE-BECHE DANS L'ATRE DE LA CHEMINEE: C'EST NOEL".

Whereas the Canadian French is:
"LES BÛCHES BRÛLENT TÊTE-BÊCHE DANS L'ÂTRE DE LA CHEMINÉE: C'EST NOËL"

There are also cases in application systems where upper to lower case conversion must be done and a new problem is obviously encountered here when a non-accented **BUCHES** must convert to the accented **bûches**.

No products can do this now and the conversion function is usually achieved by

software with loss of accents from lower-case to upper-case. Some display hardware, however, is equipped to do the right job. This is to say that it is possible.

The requirement is that case conversion can be done in both directions and without loss of information.

3.2.11 Translating (grade A)

Incorrect results are often obtained via the use of translate tables incorporated into products. These tables are used variously for

- case conversion
- code conversion
- screening of terminal I/O
- etc.

The tables are invariably set up for US standard usage and do not reflect national use characters. When the customer realises what is wrong, he begins the search for the translate tables, which may not be easy to find and may not be documented.

The ultimate requirement is of course that no manipulation of translate tables be necessary - that is that the required national tables are automatically selected and used, or that one standard table is sufficient. The interim requirement is that tables are well documented as to position and usage so that the customer can make any required modifications easily.

3.2.12 Double characters

There are several problems associated with the existence of 'double characters' in written languages. All have a tendency to be special cases and the only common denominator is the fact that the resulting two contiguous character positions must be treated as one unit during processing. There is no IBM hardware or software solution for any of these problems.

Example 1: Single character and equivalent double character may coexist in same text.

> The Danish 'long a' is correctly written as Å, å. However this usage stems from a fairly recent spelling reform which allowed continued use of the old forms AA, Aa, aa in proper names.
>
> Hence **AA**LBORG, **Aa**lborg, **Å**rhus, **Aa**rgaard, **Å**se, **Aa**se, mål etc.

Example 2: Single character lower case character converts to double upper case character.

> The German lower case 'double s' is correctly written as 'ß' which becomes 'SS' in upper case.

Hence Hauptstraße, HAUPTSTRASSE.

Example 3: Unable to enter a national-use character, which therefore has to be split into a (strictly illegal) double character.

The French (and latin) dipthongs Æ, æ, Œ, œ cannot be entered as such on a French keyboard. The only alternative is to enter them as the double characters AE, ae, OE, oe.

Hence the correct œuvre becomes oeuvre.

An interesting (and deplorable) side effect of the double character problem is introduced in the last example. Many DP-personnel have given up the unequal struggle of transmitting national-use characters outside 'the country of origin' and use artificial double character combinations instead.

This is specially prevalent in names and addresses:

Günther Krysmanski	becomes	Guenther Krysmanski
CH 8050 Zürich	becomes	CH 8050 Zuerich
DK 8200 Århus N	becomes	DK 8200 Aarhus N

All of these are glaring inaccuracies caused by the lack of proper international character support in current systems.

3.2.13 Sorting single case text (grade A).

The order of items sorted alphabetically on one or more character fields is incorrect.

Example from France

EBCDIC collating	correct sequence
frêle	frêle
frêne	frelon
frelon	frémissement
fret	frêne
frémissement	frère
frère	fret

Examples from Denmark :

EBCDIC collating	correct sequence
ÅGÅRD	ANDERSEN
ÆGIDIUS	NIELSEN
ØRSTED	ZEUTHEN
ANDERSEN	ÆGIDIUS
NIELSEN	ØRSTED
ZEUTHEN	ÅGÅRD

3270 alternate	3270 standard	correct sequence
ørsted	ørsted	andersen
ågård	andersen	nielsen
ægidius	nielsen	zeuthen
andersen	zeuthen	ægidius
nielsen	ægidius	ørsted
zeuthen	ågård	ågård

Example from Germany :

3270 alternate	3270 standard	correct sequence
Buße	Busfahrt	Busfahrt
Busfahrt	Buße	Buße
Busse	Busse	Busse
Göfis	Göfis	Gasse
Güte	Gasse	Gäßlein
Gäßlein	Güte	Göfis
Gasse	Gotte	Gotte
Gotte	Guthaben	Güte
Guthaben	Gäßlein	Guthaben

If the product which executes the sort uses the Sort/Merge utility it may be possible to specify a national 'ALTSEQ'. If this parameter cannot be specified or if Sort/Merge is not used, the sequence will be incorrect (e.g. sorting in the EDITOR of ISPF).

Even when ALTSEQ can be specified (e.g. when using Sort/Merge directly), some extra user action is usually/always (?) required. When the sorting algorithm is coded directly in the application program, complicated procedures are necessary in order to ensure correct sequence (at least one translate of all sort fields, or two translates, depending on the methods used).

It is an accepted fact that correct sort sequence cannot be achieved without causing some degree of CPU overhead (for example between 2 and 5% for typical sorts using Sort/Merge and ALTSEQ).

The requirement is that correct national sort sequences be achieved automatically (as for 3.2.11 Translating). It should be noted, however, that it must still be possible for the user to manipulate the sort sequence (as with ALTSEQ) for special applications.

3.2.14 Sorting mixed case text (grade B/C).

It is impossible to sort mixed case text correctly unless some type of 'ALTSEQ' mechanism is used. The code points for the lower and upper case latin alphabet in EBCDIC cause the difficulty. The situation is further complicated by the presence of national characters.

Sorting mixed case text thus requires special mechanisms and involves CPU overhead.

National double characters require very complicated pre-processing of the input to the sort if correct sequence is to be achieved. For example :

Germany Lower case double s is a single character ß which must sort in company with upper case 'SS'. A-umlaut Ä and a-umlaut ä must sort in company with 'ae', 'Ae', 'AE' or with 'a' 'A'.

France The (single character) dipthongs Æ, æ and Œ, œ must sort in company with 'ae', 'Ae', 'AE' and 'oe', 'Oe', 'OE'.

Denmark Å,å 'AA', 'Aa' and 'aa' are all functionally equivalent and must sort in each other's company. Note that the equivalence of Å and AA also gives problems when sorting in upper case!

There is currently no IBM support for correct mixed case sorting with double characters. There are, however, non-IBM systems on the market which handle this situation correctly. The problem area is of increasing concern as more applications make use of mixed case text data.

The requirement is that sorting of mixed case text be automatically and correctly achievable as for 3.2.13.

3.2.15 Multiple languages (grade B)

The problems described in section 3.2 relate to the situation in a single European country using a single written language. However, as has been noted in section 3.1, most installations require the ability to process some words containing national use characters from neighbouring countries and many installations require the ability to process complete texts in other languages. These problem areas are in fact only subsets of the general problem which is seen at its worst in multilingual countries, where several different languages must be processable at all installations. Major examples of these countries are:

Switzerland	(German, French, Italian)
Belgium	(Dutch, French)
Luxemborg	(French, German)
Spain	(Spanish, Catalan)

In addition we must not forget the increasing number of 'international facilities', for example information retrieval systems which may be accessed from all countries via international tele-networks.

An additional and extremely important requirement for the whole of section 3.2 is thus the ability to operate software in the language of one's choice.

The 'switch' from one language to another should be at minimum possible on a job-step or on-line session basis:

not	`ispf`
but	`ispf lang=german`
not	`// EXEC PGM=SORT`
but	`// EXEC PGM=SORT,LANG=SWEDISH`

For on-line sessions it would obviously be desirable in multilingual countries to be able to switch languages in mid-session, without the need to logoff.

Note that the language switch must initiate all aspects of national language and national character support such as translate tables, sort sequences, error messages etc. etc.

3.3 Keyboard related problems

It is difficult to talk about the national character aspect of keyboards without mentioning the many other deficiences in current keyboard implementations. We therefore make no excuse for expanding the subject matter of this section to include aspects which are not specifically related to the national character problem area.

The important point to note is that the national character aspects of keyboard operation and geography must be integrated into a much needed 'international standard keyboard solution' such that national keyboards appear as well defined and easy-to-use subsets of a specific standard.

IBM offers a bewildering variety of keyboards with different layouts, but the tendency nowadays is that a given keyboard may in fact be in use for one or more application types arbitrarily lelected from

- Programming and computer operations
- Data entry
- On-line DB/DC preprogrammed applications
- Information center (including APL)
- Text processing
- Graphics
- etc. etc.

For most of these keyboards it can be stated that

- The geography is wrong for a given application (for example a typist asked to use a 3278 keyboard, or anyone trying to activate 'clear screen' or 'PF1' on this keyboard with one hand, holding a telephone or a pencil in the other!)
- National character support may be poor for the country in question, bad for neighbouring countries and probably non-existent for international use.
- Confusion and low productivity result when a user has to deal with a number of different keyboards at the same site (watch a PC-user trying to manipulate the cursor control keys on a 3278!).

3.3.1 Non consistent character sets across products (grade B)

Look at these two IBM products: two keyboards designed for France.

French IBM 3278 keyboard

```
! " $ $ % + / ( ) = ? ¨
1 2 3 4 5 6 7 8 9 0 ' ^

A Z E R T Y U I O P ç *
a z e r t y u i o p à &

Q S D F G H J K L è ° £
q s d f g h j k l é ù

> W X C V B N M ; :
< w x c v b n m , . -
```

French IBM typewriter keyboard

```
1 2 3 4 5 6 7 8 9 0 °
& é " ' ( § è ! ç à ) -

A Z E R T Y U I O P ¨
a z e r t y u i o p ^

Q S D F G H J K L M %
q s d f g h j k l m ù

W X C V B N ? . / +
w x c v b n , ; : =
```

The first keyboard comes from my 3278 terminal, the second one represents the keyboard of my secretary's IBM typewriter. It is obvious that decimal digits are in lower case on the 3278 and upper case for the typewriter but they are at least on the same keys.

Special characters are not placed at the same location. One key, a dead key on the typewriter with the circonflex accent (â,ô,û) and the trema (ë,ü), does not work in the same way on my 3278 terminal which has no dead key facility. Even the PRPQ which allows this facility does not give the full flexibility of the dead key. But the worst is the **M** key, an important alphabetic character, which is not on the same row in the two keyboards.

This simple example explains why French secretaries are very reluctant to use IBM 3278 terminals for Word Processing. The French case is specific to the French character set but similar situations exist in other European countries. German secretaries, Scandinavian secretaries and so on are also reluctant to move from their typewriter keyboard to a WP terminal which changes their habits, decreases their

productivity, adds more mistakes and does not contain all the needed features.

Even UK, an English, not American speaking country, is concerned with the pound key (£) and the dollar ($) key when the internal representation is exchanged. Just note also that my French keyboard contains the pound (£) as the national monetary symbol and not the French franc (F) but the IBM 3203 printer chain prints the pound sometimes as a big F, the french franc, or as the US number symbol # which is no longer available on my screen.

3.3.2 National and international keyboards

Specialized hardware exists, e.g. for newspaper composition, but in fact, we don't know any "infocenter" keyboard able to enter in a simple way the whole set of characters that it would be advisable to have under the hand for current national work, and furthermore for international needs. The alternatives offered are very poor:

- if one uses for example DCF/GML, one can enter text, but one has to remember to enter some complex expression for each character combination provided and to spend extra software overhead when interpreting it, and to spend new human time to verify it in output, and to rerun in case of mistakes ...
- if one is not entering text, one can choose a technique (well-known in APL) which consists of selecting a needed character from a matrix by indexing, but this is not possible in all contexts, and doesn't allow character combinations.

A better approach is given with use of the Programmed Symbol Sets 3279-2b or 3b (and also some 3278 special models), which enable one to load and select the character sets required. They are named "Programmed Symbol Sets", and one can define them oneself (using a special editor) if necessary. But there are several major inconveniencies:

- In fact, one needs a special environment, installed and maintained by trained personnel (not cheap by itself !), which does not give any immediate support.
- One is producing codes that can be recognized and processed only (today) in the same environment, so it's very difficult to merge them with other sources and produce an external text output.
- One has to remember your codes rather than find them on the key-tops. The keyboard template supplied with some models is not usable for serious work and it doesn't fit on the handy models.

3.3.3 Lack of grouping of functions (grade B)

Existing keyboards for 327x devices try to group several functions like screen control, programmed functions, cursor and field control. Since the functions of terminals have become more complex (colours, screen sizes, ...) or a device has become multi functional (3270 PC or common PC with emulation etc.) a good grouping of functions becomes very necessary. But basic design principles are often missing:

3270 PC

- HOME only in conjunction with ALT
- PC functions intermixed with terminal functions on same key (CAPS LOCK / SHIFT LOCK). Shift lock is a reminescence from the stone age of typewriters and should be abandoned completely.
- different graphics assigned to a key in terminal and PC mode (!¢ with {[ ).
- mixing of cursor control in PC mode with numeric cluster
- etc. etc.

3278

- HOME only in conjunction with ALT
- extra key for < > but comma and period twice (upper and lower case) on same key

3279

- PF1-PF12 in one row on top rather than in a PF-cluster and PF13-PF24 on shift-(PF1-PF12). The linear arrangement may be good for placing a template, but it is very hard to pick a specific key without reading the keytops. The geometrical arrangement 3x4 can be used by intuition.

In general there are too many keys for normally sized hands in the central area of the keyboard (typewriter area). Although this is according to an ISO standard, the key left of Z is very annoying. Also the rightmost key in the "home row" with { } is not good. They both increase the distance to the SHIFT and RETURN keys. This is at least true for devices sold in Switzerland.

Currently we have to deal with many keyboard layouts:

- typewriter keyboard
- data entry keyboard
- data entry keypunch keyboard
- APL keyboard
- text keyboard

In many cases one person is responsible for several jobs and thus has to become familiar with all of these layouts. Of course the usage of so many different keyboards is error prone and decreases productivity.

4 Required characteristics of solution

This section lays out the conditions proposed by the SEAS user community for future solutions to the problems described in section 3. The technically expert reader will have realised that the **general** solutions for the three problem areas in section 3 are

National character problems

A character set which completely satisfies the needs of a particular language and which additionally allows the use of characters for all European countries without compromise of any kind. Each character has to have a unique representation in order to enable communication with other countries.

Hardware devices which support this character set.

National language problems

Software products and processes which are language independent and which can be switched to the desired language by a simple and automatic process.

Keyboard related problems

Keyboards which are nationally standardised so as to give optimal efficiency for entering text in the primary language or languages in a given country, and which also allow true multilingual use, i.e. all national use characters can be entered.

Standard layout and operation for other, non-language-dependent, functions.

For all three problem areas there is also the requirement that the present differences in character set and function between data processing (DP), word processing (WP) and personal computing (PC) be resolved into one 'system image'.

The chaotic state of affairs described in section 3 demands some sort of integrated approach to problem-solving. Many problem areas are so interlinked that further blatantly ad-hoc development (for example the PC character set) or piecemeal 'solutions' (for example the alternate character sets) must be banished.

SEAS believes that all future development in the national character, national language and keyboard areas must be coordinated in a well defined direction and that the objective should henceforth be a new level of system architecture.

SEAS requests that IBM give top priority to the developement and implementation of this architecture. SEAS also requests IBM to provide its customers with one or more 'statements of intent' as early as possible in the developement process.

These statements must be such as to enable customers to protect their considerable present and future investments in character set dependent hardware and software during the period until products based on the new architecture are generally available.

We will take the liberty of naming the new architecture **'National Language Architec-**

ture' or **'NLA'** in the remainder of section 4. Section 4.1 further outlines the required concepts in NLA. Sections 4.2 to 4.5 outline requirements for the conversion process.

We would like to emphasize here that these sections are very much 'open-ended'. We feel that a useful dialogue has been started with IBM in this very difficult and complex area. We hope that IBM will see the need to proceed with an untraditionally high level of customer consultation when planning for NLA. The present White Paper is a starting point and we believe that SEAS and the other IBM user organizations, GUIDE and SHARE, will be able to provide a focus for future discussions with IBM.

We do not specify here that NLA should be based on existing international standards. However we would strongly recommend that IBM consider the use of international standards wherever applicable. We would also ask IBM to take note of SEAS' special interests in this area, together with SEAS' expressed interest in participating in international standards work in the future.

The Task Force has also discussed the possible nature of the conversion process from the current status quo to NLA. There seem to be two extremes - **an evolutionary process** and a **revolutionary process**.

Again we cannot at this stage recommend either extreme; this decision must also be the subject of further analysis and consultation with the user community.

In practice we suspect that the process will be mixed - evolution over longer periods to protect customer investment coupled with one or more revolutionary stages to attain the final goal. In this light it is difficult to lay down requirements for the conversion process. Suffice it to say that sections 4.2 to 4.5 are based on the supposition of a single revolutionary change. Some items will obviously be redundant in an evolutionary process, but we feel obliged to guard customer interests in a 'worst case' situation (and here we think that the traumatic experience may easily be several degrees worse than a VSE to MVS/XA conversion).

Many of the items discussed will apply to any type of change and we are confident that IBM does its best in most cases to protect customer interests - which is what sections 4.2 to 4.5 are concerned with. However the Task Force is of the opinion that IBM's track record to date on protecting customer interests in the problem areas of section 3 is so bad that all possible initiatives must be taken to improve on this in the future.

As regards the specific area of keyboard design, we include the section 'Keyboard requirements' as an appendix to the White Paper. The literature is rich with proposals and new developments in this area and, as mentioned already, the NCTF has felt obliged to state its point of view here - keyboard design for NLA must be coordinated with the many other aspects of keyboard use.

Having said this, we acknowledge that the subject matter is in many ways peripheral to the main stream of the White Paper. 'Keyboard requirements' is thus to be seen as an 'appetizer' for very necessary future work and **not** as a set of requirements on the same level as the present section.

4.1 System architecture

As already stated, SEAS believes that the ultimate goal should be a new level of system architecture - **National Language Architecture** or **NLA**.

Amongst the characteristics of NLA should be

- A set of well defined rules at the highest level regarding the entering, displaying, storage, processing and interchange of data containing national-use characters, Greek characters, mathematical symbols and foreign alphabets.
- A well defined open-endedness to allow future development without the need for expensive hardware or software changes. It is acceptable that some expensive hardware features - like multiple character sets on an impact printer - are an optional enhancement.
- An end-to-end protocol for IBM system sender and receiver without the need for translation - even on an international basis.
- Transmission into and out of the IBM system environment (from and to equipment using other character standards, either supplier based or internationally agreed) via an open ended set of automatic translation processes.
- All graphics of all countries available everywhere and assigned to internationally unique code points.
- Provision of new hardware and/or software basic functions where necessary.
- Intelligent treatment of lacking graphic support for a given code point on all output devices. The action to be taken in each case to be user selectable/definable as an integral part of the architecture.
- Standardisation of keyboard geography and method of operation for all system types (DP, WP, PC etc.).

Once the basic concepts of NLA are agreed on and statements of intent or planning guides are available, the necessary software can be developed.

Note though, that many of the problems outlined in section 3.2 can in fact be tackled at once and without compromising future use of NLA facilities.

Amongst the characteristics of NLA-based software should be

- Character set independence.
- Language independence and language tailorability in **all** products and to a degree which can be controlled by the customer.
- National language versions for all 'major' end-user products and languages without the loss of tailorability if so desired.
- Full documentation for all language facilities and the process for tailoring them.

- Choice of language on at least a per session and per job-step basis without incurring the overhead of having to run multiple copies of the same product.
- National 'typography' (prefered quotation marks, date format, decimal point/comma, currency symbols etc.) and national sort sequence to be considered part and parcel of the language in effect at any time.
- Current 'language in effect' to be available to executing programs via a system variable. The syntax and value set of this 'language indicator' (and other necessary indicators) to be internationally agreed.
- Guidelines for writing language independent and language tailorable software. These guidelines to be enforced for all third party products marketed by IBM.

4.2 Coexistence of existing and new solutions

Existing methods and NLA will have to live alongside each other in a typical installation for many years, both during the period of NLA development **and** after NLA becomes generally accepted (witness the timelags experienced with conversions from BTAM to VTAM and BSC to SDLC). This coexistence of different methods must be capable of being controlled by the customer by means of existing (or new) methods of parameterization such as

- Default software installation parameters.
- Run-time execution parameters

```
// EXEC....PARM=
// DD......RECFM=
/*JOBPARM....
```

Session parameters

- VTOC/LABEL information
- On-line commands in TP-systems, System macro calls in application programs
- etc. etc.

The required mechanisms must be installed in all environments: OS, VSE, VM etc. and in all products: CICS, IMS, TSO, CMS, SORT/MERGE, SQL, COBOL, PL/I etc.

Traditional or NLA support must be definable for all I/O paths which are not directly specifiable via JCL such as

- JES writers

- Individual TP-nodes under control of CICS or IMS
- etc.

New hardware with NLA-capability must be designed so as to be mode-selectable by simple switching (or must be easily field upgradeable) from traditional support to NLA support. This applies specially to TP- equipment which may by distributed over a wide geographical area and which will introduce severe logistical problems if switching is a complicated process.

Needless to say, traditional (existing) and NLA support must be able to coexist on the same TP-network under control of one TP-system. It must be possible to drive NLA-devices from existing application systems without the need to modify user coding and without incurring loss of function or efficiency.

It is of course accepted that some degree of program modification will be necessary in order to fully utilise the new functionality in NLA.

4.3 Continuance of support

The user must not be forced into making conversion decisions because of the sudden unavailability of a product or the withdrawal of existing IBM support.

Hardware/software products must allow continued use of existing methods for a time period which is commensurable with the user's investment in these products. The Task Force has discussed periods of the order of 5 years but it should be noted that for example 3270 equipment on large DP networks often has a lifespan of over 10 years. The investments to be protected are not only those of the IBM customer-installation but also those of the installation's own customers in turn.

It is accepted that IBM will implement new functionality in new product releases or versions which support NLA, but every reasonable effort should be made to avoid situations where a customer is forced to accept NLA in order to benefit from new **non-NLA** functions. It is suggested that products be developed so as to allow 'NLA/NONLA' parameterization, rather than offering two versions of the same product.

Support for existing data representation must never be completely dropped - at least the required conversion products must always be available. This to guarantee the survival of large data archives which it would be impracticable to convert other than when actually accessing them.

4.4 Cost of new solutions

IBM should note that the European user community has already paid a high price for living with the present day chaos of character and language problems and product incompatibilities. SEAS would therefore suggest that every effort be made to reduce conversion costs, plus the cost of the actual NLA-support, to an absolute minimum.

Basic data and software conversion tools should be provided free of charge (or very cheaply) by IBM.

Conversion will under the best circumstances probably be a major upheaval for the typical installation. Potential 'cost-barriers' should therefore be reduced to an absolute minimum.

Ultimately we expect both customers and IBM itself should experience a marked reduction in effective costs as conversion to NLA is achieved.

4.5 The need for flexible conversion tools

IBM tools must be available for the conversion of stored data (disk, diskette, tape,...) to NLA format. These tools must be able to convert data organized for all access methods (QSAM, VSAM, ...).

Tools should also be provided for the conversion of existing customer-written source code.

Tools must be flexible, cheap in operation and easy to use - also by untrained personnel (end-users for example).

Conversion must be possible without corrupting existing levels of data security and integrity.

Full details of the hardware/software implementation of NLA together with the conversion tools must be available in the public domain.

Planning information must be released as early as possible and at suitably frequent intervals so that IBM customers and third party suppliers can plan for conversion.

5 Epilogue

The chairman of a task force can, I hope, be allowed the luxury of appending a few words of his own.

This White Paper has taken form over a period of roughly three years. The process has never been easy. Looking back on the result, I think I speak for all of us when I say that there are still a number of loose ends and of course there are errors and misunderstandings here and there. We have all had problems finding time to do the work, which has had to be squeezed out of the usual tight schedules which are the constant burden of all DP people.

On the other hand I would like to emphasize that the subject matter has come 'straight from the horse's mouth', from people in many different countries who experience the problems of section 3 as very real frustrations in their daily work. I hope that this rather direct approach is sufficient to offset any lack of technical completeness in the White Paper.

I am confident that our primary objective has been reached - a concentrated survey of a widespread set of related problems has seen the light of day and the IBM organization has been made aware in no uncertain terms of the plight of its European customers.

On my own behalf and on behalf of SEAS I would like to record many thanks to the following people:

- NCTF member Klaus Daube who has single-handedly been responsible for the layout and typography of the White Paper (using the SUSI text formatter and a XEROX 2700 printer at Oerlikon Bührle Rechenzentrum).
- Lisbeth Dreyer who word-processed the greater part of the text at Kommunedata's Hospital Datacenter.
- Jens Lynge, Handelsbanken, whose untiring efforts finally succeeded in getting an NCTF team together.
- Knud Nielsen, Handelsbanken, the first chairman of the NCTF.
- Jerry Andersen, the IBM representative on the NCTF, who has been indespensible in many ways. Jerry's expert kwowledge of character sets has carried us over many difficulties and he has pointed us in the right direction on several occasions.
- The managers of the SEAS installations who have provided the services of the NCTF members - and also the necessary machine time.
- Last but not least, the many SEAS people who have contributed with ideas and examples.

Finally I would like to thank Daud R. Matthews of the University of Petroleum and Minerals, Dhahran, Saudi Arabia, for the excellent and comprehensive documentation he has provided. This deals with the special problems encountered with IBM products

in the Arabic character set and language areas and was compiled by ARAMCO. As noted elsewhere, the NCTF has felt unable to include these problem areas in the White Paper, but we are confident that IBM is aware of the special needs of this user community and will cater for it in a future 'NLA'.

Peter Gardner 05.06.85

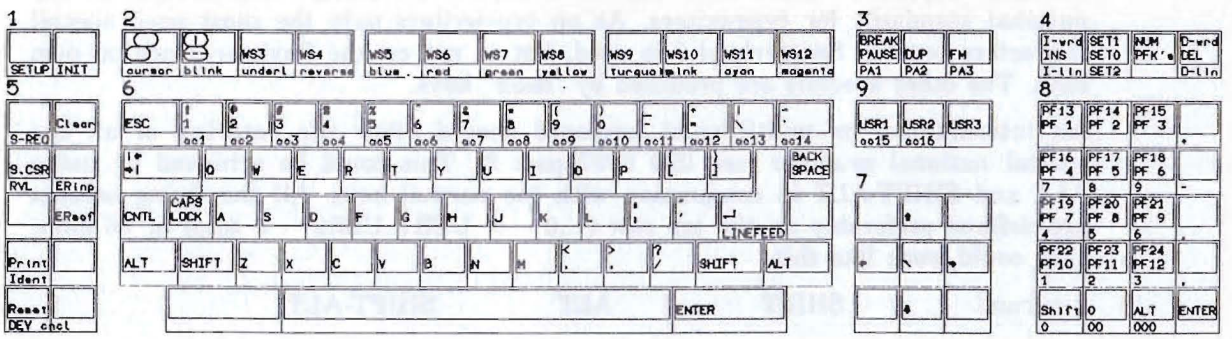
6 Appendix: Keyboard requirements

Our problem is not here to consider how to internally handle and process the entered codes, but only to give a user point of view on the new facilities that a good keyboard should offer for fair work; that is without forgetting the background of the general need to access various character sets and not only the national set.

For general purposes, we recommend that a standard keyboard offers:

- Immediate keys for a number of immediate functions such as Clear, Programmed Functions, etc. which are frequently used. These functions should be single keystroke functions.
- The same topology for 'classic keys' as in the traditional national disposition of the normal typewriter of each country. This implies that the function sub-keyboard(s) should be arranged separately from the classic keyboard. A further advantage of this would be that some of these sub-keyboards could be standard across all countries.
- Having a set of "scratch" keys that one could use for any desired internal code would be very useful. Today this is only achievable by software and results in loss of the normal usage of the affected key.
- For special usages like APL, it should be considered as a requirement that the whole range of characters be available. Also, the APL use should not modify the behaviour of the normal keyboard. An acceptable method of use must be implemented. Currently a true chinese game is required to find certain graphics when swapping normal mode to APL mode and vice-versa.

The following functions could be arranged into groups of keys with easy-to-learn locations as shown in the following diagram.



1) Local hardware functions such as SETUP and INIT (reset the "terminal" to its initial state). These functions should be performed only in conjuntion with an additional key (e.g. ALT)

2) Support of local software like workstation control on PC 3270, windowing, setting attributes and colours as on the 3279 etc. The various functions should be envoked by normal, SHIFT, ALT and SHIFT-ALT, so that a clear grouping of the functions is possible.

3) Miscellaneous like DUP and FIELD MARK and (only in conjunction with ALT) PA1..PA3

4) Mode setting, including INSERT, DELETE. INSERT should work as a flip-flop as on the PC.

To have a more unique keyboard for different applications (PC, 327x, ...) the mode keys should activate the function of some other keys and the behaviour of the "workstation", for example:

- normal mode
- character set 0 (normal) or 1(special graphics) is active
- character set 1 may be APL
- real graphics mode (needs an application)
- etc.

5) Terminal functions as used for 327x like ERASE EOF, ERASE INPUT etc. Dangerous functions like ERASE INPUT should be performed only in conjunction with an additional key (ALT)

6) Central typewriter functions. Keycaps of CR and LF as well as SHIFT need to be larger than those for the graphics. The small key caps of the PC let the fingers slip between the keys. Keys left or right of 1-Q-A-Z at left and BS-]-'-/ at the right are not easily reachable with normal sized hands.

National keyboards must have the same layout including "dead" keys as the national standards for typewriters. As on typewriters only the most used special characters (e.g. in Switzerland ï is used, but is not on the keyboard) are on own keys. The other specials are produced by "dead" keys.

An international or multilingual keyboard should allow the entering of all the special national graphics (see ISO 6937 part 2). This could be achieved by using ALT and SHIFT-ALT in conjunction with the normal keys. All the flying accents are defined preferably on the top row (1..0 - = USR1..USR3) - a total of 16 keys. This could work like this:

normal	SHIFT	ALT	SHIFT-ALT
a	A	æ	Æ
y	Y	ø	Ø
6	¬	¯	¯

Instead of "SHIFT LOCK" the function CAPS LOCK should be performed. This function also works for the special characters of ISO 6937 part 2 like æ Æ or ø Ø.

This function must be indicated by status line or LED.

7) Cursor control. Separate keys in a meaningful arrangement are necessary. Here again the PC is the worst case. More keys than shown in the sketch may be useful (e.g. "next page"). Here again the keys may serve several functions:

normal	SHIFT	ALT	SHIFT-ALT
->	-> >	-> word	-> sentence
home	bottom		

Of course some of these functions may be present only under certain software support (e.g. text applications on PC etc.). But the defined functions should be homogeneous between various applications!

8) Programmed function keys. The arrangement of PF keys should be geometric (3x4) to allow intuitive orientation (for CADAM there is a board of 32 PFK's in a more or less square arrangement operated by the left hand - imagine this in one row of keys!). A switch to the second set (PF13..PF24) should be performed by SHIFT.

In mode "numeric keypad" only PF1..PF3 are available. The numeric pad is (in our opinion) used only for data entry (where only few functions are necessary) or for few PC applications.

9) User-definable keys (USRx). These keys should allow a user to set up local code generation. When pressing one of these keys (with 3 keys a total of 12 meanings are possible) code as defined in a setup is sent to the host or used on the local workstation. This may be a single character or a sequence of characters. Only the total length of the sequences should be limited. And no restriction to the generated code should be made (except for conflicts with the communication).

National considerations

Our feeling is that the major emphasis, as stated above, has to be put on the need for a 'national standard area' very close to the national typewriter geography of each keyboard.

It's also necessary to have "dead" keys in order to merge accents with normal characters. A dead key, when typed on a normal typewriter, does not move to a new position, the mechanism stays in the same place, and so two (or more) characters can be merged. This is certainly a **need** in France and also in other countries.

International considerations

It's also mandatory to think of the international aspects of the work in a modern office. It's now often necessary to be able to enter new graphics into a letter, or other document, i.e. foreign characters, greek characters, or even mathematical symbols.

One could think of a scheme whereby ALT+a key in 6) generates characters in an alternate set - e.g. ALT-p would give a π. The meaning of this alternate set should be user selectable (greek, cyrillic, ...). A local hardware function should be available to show the keyboard layout for this alternate set. An ambitious scheme would be to

implement an actual display of the set on the key-tops of the characters (see IBM Technical Disclosure Bulletin, Vol 22, No 4).

As stated above, a set of free scratch keys could give this kind of flexibility to the keyboard without changing the customs of the national user, provided he has his usual standard keyboard.

What is SEAS?

SEAS - SHARE European Association is an independent organization of computer users and computing centres with members in Europe, the Middle East and Africa, who utilise or contemplate the use of large and medium IBM machines. SEAS works for the benefit of its membership by promoting the mutual exchange of information and experience and by liaising with IBM on matters of joint interest, in order to facilitate the effective use of computers.

The purposes of SEAS are:

To establish a forum for:

- exchange of know how,
- education in information processing techniques,
- development of improved information processing techniques,
- contributions to advances in the computer community and
- a dirct dialogue with IBM product development staff.

To represent the membership's interests by:

- influencing the development and improvement of products and services,
- communicatingn strategic concerns to IBM at the Executive Level and
- soliciting information about computer industry trends.

SEAS fulfills this mission through:

- The organization of meetings, which include an Anniversary Meeting and a Spring Meeting with programmes covering broad aspects of current and future data processing techniques.
- Project work on specific topics such as Office Automation, High Performance Processing, Interactive Computing, Operating Systems, Networking, Graphics.
- Publications such as the SEAS Newsletter, Proceedings of SEAS Meetings, Project White Papers.