SM 80

USER EXPERIENCES WITH COMMUNICATIONS

Michael Martin, Den Danske Bank af 1871 A/S, Copenhagen, DK

I have sub-titled this lecture "Getting IBM to talk Danish in the Danish Bank", in order to give a slightly better idea of which area of communications we are dealing with. At the Danish Bank we have used a not inconsiderable effort in "finishing off" a range of IBM software products. The reasons for which we have had to do this can be classified into three categories:

(1) Danish alphabet
(2) different types of printers installed
(3) Danish language

The problems will be explained for the uninitiated, and I will finish off by trying to demonstrate how much effort we have expended on solving these problems.

(1) Danish alphabet

It may surprise some that not all alphabets have 26 letters, where the last is Z. The Danish alphabet has 28 letters, or 29 for those who view W and V as being distinct from one another. There are three letters in addition to the English alphabet

- Æ, Ø, Å

These letters are placed at the end of the alphabet, i.e. after the letter Z. In addition, Danish has the following peculiarities:

- Å can be replaced by Aa, at the whim of the writer
- V and W are regarded as identical, W being a borrowed letter as far as Danish is concerned

Thus a Danish telephone directory starts with entries under Ab...., and not Aa.... as might be expected. (Aa.... entries are treated as Å.... entries, which are found at the end of the directory). For similar reasons, V.... entries are mixed together with W.... entries (there is no separate section for names beginning with W).

To further confuse matters, there are a number of people of Swedish and German extraction living in Denmark, whose names include the letters

- Ä, Ö, Ü or ä, ö, ü

It is the custom in Denmark to spell these names correctly, especially as far as a service organisation such as ours is concerned.

The problems associated with the alphabet as far as the computer installation is concerned, fall mainly into the following categories:

- character translation
- different versions of EBCDIC
- hardware restrictions
- sorting sequence

Character translation is commonly performed in many software products to

- translate from lowercase to uppercase
- remove invalid characters before either transmission or printing

The problem is that <u>all</u> IBM software products performing such translation either neglect to translate the Danish letters, or remove them as being invalid. As there is no centralised facility for character translation, this single problem turns out to be very time consuming for the installation. The problem cannot be ignored, as results of incorrect translation are unacceptable for both our internal users and our customers.

EBCDIC representations are also a problem for the installation, because there are three versions of EBCDIC in existence:

- word-processing (EBCDIC-WP)
- data-processing (EBCDIC-DP): standard interface code
- EBCDIC-DP: alternate interface code

The representations of Æ, Ø and Å are different between EBCDIC-WP and EBCDIC-DP. Within the EBCDIC-DP code, there are two valid representations for each of æ, ø and å (standard or alternate I/O interface code). Needless to say, we have terminals installed using each of the three EBCDIC codes.

Hardware restrictions can be illustrated by two examples from our installation:

- when wishing to enter the characters Ä, Ö or Ü, users have to use the keys marked <, " or ? respectively. (There is actually a key marked ü on the Danish 3278 keyboard, but this gives an incorrect result)
- in considering the installation of a 6670 laser printer, Danish users have to be satisfied with 2 concurrent fonts, whereas Americans can have 4.

Finally, the EBCDIC sorting sequence is different from that of the alphabet, and cannot be used in any reasonable way for applications requiring ascending alphabetic sequence. There is another more principal problem, in that IBM sorting techniques do not recognise the equivalence of single-character elements with multiple-character elements, as is the case with Aa and Å, or aa and å.

(2) Printer variety

As is the case in many larger installations, our users have access to a number of different types of printer, depending on the nature of the job to be done. The following list shows some of the IBM printer models we have installed:

- 3203
- 3211
- 3287 / 3289
- 3610 / 3611
- 6640

A user will tend to prefer one printer model to another, depending on various changing criteria - e.g.

- quantity (3211)
- quality (6640)
- turn-around (3287)
- uppercase/lowercase
- forms availability

The IBM supplied parameters to aid the user in selecting a printer and forms are:

- SYSOUT class (e.g. test, production, microfiche)
- forms number (pre-printed forms identification)
- FCB id. (printer carriage control)
- destination code (e.g. local, RJE station id., VTAM printer id.)

These should be perfectly adequate for the majority of installations. However, a quick survey of some of the IBM software products we have installed reveals that support for these parameters is somewhat random. Table 1 shows a list of 11 functions/products, all of which except DYNALLOC are commonly used. The most surprising revelation in this table is perhaps that the TSO LIST product (for which the installation has to pay IBM) has no support for any of the parameters!

These inconsistencies in support mean that the user has to remember which facilities are available to him in his current environment. For example, terminal printers running under an external writer (i.e. not supported by JES2) have their output selected by SYSOUT class, whereas true RJE printers use the destination parameter (DEST). It must be said that the problem is no more than irritating for the user, who must remember "SYSOUT" for this printer, "DEST" for that printer. Nonetheless, it does restrict the installation from obtaining the full function of the available parameters. In the example quoted, there is a limit of 36 SYSOUT classes, which can easily be exhausted in practice if one SYSOUT class has to be allocated to each terminal printer. By comparison, the range of DEST codes available is virtually unlimited (1-8 character name).

The conclusion for the installation is that it must

- either restrict itself to a sub-set of the available functions
- or it must expend effort to develop a more long term solution to these problems

(3) <u>Danish language</u>

The IBM language problem is well-known in Europe, but it should be stressed that this particular discussion is not about whether or not IBM technical manuals should be available to us in Danish. Our problem is that we have installed, and will certainly continue to install, software products which conduct a terminal conversation with an end-user. These users are accustomed to working with Danish non-technical messages, and are suddenly exposed to technical dialogues in the English language. (Two examples: retrieving print output from batch at a VSPC terminal; using ADF).

This problem is compounded by the poor structure of many of the software products involved. Messages are frequently imbedded arbitrarily in the programs which issue them, and are difficult for the installation to modify.

Another type of language problem where we feel badly done by, is exemplified by SCRIPT. This otherwise excellent product has facilities for hyphenation and spelling verification, but there is neither a Danish dictionary available, nor are the rules of hyphenation in SCRIPT (i.e. English) the same as in Danish. Thus a Danish installation gets less for its money than, say an American one.

Conclusion

The problems we have described are some of those we have encountered in the area of printing the Danish language at our IBM installation. Some of the problems are completely general for all IBM users, and others must certainly be problems in most European countries outside the U.K.

We have not attempted to solve anything like all of the problems, but have only concentrated on those which we felt were really important. Most of the problems which we have solved, have been difficult to solve. They have usually involved (expensive!) systems programming staff in considerable research to find out how to solve the problem. The solutions implemented have on the whole been trivial (e.g. modifications to translate tables), but have nonetheless been non-standard - i.e. they have involved direct modification of IBM supplied software, and have not been implementable through exit routines, or the like.

Table 2 shows which products have been modified (or implemented), including the approximate numbers of lines of coding required for each change. I wonder how many installations round the world are expending similar effort on the same problems? In presenting this material, I hope to provoke a response from other installations and IBM, to ensure that this duplication of effort will not be necessary in the future.

Product	Parameters supported			
	SYSOUT class	Forms no.	FCB	DEST
J.C.L.	x	x	x	x
JES2 op. cmd.	-	-	-	x
TSO OUT cmd.	x	-	-	x
TSO/SPF OUTLIST	x	-	-	-
SCRIPT (TSO)	x	-	x	x
TSO ALLOC	x	-	-	x
DYNALLOC	x	x	x	x
JES2 ext. wtr.	x	-	-	-
TSO/SPF general	x	-	-	-
TSO LIST (!)	-	-	-	-
TSO PRINT	½	-	-	½

Table 1: incompatibilities in support of printer parms

Product	no. lines of mods	non-std	problem category
MVS DIDOCS (op console)	3	x	1
JES2	180	x	1
CICS	170	x	1
IMS	3	x	1
TCAM	170	x	1
TSO/VTAM (TPUT)	11	x	1
TSO/VTAM	60		1
IIS (3601)	90		1
SCRIPT	10	x	1
SCRIPT aux. processor	644		2
SCRIPT Danish dictionary	growing!		3

Table 2: extent of modifications to standard IBM products