

SERIAL NUMBER					TEXT	TYPE/ MIN.	STA- TUS
SPLIT	SPLIT	SPLIT	SPLIT	SPLIT			
= 0	= 70	= 74	= 78	= 81			
202	-	-	-	27	Local elections 1981, listaggrega- tiongroup 2. Valid votes cast	I6	0
203	-	-	-	28	Local elections 1981, listaggrega- tiogroup 2. Perso- nal votes cast	I5	0
204	-	-	-	29	Local elections 1981, listaggrega- tiongroup 2. Number of elected candidates	I2	0
205	-	-	-	30	Local elections 1981, listaggrega- tiongroup 2. Number of elected female candidates	I2	0
206	-	-	-	31	Local elections 1981, listaggrega- tiongroup 2. Mayor's office, yes=1, no=0	I1	0

SERIAL NUMBER					TEXT	TYPE/ MIN.	STA- TUS
SPLIT	SPLIT	SPLIT	SPLIT	SPLIT			
= 0	= 70	= 74	= 78	= 81			
207	-	-	-	32	Local elections 1981, listaggrega- tiongroup 3. Number of lists in aggrega- tion	I2	0
208	-	-	-	33	Local elections 1981, listaggrega- tiongroup 3. Number of candidates put up	I3	0
209	-	-	-	34	Local elections 1981, listaggrega- tiongroup 3. Number of female candidates put up	I2	0
210	-	-	-	35	Local elections 1981, listaggrega- tiongroup 3. Valid votes cast	I6	0
211	-	-	-	36	Local elections 1981, listaggrega- tiongroup 3. Perso- nal votes cast	I5	0
212	-	-	-	37	Local elections 1981, listaggrega- tiongroup 3. Number of elected candidates	I2	0

SERIAL NUMBER					TEXT	TYPE/ MIN.	STA- TUS
SPLIT	SPLIT	SPLIT	SPLIT	SPLIT			
= 0	= 70	= 74	= 78	= 81			
213	-	-	-	38	Local elections 1981, listaggrega- tiongroup 3. Number of elected female candidates	I2	0
214	-	-	-	39	Local elections 1981, listaggrega- tiongroup 3. Mayor's office, yes=1, no=0	I1	0
215	-	-	-	40	Local elections 1981, listaggrega- tiongroup 4. Number of lists in aggrega- tion	I2	0
216	-	-	-	41	Local elections 1981, listaggrega- tiongroup 4. Number of candidates put up	I3	0
217	-	-	-	42	Local elections 1981, listaggrega- tiongroup 4. Number of female candidates put up	I2	0
218	-	-	-	43	Local elections 1981, listaggrega- tiongroup 4. valid votes cast	I6	0

SERIAL NUMBER					TEXT	TYPE/ MIN.	STA- TUS
SPLIT	SPLIT	SPLIT	SPLIT	SPLIT			
= 0	= 70	= 74	= 78	= 81			
219	-	-	-	44	Local elections 1981, listaggrega- tiongroup 4. Perso- nal votes cast	I5	0
220	-	-	-	45	Local elections 1981, listaggrega- tiongroup 4. Number of elected candidates	I2	0
221	-	-	-	46	Local elections 1981, listaggrega- tiongroup 4. Number of elected female candidates	I2	0
222	-	-	-	47	Local elections 1981, listaggrega- tiongroup 4. Mayor's office, yes=1, no=0	I1	0
223	-	-	-	48	Local elections 1981, listaggrega- tiongroup 5. Number of lists in aggrega- tion	I2	0
224	-	-	-	49	Local elections 1981, listaggrega- tiongroup 5. Number of candidates put up	I3	0

SERIAL NUMBER					TEXT	TYPE/ MIN.	STA- TUS
SPLIT	SPLIT	SPLIT	SPLIT	SPLIT			
= 0	= 70	= 74	= 78	= 81			
225	-	-	-	50	Local elections 1981, listaggrega- tionsgroup 5. Number of female candidates put up	I2	0
226	-	-	-	51	Local elections 1981, listaggrega- tiongroup 5. Valid votes cast	I6	0
227	-	-	-	52	Local elections 1981, listaggrega- tiongroup 5. Perso- nal votes cast	I5	0
228	-	-	-	53	Local elections 1981, listaggrega- tiongroup 5. Number of elected candidates	I2	0
229	-	-	-	54	Local elections 1981, listaggrega- tiongrup 5. Number of elected female candidates	I2	0
230	-	-	-	55	Local elections 1981, listaggrega- tiongrup 5. Mayor's office, yes=1, no=0	I1	0

SERIAL NUMBER					TEXT	TYPE/ MIN.	STA- TUS
SPLIT	SPLIT	SPLIT	SPLIT	SPLIT			
= 0	= 70	= 74	= 78	= 81			
231	-	-	-	56	Local elections 1981, listaggrega- tiongroup 6. Number of lists in aggrega- tion	I2	0
232	-	-	-	57	Local elections 1981, listaggrega- tiongroup 6. Number of candidates put up	I3	0
233	-	-	-	58	Local elections 1981, listaggrega- tiongroup 6. Number of female candidates put up	I2	0
234	-	-	-	59	Local elections 1981, listaggrega- tiongrup 6. Valid votes cast	I6	0
235	-	-	-	60	Local elections 1981, listaggrega- tiongroup 6. Perso- nal votes cast	I5	0
236	-	-	-	61	Local elections 1981, listaggrega- tiongrup 6. Number of elected candidates	I2	0

SERIAL NUMBER					TEXT	TYPE/ MIN.	STA- TUS
SPLIT	SPLIT	SPLIT	SPLIT	SPLIT			
= 0	= 70	= 74	= 78	= 81			
237	-	-	-	62	Local elections 1981, listaggrega- tiongroup 6. Number of elected female candidates	I2	0
238	-	-	-	63	Local elections 1981, listaggrega- tiongroup 6. Mayor's office, yes=1, no=0	I1	0

### 9.1. Program 'FORKODE'

In a sense the establishment of a statistical data archive is a never terminating endeavour. Large amounts of data are available on the commune level which are not included in this archive and more data will of course appear in the future. Especially this archive will be obsolete if not future elections and censuses are added to the existing information.

From a technical point of view it is a rather simple task to add new variables to the existing archive. Using a SPSS-version of the archive the ADD VARIABLES will highly facilitate the endeavour (See Nie et al., pp. 172-173). Nevertheless it can be a cumbersome enterprise to enlarge the archive in practice. Firstly, it is inconvenient to code statistical variables in fixed format owing to the varying number of digits across the units. Secondly, the number and order of units must exactly match that of the archive. Thirdly, it is difficult to control, that no coding errors have occurred.

Therefore, three programs have been developed which aim at facilitating the different operations involved in creating and adding new variables to the archive:

Firstly, the program 'FORKODE' creates a file containing unit identifications and names for all or a subset of units in the archive. In that way, the user does not need to code the unit identifications himself and another serious source of coding errors is removed.

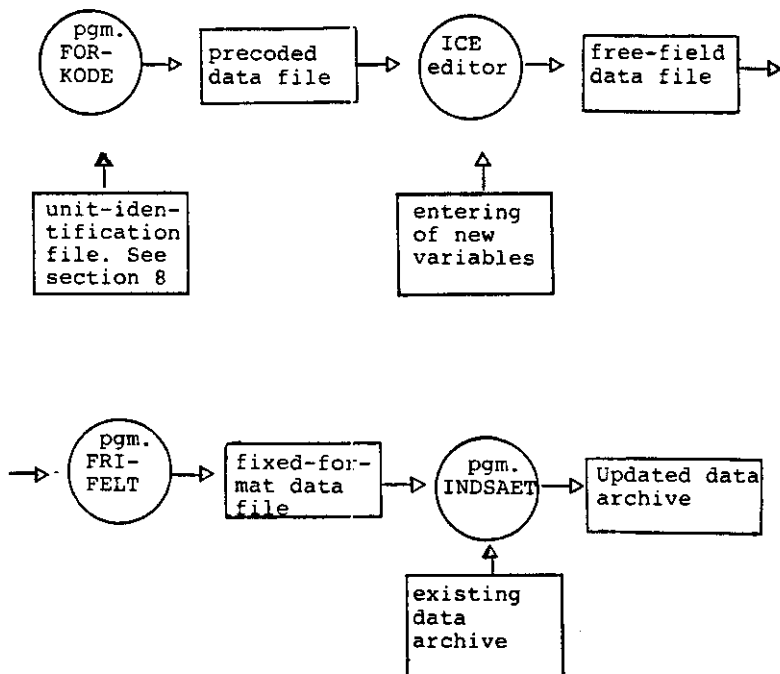
Secondly, the user can apply the ICE-editor to type the desired variable values on the above mentioned file in a convenient free-field format.

Thirdly, the program 'FRIFELT' in the following subsection converts the file into fixed-format positions of the variables. Furthermore the variables are summed up and the results printed inside user-specified groups of units. This procedure highly facilitates control of eventual typing errors.

Fourthly, the user can apply the program 'INDSAET', described in subsection 11, to insert the new variables in the existing archive.



We can summarize the procedure in the following figure:



Following the general overview above we enter upon a more detailed description of the working of program 'FORCODE'.

## 9.2. INPUT FILES

TAPE1=Z7OAOOA

A standard unit identification file of which a transcript is given in section 5. It has the following format:

variable name	format	line- number	position
UNID1	I6	1	1 - 6
UNID2	R4	1	8 - 11
UNID3	R4	1	12 - 15
UNID4	R4	1	16 - 19
UNID5	R4	1	20 - 23
UNID6	R4	1	24 - 27
UNID7	R4	1	28 - 31
UNID8	R4	1	32 - 35
UNID9	R2	1	36 - 37
UNID10	I6	1	39 - 44
UNID11	I6	1	46 - 51
UNID15	I3	1	53 - 55
UNID16	I4	1	57 - 60

### 9.3. OUTPUT-FILES

#### TAPE2

An output data-file. It contains precoded informations for unit-identifications and -names. From 1 to 5 lines per unit are produced depending on MODIFY instructions.

### 9.4 MODIFY INSTRUCTIONS

The program 'FORKODE' is stored as a MODIFY source deck, where OPL=U99EO1A. MODIFY is called in order to create a version of program 'FORKODE' on file COMPILE for later compilation and execution.

.

.

GET,U99EO1A. (MODIFY program library on indirect file U99EO1A)  
 MODIFY,P=U99EO1A,U. (MODIFY writes a version of program library  
 on COMPILE file)

-EOR-

\*DECK FORKODE

Only deck FORKODE is written on COMPILE file. Required.

\*IDENT KONTROL

All modifications are identified by linename KONTROL. Optional.

\*INSERT 11

LINANT=V

V equals number of lines per unit written on TAPE2. A maximum number of 5 lines per unit can be requested. Default value:

1. Optional.

\*INSERT 12

TIDSPKT=X

X equals a 6 digits date. 1-2 position = two last digits of year. 3-4 position = month. 5-6 position = day. For example X=811206 means december 6th, 1981. Only those units existing at the defined date will be selected. If X is set to 999999, all units will be selected irrespectively of existence at a given moment. Default value: 999999. Optional.

\*INSERT 13

SUBKOM=Y

If Y is set to 1 both commune and sub-commune units will be written on TAPE2. If Y is set to 0, only commune units will be written on TAPE2. Default value: 0. Optional.

\* INSERT 14

KONTROL=Z

If Z is set to 1 the 20 first and 20 last lines of TAPE2 will be printed on the OUTPUT-file. If Z is set to 0, no lines will be printed. Default value: 1. Optional.

#### 9.5. SORTMRG

The unit identification file Z70A00A is sorted in ascending order on variable UNID1. It means that subcommunal units are placed immediately after the corresponding communal unit and communes are sorted first according to county number and finally in alphabetic order. If the communes are not placed in the same order in the statistical publications, the coding procedure will be unnecessarily impeded. In that case it will be convenient to alter the unit order of Z70A00A by sorting the file. The SORTMRG utility program is available for that purpose <sup>1)</sup>.

SORTMRG is intended to sort records or lines. If an unit goes over several lines severe problems arise. Under certain conditions the problem may be solved by means of the CYBER RECORD MANAGER<sup>2)</sup>. All standard files in the archive have been written with a FORTRAN formatted WRITE procedure. Consequently, they are all Z-type records <sup>3)</sup>. A type Z record consists of a multiplum of words each of which contains 10 characters. End of record is marked by a double-zero in the two last character positions of the last character-filled word on the record. If the characters fill nine or ten positions a new word is added ended by a double zero. Since all variables are right positioned and are given at least a zero value, it is possible to calculate the exact number of words representing 10 characters on each record or line.

- 
1. CDC-60343900: Sort/Merge version 4 and 1, Reference Manual
  2. CDC-60495800: Cyber Record Manager, Basic Access Methods, Version 1.5, User's guide
  3. CDC-60305600: Fortran Extended, Version 4, Reference Manual, p. III-5-2.

The individual Z-type records will often be of different size, but the total length of a number of records constituting a user defined unit will always be the same from unit to unit. The property can be used to treat the Z-type records as F-type records. F-type records always contain the same number of characters in every record without any special end-of-record-mark.

By defining the number of characters in the F-type record equal to the total number of characters in the user defined unit consisting of several Z-type records, SORTMRG can be able to sort over several Z-type records.

Perhaps an example make the reasoning more clear. Section 2.6. gives a format specification of the standard data archive.

Interpreted as a Z-type file, each unit consists of a large number of records, the number of which equal the number of slashes plus one (/) in the format statement, i.e. 127 records or lines per unit. Interpreted as a F-type file, all 127 lines can be handled as one record of a 9370 characters length.

The application of Z-type and F-type records in connexion with SORTMRG will be demonstrated below.

#### 9.6. BATCH JOB EXAMPLE

As an example we will show, how the 1984 European election was coded and inserted in the archive. In the statistical reports the communes are placed according to nomination districts and constituencies, an order different from the standard order of Z70A00A. Using the SORTMRG utility program, we can sort Z70A00A according to the sorting order of communes in the statistical reports. It will highly facilitate the coding procedure. In the following example it is assumed, that all input and output files are available as direct or indirect files on mass storage, and that the batchjob is submitted from a terminal:

```
/JOB
jobname, CM650000.
USER, usernumber, password.
CHARGE,chargernumber,projectnumber.
ATTACH,Z70A00A.(unit identification file)
FILE,Z70A00A,RT=Z,FL=80,BT=C.
FILE,TAPE1,RT=Z,FL=80,BT=C.
SORTMRG.
GET,U99E01A. (modify program library)
MODIFY,P=U99E01A,U.
FTN,I.
LGO.
SAVE,TAPE2=B84A00A.
/EOR
SORT
FILE,INPUT=Z70A00A,OUTPUT=TAPE1
FIELD,KEY1(57,4,DISPLAY),KEY2(1,6,DISPLAY)
KEY,KEY1(A,COBOL6),KEY2(A,COBOL6)
END
/EOR
*DECK FORKODE
*IDENT KONTOL
*INSERT 11
      LINANT=:2
*INSERT 12
      TIDSPKT=840614
*INSERT 13
      SUBKOM=:1
```

\*INSERT 14

KONTROL=1

The program 'FORKODE' will produce a transcript of the first 20 and last 20 records of TAPE2=B84A00A as follows:

000100,1,KØBENHAVN>>  
000100,2>>  
000200,1,FREDERIKSBERG>>  
000200,2>>  
100600,1,GENTOFTE>>  
100600,2>>  
161700,1,ODENSE>>  
161700,2>>  
212600,1,ÅRHUS>>  
212600,2>>  
232600,1,ÅLBORG>>  
232600,2>>  
000101,1,2.0.,A. CHRISTIANSHAVN>>  
000101,2>>  
000102,1,2.0.,A. SYD+ØST>>  
000102,2>>  
000103,1,2.0.,A. VEST>>  
000103,2>>  
000104,1,3.0.,A. ÅRHUS>>  
000104,2>>  
000105,1,3.0.,A. NORD>>  
000105,2>>  
000106,1,3.0.,A. ØST>>  
000106,2>>  
000107,1,3.0.,A. VEST>>  
000107,2>>  
000108,1,4.0.,A. SUNDBY>>  
000108,2>>  
000109,1,4.0.,A. ØST>>  
000109,2>>  
000110,1,4.0.,A. VEST>>  
000110,2>>  
000111,1,5.0.,A. BLÅGÅRD>>  
000111,2>>  
000112,1,5.0.,A. NORD>>  
000112,2>>  
000113,1,5.0.,A. SYD>>  
000113,2>>  
000114,1,9.0.,A. ÅMAGERBRO>>  
000114,2>>

Σ

```

230200,1,BROVST>>
230200,2>>
230600,1,FJERRITSLEV>>
230600,2>>
231500,1,LØKKEN-VRA>>
231500,2>>
231800,1,PANDRUP>>
231800,2>>
230900,1,HALS>>
230900,2>>
232500,1,ÅBYERO>>
232500,2>>
232601,1,5.0. ÅLBORG-NORD (ÅLBORG-DEL)>>
232601,2>>
232602,1,6.0. ÅLBORG-VEST>>
232602,2>>
232603,1,7.0. ÅLBORG-ØST>>
232603,2>>
230100,1,ARDEN>>
230100,2>>
230800,1,HADSUND>>
230800,2>>
231200,1,HOBRO>>
231200,2>>
231900,1,SEJLFLØD>>
231900,2>>
232200,1,SKØRPING>>
232200,2>>
230500,1,FARØ>>
230500,2>>
231400,1,LØGSTØR>>
231500,2>>
231600,1,NIBE>>
231600,2>>
231700,1,NØRAGER>>
231700,2>>
232300,1,STØVRING>>
232300,2>>
232700,1,ÅRS>>
232700,2>>

```

The variables from the European election 1984 are supposed to be entered after the precoded information on each record

#### 9.7. ICE EDITOR

When variables are to be entered on TAPE2=B84A00A, the following conventions must be observed:

- a) Variables must be entered on the blank space after the double >>. Each line contains a maximum of 150 characters inclusive of the prewritten information. In the pro-



gram 'FORKODE' the user can request a maximum of 5 lines per case.

- b) Variables should be entered in free-field format. The variables must be separated by one or more special terminators. A special terminator is defined as a blank character, a special character such as / , . + ? except a colon (:) or a line-shift.
- c) If each case has more than one line, the number and order of lines must not be changed.
- c) The number and order of variables entered on each case must be the same for all cases, Therefore, a variable value missing for a given case must be entered as a zero value.
- e) The user can insert special lines of '999999' in the first six positions of the line. Such lines serve as separators of groups of cases. The variables values inside such groups can be summed up and printed for controlling purposes.

The ICE-editor<sup>1)</sup> is a very convenient tool in the process of entering new variables on TAPE2=B84A00A from an on-line terminal. An example will demonstrate the use of ICE:

GET,B84A00A

ICE,B84A00A

(ICE then prompts for a command line by typing two question marks (??) followed by a space at the beginning of the line:)

?? P1 (ICE prints the first floating line of B84A00A)

1 000100,1,KØBENHAVN>>

?? A/ (the user then types the variable value after the slash (/) according to the above mentioned conventions. Pressing the return key, ICE will append the typed information immediately after the double >> and print the updated line:)

1 000100,1,KØBENHAVN>>394850,206856,1242 774 204840 43676

(In that case the user has typed the first 6 variables of the European election 1984 in Copenhagen)

-----  
1) ICE, Reference Manual. RECAU-80-112-M.

?? LP (ICE prints the first line following the current line)  
 2 000100,2>>  
 ?? A/4108;;;28636 32005,,8169>68830+2529 3593 6775 5519  
 (return key)  
 2 000100,2>>4108;;;28636 32005,,8169>68830+2529 4593 6775 5519  
 ?? I/999999 (a group delimiter is inserted)  
 ?? LP  
 3 000200,1,FREDERIKSBERG>>  
 ?? (the user can then continue typing variables on the following  
 lines of B84A00A)  
 .  
 .  
 ?? ER (the user replaces the updated version of B84A00A  
 and exits form ICE)

The following contains a transcript of the first 20 and the  
 last 20 records of the updated version of B84A00A. On each  
 unit of two lines is typed 15 variables in free-field format  
 and a number of 999999 group separation lines are inserted:

```
000100,1,KØBENHAVN>>394850,206856,1242 774 204840 43676
000100,2>>4108;;;28636 32005,,8169>68830+2529 4593 6775 5519
999999
000200,1,FREDERIKSBERG>>73559 41367 353 143 40972 5539 1064
000200,2>>12056 4377 2360 11443 600 1248 1092 1157
999999
100600,1,GENØFTE>>53399 32700 207 48 32445 2525 1082 14070
100600,2>>1887 3291 5597 560 1769 507 1157
999999
161700,1,ODENSE>>130362.65992.65992.582.105.65305.15542.1947
161700,2>>14384.7578.3903.14119.1091.4162.955.1624
999999
212600,1,ÅRHUS>>191043.108038.809.204.107025.18613.2943.20299
212600,2>>14955.5879.30341.1573.6777.2927.2718
999999
232600,1,ÅLBORG>>119397.56691 407 132 56152 13916.1380.11058
232600,2>>6352.3519.11625.1286.4306.823.1887
999999
```

000101,1,2.0.,A. CHRISTIANSHAVN>>6669 3685 18 14 3653 456 85  
 000101,2>>462 508 101 1588 29 84 286 54  
 000102,1,2.0.,A. SYD-ØST>>15146 7485 52 26 7407 1563 139 905  
 000102,2>>1294 247 2619 85 165 213 177  
 000103,1,2.0.,A. VEST>>5744 2962 25 16 2921 639  
 000103,2>>49 280 510 94 1060 35 51 110 93  
 000104,1,3.0.,A. RADHUS>>8221 4261 25 9 4227 510 100 795 545  
 000104,2>>185 1479 38 140 299 136  
 000105,1,3.0.,A. NORD>>4235 1356 7 46 1303 277 41 518 59 69  
 000105,2>>115 18 164 13 29  
 000106,1,3.0.,A. ØST>>6865 3507 16 17 3474 367 92 846 415 234  
 000106,2>>1104 34 137 155 90  
 000107,1,3.0.,A. VEST>>7094 3798 29 12 3757 398 121 861 463  
 000107,2>>197 1214 62 146 183 112  
 000108,1,4.0.,A. SUNDBY>>9313 4740 19 26 4695 1173 92 608  
 000108,2>>782 170 1519 54 79 99 119  
 000109,1,4.0.,A. ØST>>4547 2518 14 7 2497 514 70 483 371 113  
 000109,2>>745 33 46 43 79  
 000110,1,4.0.,A. VEST>>14421 7050 32 16 7002 1496 133 952  
 000110,2>>1216 320 2301 98 142 166 178  
 000111,1,5.0.,A. BLÅGÅRD>>6117 3126 22 11 3093 399 52 306  
 000111,2>>455 86 1394 24 66 239 72  
 000112,1,5.0.,A. NORD>>4239 2350 17 9 2324 411 55 370 368  
 000112,2>>108 759 49 50 104 50  
 000113,1,5.0.,A. SYD>>5001 2529 15 9 2505 420 31 253 401 59  
 000113,2>>1039 31 36 163 72  
 000114,1,9.0.,A. AMAGERBRO>>14595 6991 51 24 6916 1655 98 720  
 000114,2>>1281 221 2420 57 122 174 168

Σ

230200,1,BROVST>>6370 3006 29 4 2973 517 94 509 136 277  
 230200,2>>309 283 635 18 195  
 230600,1,FJERRITSLEV>>6122 2996 12 4 2980 444 82 485 71 268  
 230600,2>>282 234 946 17 151  
 231500,1,LØKKEN-VRA>>6941 3240 24 1 3215 478 147 494 138 245  
 231500,2>>404 149 963 24 173  
 231800,1,PANDRUP>>7588 3393 28 3 3362 752 97 584 174 247  
 231800,2>>510 158 634 17 189  
 230900,1,HALS>>7593 3546 20 2 3524 680 113 561 212 230  
 230900,2>>612 183 173 25 195  
 232500,1,ÅBYBRO>>7740 3528 40 3 3485 723 96 599 253 300  
 232500,2>>438 149 731 20 176  
 232601,1,5.0. ÅLBORG-NORD (ÅLBORG-DEL)>>28967 13450 107 29  
 232601,2>>13317 3343 332 2427 1307 916 2638 363 1314 137 540  
 232602,1,6.0. ÅLBORG-VEST>>38313 19170 126 49 18995 4068 477  
 232602,2>>4785 1906 1311 3633 468 1433 307 607  
 232603,1,7.0. ÅLBORG-ØST>>52117 24068 174 54 23840 6505 571  
 232603,2>>3846 3139 12 5354 455 1559 379 740  
 230100,1,ARDEN>>6034 2841 26 11 2804 536 125 463 94 257  
 230100,2>>303 83 764 19 160  
 230800,1,HADSUND>>7730 3531 38 2 3491 115 790 172 239  
 230800,2>>400 77 616 15 129  
 231200,1,HOBRO>>10422 5304 56 8 5240 1195 189 1140 316 398  
 231200,2>>734 165 847 16 240  
 231900,1,SEJLFLOD>>6268 2957 22 3 2932 815 101 468 201 202  
 231900,2>>365 84 509 13 174

232200,1,SKØRPING>>7030 3513 33 7 3473 761 167 604 230 242  
232200,2>>427 89 745 28 180  
230500,1 FARSØ>>5974 2742 18 1 2723 365 112 499 64 258  
230500,2>>224 212 847 14 128  
231400,1,LØGSTØR>>8148 3870 33 10 3827 695 119 502 199 219  
231400,2>>462 165 1246 24 196  
231600,1,NIBØ>>5387 2490 20 1 2469 492 79 453 133 167  
231600,2>>288 78 647 12 120  
231700,1,NØRAGER>>4152 2138 22 3 2113 271 103 332 76 153  
231700,2>>149 190 692 6 141  
232300,1,STØVRING>>8173 3814 32 6 3776 654 135 730 195 294  
232300,2>>422 147 1002 33 164  
232700,1,ARS>>9093 4244 44 9 4191 639 152 773 139 290  
232700,2>>467 276 1253 14 188  
999999



### 10.1. PROGRAM 'FRIFELT'

The updated version of B84A00A containing variables in free-field format can't be read in any comprehensible way in standard programs of various kinds. The free-field format must be transformed into a fixed-format. The program 'FRIFELT' performs among other things that function:

- a) The program controls, that each updated record on B84A00A fulfills the formal requirement mentioned above.  
If the requirement is not met, the program prints an error message, saying that the record can't be properly interpreted. The user must then correct the record and try again.
- b) If the record fulfills the formal requirements, the free-field variables are transformed into a user specified fixed format and written upon an alternative file.
- c) Depending upon how the user has defined groups of records by means of 999999 lines, the program calculates and prints value sums of each variable inside the defined groups. This procedure highly facilitates the control of eventual typographical errors, since the statistical reports always present various aggregate values of the variables in question, especially on the county level and of course also as regards the whole country. Thereby a typing error can be identified inside a smaller group of units facilitating the exact identification and correction of the unit in error.

### 10.2. INPUT FILES

TAPE1=B84A00A

Input file of variables in free-field format

### 10.3. OUTPUT FILES

TAPE2=B84A00B

Output files of variables in user specified fixed-format

10.4. MODIFY INSTRUCTIONS

The program 'FRIFELT' is stored as a MODIFY source deck, where OPL=U99E01A. MODIFY is called in order to create a version of program 'FRIFELT' on file COMPILE for later compilation and execution:

```

.
.
GET,U99E01A. (MODIFY program library on indirect file U99E01A)
MODIFY,P=U99E01A,U. (MODIFY writes a version of program library
                     on COMPILE file)
.
-EOR-
*DECK FRIFELT
Only deck FRIFELT is written on COMPILE file. Required.
*IDENT KONTROL
All modifications are identified by linename KONTROL. Optional
*INSERT 9
      INTERGER VARA(T),VARB(T),VARC(T),VARD(T)
T equals number of variables in free-field format on TAPE1. Re-
quired.
*INSERT 10
      VARIANT=T
T equals number of variables in free-field format on TAPE1. Re-
quired.
*INSERT 11
      LINANT=U
U equals number of lines per unit on TAPE1. Default value: 1. Op-
tional.
*INSERT 12
      KOMNAVN=V
If V is set to 1, the name of the communal or sub-communal unit
will be written on TAPE2. The name has the format '3R10' and
will be positioned after the first identification variable.
Default value: 0. Optional.
*INSERT 13
      TOTSUM=X
If X is set to 1, the group structure of TAPE1 separated by
999999 lines will be ignored, and the variables will be summed up
to include all units on TAPE1. Default value 0. Optional.

```

\*INSERT 14

REWIND=2

If Z is set to 0, TAPE1 and TAPE2 will not be rewound after program execution, otherwise both files will be rewound.

Default value 1. Optional.

\*INSERT 351

7 FORMAT(XXX)

XXX represents a formatted FORTRAN statement for TAPE2. It consists of a number of integer variables (I) and eventually a number of alphanumeric variables (R). The first variable will always be the prewritten identification variable from TAPE1 and must be allocated at least 6 digits, i.e.

(I6). If KOMNAVN=1, then follows the unit name having the standard format (3R10). Finally the variables in free-field format come on TAPE1. They are to be given a fixed integer format on TAPE2. An example: 7 FORMAT(8I8,/,8I8). The format in question means, that there are 15 free-field variables on TAPE1, since the first variable is the identification variable. All variables are allocated 8 digits, unused digits are automatically blank filled from left.

The unit on TAPE2 consists of two records or lines, each of which fill 64 characters. KOMNAVN is set to 0. Required.

#### 10.5. BATCH JOB EXAMPLE

As an example, we will show how the insertion of free-field format variables from the 1984 European election on file B84A00A can be transformed into fixed format variables using the program 'FRIFELT'. The SORTMRG utility program is used to resort the units to the standard sorting order for the archive. It is assumed, that all files are available as direct or indirect permanent files on mass storage, and that the batch job is submitted from a terminal:

/JOB

jobname,CM60000.

USER,username,password.

CHARGE,chargenumber,projectnumber

GET,TAPE1=B84A00A. (variables in free-field format on B84A00A)

GET,U99E01A. (MODIFY program library on U99E01A)



MODIFY,P=U99E01A,U.

FTN,I.

LGO.

ATTACH,B84A00B/M=W. (formatted variables to be written on  
B84A00B)

FILE,TAPE2,RT=F,FL=140,BT=C.

FILE,B84A00B,RT=F,FL=140,BT=C.

SORTMRG.

/EOR

\*DECK FRIFELT

\*IDENT KONTROL

\*INSERT 9

INTEGER VARA(15),VARB(15),VARC(15),VARD(15)

\*INSERT 10

VARANT=15

\*INSERT 11

LINANT=2

\*INSERT 12

KOMNAVN=0

\*INSERT 13

TOTSUM=1

\*INSERT 351

7 FORMAT(8I8,/,8I8)

/EOR

SORT

FILE,INPUT=TAPE2,OUTPUT=B84A00B

FIELD,KEY1(1,8,DISPLAY)

KEY,KEY1(A,COBOL6)

END

/EOI

In the following is given a transcript of the resorted frist  
20 and last 20 units of B84A00B corresponding to the transcript of  
B84A00A on page A-9-11 to A-9-13:

100	394850	206856	1242	774	204840	43676	4108
28636	32005	8169	68830	2529	4593	6775	5519

100	394850	206856	1242	774	204840	43676	4108
28636	32005	8169	68830	2529	4593	6775	5519
101	6669	3685	18	14	3653	456	85
462	508	101	1588	29	84	286	54
102	15146	7485	52	26	7407	1563	139
905	1294	247	2619	85	165	213	177
103	5744	2962	25	16	2921	639	49
280	510	94	1060	35	51	110	93
104	8221	4261	25	9	4227	510	100
795	545	185	1479	38	140	299	136
105	4235	1356	7	46	1303	277	41
518	58	69	115	18	164	13	29
106	6865	3507	16	17	3474	367	92
846	415	234	1104	34	137	155	90
107	7094	3798	29	12	3757	398	121
861	463	197	1214	62	146	183	112
108	9313	4740	19	26	4695	1173	92
608	782	170	1519	54	79	99	119
109	4547	2518	14	7	2497	514	70
483	371	113	745	33	46	43	79
110	14421	7050	32	16	7002	1496	133
952	1216	320	2301	98	142	166	178
111	6117	3126	22	11	3093	399	52
306	455	86	1394	24	66	239	72
112	4239	2350	17	9	2324	411	55
370	368	108	759	49	50	104	50
113	5001	2529	15	9	2505	420	31
253	401	59	1039	31	36	163	72
114	14595	6991	51	24	6916	1655	98
720	1281	221	2420	57	122	174	168
115	10526	5512	43	19	5450	1313	100
549	1045	169	1874	36	94	141	129
116	9771	5403	37	24	5342	1112	122
866	790	238	1679	81	133	185	136
117	4597	2705	6	8	2691	834	58
392	341	130	767	42	46	33	48
118	16134	7955	36	26	7893	1744	173
1280	1163	331	2436	118	199	232	217
119	11106	6407	44	12	6351	927	180
1320	924	331	1891	91	214	298	175

Z

231100	25835	12445	97	29	12319	2500	415
2190	909	866	2057	542	2297	54	489
231200	10422	5304	56	8	5240	1195	189
1140	316	398	734	165	847	16	240
231300	1961	740	5	2	733	99	16
128	33	70	188	75	150	4	40
231400	8148	3870	33	10	3827	695	119
502	199	219	462	165	1346	24	196
231500	6941	3240	24	1	3215	478	147
494	138	245	404	149	963	24	173
231600	5387	2490	20	1	2469	492	79
453	133	167	288	78	647	12	120
231700	4152	2138	22	3	2113	271	103
332	76	153	149	190	692	6	141
231800	7588	3393	28	3	3362	752	97

584	174	247	510	158	634	17	189
231900	6268	2957	22	3	2932	815	101
468	201	202	365	84	509	13	174
232000	6998	3345	28	4	3313	531	101
545	113	230	339	274	993	14	173
232100	10251	3703	22	7	3674	756	70
832	227	257	759	121	452	32	168
232200	7030	3513	33	7	3473	761	167
604	230	242	427	89	745	28	180
232300	8173	3814	32	6	3776	654	135
730	195	294	422	147	1002	33	164
232400	13282	5998	59	6	5933	978	179
1084	285	492	661	476	1387	31	360
232500	7740	3528	40	3	3485	723	96
599	253	300	438	149	731	20	176
232600	119397	56691	407	132	56152	13916	1380
11058	6352	3519	11625	1286	4306	823	1887
232601	28967	13453	107	29	13317	3343	332
2427	1307	916	2638	363	1314	137	540
232602	38313	19170	126	49	18995	4068	477
4785	1906	1311	3633	468	1433	307	607
232603	52117	24068	174	54	23840	6505	571
3846	3139	1292	5354	455	1559	379	740
232700	9039	4244	44	9	4191	639	152
773	139	290	467	276	1253	14	188

### 11.1. PROGRAM 'INDSAET'

The final problem concerns how to insert the new variables into the existing archive. For example it is possible to use the ADD VARIABLES facility in SPSS, but it requires, that the number and the order of units are the same on both sets of data. A special program is available, which doesn't require the same number of units in both sets of data. It means, that more complicated problems of merging files can be solved, such as merging the archive with survey-data.

In order to use the program 'INDSAET' to merge two files, they must be compatible in three ways:

- a) A similar unit-identification variable must be the first variable on both sets of data.
- b) Both sets of data must be sorted in ascending order on the first unit-identification variable.
- c) One of the two data sets (TAPE1) is considered the primary file, which means that a given unit-identification on the secondary file (TAPE2) must always be found on the primary file too, while the opposite doesn't necessarily have to be the case. In the last case, where the number of units are different in the two sets of data, the user can choose either to exclude units or insert zero-filled variables in the merged file.

### 11.2. INPUT FILES

#### TAPE1

The primary input data-file.

#### TAPE2

The secondary input data-file. Unit-identifications on TAPE2 must be existing on TAPE1 also.

11.3. OUTPUT-FILESTAPE3

Merged output-file of TAPE1 and TAPE2. The number of units and order of variables depend on MODIFY-instructions.

TAPE4

An output data-file. TAPE4 contains a short transcript of TAPE3 and is copied on the print output-file by COPYSBF,TAPE4.

11.4. MODIFY INSTRUCTIONS

The program 'INDSAET' is stored as a MODIFY source deck, where OPL=U99E01A. MODIFY is called in order to create a version of program 'INDSAET' on file COMPILE for later compilation and execution.

.

GET,U99E01A. (MODIFY program library on indirect file U99E01A)  
 MODIFY,P=U99E01A,U. (MODIFY writes a version of program library  
 on COMPILE file)

.

-EOR-

\*DECK INDSAET

Only deck INDSAET is written on COMPILE file. Required.

\*IDENT KONTROL

All modifications are identified by linename KONTROL. Optional

\*INSERT 12

INTEGER GVARA(R),GVARB(S),NVARA(T),NVARB(U),  
 +NULVARA(T),NULVARB(U)

The arrays GVARA and GVARB comprise the variables on TAPE1 minus the first unit identification variable. Consequently R + S equal the number of variables on TAPE1 minus 1. The arrays NVARA and NVARB contain the variables on TAPE2 minus the first unit identification variable. Consequently T + U equal the number of variables

on TAPE2 minus 1. The arrays NULVARA and NULVARB correspond to NVARA and NVARB. Not all arrays have to be defined. It depends on the following options. Required.

\*INSERT 14

VARANTA=T

VARANTB=U

T and U equal T and U in former statement. If array NULVARA is defined VARANTA must be set. If array NULVARB is defined VARANTB must be set too. Conditional.

\*INSERT 15

POSIT=V

The value of V determines, in what order the variables on TAPE1 and TAPE2 will be merged on TAPE3.

Value of V	order of variables on TAPE1	order of variables on TAPE2	order of variables on TAPE3
---------------	--------------------------------	--------------------------------	--------------------------------

1	IDENTA <sup>1)</sup> + GVARA(R)	IDENTB <sup>2)</sup> + NVARA(T)	IDENTA + GVARA(R) + NVARA(T) or NULVARA(T)
---	------------------------------------	------------------------------------	---

Variables on TAPE2 will be appended after variables from TAPE1 on TAPE3. Arrays GVARB and NVARB shall not be defined

2	IDENTA + GVARA(R) + GVARB(S)	IDENTB + NVARA(T)	IDENTA + GVARA(R) + NVARA(T) or NULVARA(T) + GVARB(S)
---	------------------------------------	----------------------	---

Variables on TAPE2 will be inserted between GVARA and GVARB from TAPE1 on TAPE3. Array NVARB shall not be defined

3	IDENTA + GVARA(R)	IDENTB + NVARA(T)	IDENTA + NVARA(T) or NULVARA(T) + GVARA(R)
---	----------------------	----------------------	---

Variables on TAPE1 will be appended after variables from TAPE2 on TAPE3. Arrays GVARB and NVARB shall not be defined

---

4	IDENTA + GVARA(R)	IDENTB + NVARA(T) + NVARB(U)	IDENTA + NVARA(T) or NULVARA(T) + GVARA(R) + NVARB(U) or NULVARA(U)
---	----------------------	------------------------------------	--

Variables on TAPE1 will be inserted between NVARA and NVARB from TAPE2 on TAPE3. Array GVARB shall not be defined

---

- 1) IDENTA = unit identification variable on TAPE1
- 2) IDENTB = unit identification variable on TAPE2

Default value of V is 1. Optional.

\*INSERT 16

OPTION=X

The value of X determines how the units on TAPE1 and TAPE2 will be merged on TAPE3.

If X = 1. It is assumed, that TAPE1 and TAPE2 contain the same number of units. Consequently, the units will be written on TAPE3 without modifications. The arrays NULVARA and NULVARB shall not be defined.

If X = 2. TAPE1 will eventually contain units which do not exist on TAPE2. The units in question will be written on TAPE3, and the non-existent variables on TAPE2 will be inserted as zero-value variables. If POSIT equal 1, 2 or 3, the array NULVARA must be defined, and if POSIT equals 4, the arrays NULVARA and NULVARB must be defined. the two arrays represent zero-value variables to be inserted on TAPE3. Choosing this option, no information will be lost, and TAPE3 will contain the same number of units as TAPE1.

If X = 3. TAPE1 will eventually contain units which do not exist on TAPE2. The units in question will not be written on TAPE3. Consequently, the number of units on TAPE3 and TAPE2 will be the same. X The arrays NULVARA and NULVARB shall not be defined.

If  $X = 4$ . This option is specially suited for merging the archive with survey data. Since the statistical data will serve as contextual variables in the survey, many units in the survey will probably have the same unit identification, representing a group of respondents inside a commune unit. Moreover there will certainly not be respondents from all commune units. The program assumes, that the data archive is on TAPE1, and the survey on TAPE2. The survey must have been arranged in such a way, that the first variable represent the geographical unit in which the respondent in question resides corresponding to the unit identification in the data archive. Moreover the respondents must have been sorted in ascending order on that variable. Variables from TAPE1 will then be repeated, together with variables from TAPE2 for units on TAPE2 having the same unit identification. The number of units on TAPE3 and TAPE2 will be the same. If the user wish the statistical variables to be placed after the survey variables, POSIT = 4 must be used. The arrays NULVARA and NULVARB shall not be defined.

Default value of X is 1. Optional.

\*INSERT 17

1 FORMAT( )

Format statement for variables per unit on TAPE1. The number of variables equals  $1 + R + S$ , where 1 represents IDENTA.

\*INSERT 18

2 FORMAT( )

Format statement for variables per unit on TAPE2. The number of variables equals  $1 + T + U$ , where 1 represents IDENTB.

\*INSERT 19

3 FORMAT( )

Format statement for variables per unit on TAPE3. The number of variables equals  $1 + R + S + T + U$ , where 1 represents IDENTA.



Since IDENTB is not written on TAPE3, the format statment on TAPE3 must define 1 variables less than on TAPE1 plus TAPE2.

\*INSERT 21

REWIND=Y

If Y is set to 0, TAPE1 and TAPE2 will not be rewound after program execution, ctherwise both files will be rewound. Default value 1. Optional.

\*INSERT 22

KONTROL=Z

If Z is set to 0, a short control transcript of TAPE3 will not be written on TAPE4. Otherwise a transcript will be written. Default value 1. Optional.

#### 11.5. BATCH-JOB EXAMPLE

As an example, we will show how the file B84A00B containing formatted variables for 1984 European election, see section 10.5. can be merged with an edition of the data-archive. It is assumed, that all files are available as direct or indirect permanent files on mass storage, and that the batchjob is submitted from a terminal:

/JOB

jobname,T200.

USER,username,password.

CHARGE,chargenumber,projectnumber.

ATTACH,TAPE1=P84A11A (version of the data-archive)

ATTACH,TAPE2=B84A00B. (formatted 1984 European Election)

ATTACH,TAPE3=P84A12A/M=W. (updated data-archive)

GET,U99E01A. (MODIFY program on file U99E01A)

MODIFY,P=U99E01A,U.

FTN,I.

LGO.

COPYSBF,TAPE4.

/EOR

The result, TAPE3=P84A12A represents a standard version of the data-archive exclusive of local elections.

\*DECK INDSAET  
 \*IDENT KONTROL  
 \*INSERT 12

INTEGER GVARA(367),NVARA(15),NULVARA(15)

\*INSERT 14  
 VARANTA=15

\*INSERT 15  
 POSIT=1

\*INSERT 16  
 OPTION=2

\*INSERT 17

1 FORMAT(18,2X,7A4,A2,4I8,/,5I8,/,	(UNID1 to UNID18)
+9I8,/,4I8,/,	(CELA1 to CELA13)
+9(9I8,/,1,2I8,/,	(CENA1 to CENA83)
+9I8,/,4I8,/,	(GELA1 to GELA13)
+8I8,/,	(REFA1 to REFA8)
+9I8,/,5I8,/,	(GELB1 to GELB14)
+9I8,/,6I8,/,	(CELB1 to CELB15)
+9I8,/,5I8,/,	(GELC1 to GELC14)
+6(9I8,/,1,3I8,/,	(CENB1 to CENB57)
+9I8,/,7I8,/,	(GELD1 to GELD16)
+2(9I8,/,1,I8,/,	(CELC1 to CELC19)
+6I8,/,	(REFB1 to REFB6)
+9I8,/,7I8,/,	(EURA1 to EURA16)
+9I8,/,9I8,/,	(GELE1 to GELE18)
+9I8,/,9I8,/,2I8,/,	(CELD1 to CELD20)
+2(9I8,/,1,I8,/,	(GELF1 to GELF19)
+9I8,/,9I8,/,I8)	(GELG1 to GELG19)

\*INSERT 18

2 FORMAT(3I8,/,8I8) (IDENTB + EURB1 to EURB15)

\*INSERT 19

3 FORMAT(18,2X,7A4,A2,4I8,/,5I8,/,	(UNID1 to UNID18)
+9I8,/,4I8,/,	(CELA1 to CELA13)
+9(9I8,/,1,2I8,/,	(CENA1 to CENA83)
+9I8,/,4I8,/,	(GELA1 to GELA13)
+8I8,/,	(REFA1 to REFA8)
+9I8,/,5I8,/,	(GELB1 to GELB14)
+9I8,/,6I8,/,	(CELB1 to CELB15)

+9I8,/,5I8,/,	(GELC1 to GELC14)
+6(9I8,/),3I8,/,	(CENB1 to CENB57)
+9I8,/,7I8,/,	(GELD1 to GELD16)
+2(9I8,/),I8,/,	(CELC1 to CELC19)
+6I8,/,	(REFB1 to REFB6)
+9I8,/,7I8,/,	(EUR1 to EUR16)
+9I8,/,9I8,/,	(GELE1 to GELE18)
+9I8,/,9I8,/,2I8,/,	(CELD1 to CELD20)
+2(9I8,/),I8,/,	(GELF1 to GELF19)
+9I8,/,9I8,/,I8,/,	(GELG1 to GELG19)
+9I8,/,6I8)	(EURB1 to EURB15)

POSIT=1 indicates, that the variables from B84A00B shall be appended after the variables on P84A11A. OPTION=2 means, that P84A11A contains more units than B84A00B, and that the units in question will be written on P84A12A having zero-filled variables from B84A00B. Consequently P84A12A will contain the same number of units as P84A11A. Remember, that the unit identification variable IDENTB on file TAPE2=B84A00B will not be written on TAPE3=P84A12A, so that the format statement must be reduced from (8I8,/,8I8) to +9I8,/,6I8).

12. STANDARD FILES IN THE DATA ARCHIVE

Alle files in the archive are primarily stored in the tape ARCHIVE system. When the user wants to retrieve a file, the following control statement must be issued:

ARCGET,filename/UN=username,PW=password,OP=d or s.

Username and password is available at the donor. The filenames correspond to the syntax described in section 7.2.

Filename	comments	format
U99E01A	MODIFY program library contains the decks: KOMVALG, FORKODE,FRIFELT,INDSAET.	irrelevant
Z70A00A	Unit identifications. See section 5 and 9.2.	(I6,1X,3R10,1X,I6,1X,I6,1X,I3,1X,I4)
C81A10B	Standard input data-file to program KOMVALG.	irrelevant
C81A01B	Standard input data-file to program KOMVALG	irrelevant
C81A04B	Standard input data-file to program KOMVALG.	irrelevant
C81A30B	Standard version of local elections 1970, 1974, 1978 and 1981 comprising 22 list-groups. See section 8.5 and 8.7.	(I6,1X,3R10,2I3,2(/,I6,I7,I3,2I4,4I7,2I3,11(/,2(I2,2I3,I7,I6,I3,2I2,1X))),/,2(/,I6,I3,2I4,3I7,2I3,11(/,2(I2,2I3,I7,I6,I3,2I2,1X))))

P84A12A    Standard data-archive exclu-    see section 11.5.  
                 sive of local elections.

---

P84A22A    Standard version of da-    see section 2.6.  
                 ta archive inclusive of  
                 local elections.

---

P84A04C    SPSS-ARCHIVE version    irrelevant  
                 of P84A22A

---

This publication describes a data-archive containing aggregate electoral and census statistics since 1970. All Danish elections on national, county and local levels plus national referanda are included. Census statistics cover demography and social data. Both kinds of information are compiled from existing statistical publications. The statistical data are aggregated in units corresponding to local councils (kommuner). Furthermore, the larger metropolitan areas are subdivided into smaller electoral wards.

All data are available on electronic media. They can be analysed per se or be connected with other types of data such as surveys.

Finally, a number of computer programs are described which facilitate modification and extension of available information in the data-archive.