

# Interne notater

STATISTISK SENTRALBYRÅ

89/12

10. februar 1988

**BRUKERVEILEDNING**

for

**PRINTERTAB**

på IBM

av

**Per H. Strand, Kontor for befolkning, Kongsvinger**

## FORORD

Etter å ha brukt Printertab til framstilling av bl. a. seks store publikasjoner og en mengde andre tabeller gjennom fire år, vil jeg gjerne dele mine erfaringer med andre. Dette heftet er ikke ment som en komplett beskrivelse av Printertab, men som en igangsetter for dem som ønsker å produsere tabeller på en enkel måte.

Heftet er blitt til ved å samle opp egne notater og huskelapper etter som Printertab har kommet i nye versjoner. Det er mitt høyeste ønske at dette heftet kan hjelpe noen til å produsere bedre tabeller på kortere tid.

Det er med stor respekt for skaperen av printertab, Halvor Strømme, jeg skriver dette. Uten hans hjelp ville dette heftet ikke sett dagens lys.

Alle kommentarer vedrørende program og hefte er velkomne!  
Halvor Strømme, Kontor for befolkning, Kvgr. Rom C206. Tlf 5264.  
Per H. Strand, Kontor for befolkning, Kvgr. Rom C210. Tlf 5267.

Per H. Strand  
Kontor for befolkning, Kongsvinger

INNHALDSFORTEGNELSE

	side
1. GENERELT . . . . .	3
1.1 INNFØRING . . . . .	4
1.2 VIRKEMÅTE . . . . .	5
1.3 BRUK . . . . .	6
1.4 FILENE . . . . .	6
2. DE ENKELTE FILENE . . . . .	7
2.1 TEKSTFILEN . . . . .	7
2.1.1 OVERSKRIFTEN . . . . .	8
2.1.2 HODET . . . . .	9
2.1.3 FORSPALTEN . . . . .	10
2.1.4 KOLONNEREDIGERING . . . . .	12
2.1.5 FOTNOTEN . . . . .	14
2.2 REDIGERINGSFILEN . . . . .	15
2.3 DATABESKRIVELSEN . . . . .	17
2.4 DATAFILEN . . . . .	18
3. DATASETTENE . . . . .	19
3.1 TEKSTER . . . . .	20
3.2 REFERAN . . . . .	20
3.3 DATABESK . . . . .	20
3.4 TABELLER . . . . .	20
3.5 DATASETT ELLERS . . . . .	21
3.5.1 DATAFIL . . . . .	21
3.5.2 MELDING . . . . .	22
3.5.3 TABINN . . . . .	22
3.5.4 KORREK . . . . .	22
4. KJØRING . . . . .	24
4.1 MENYBILDE . . . . .	24
4.1.1 VALG '0' . . . . .	25
4.1.2 VALG '1' . . . . .	27
4.1.3 VALG '2' . . . . .	29
4.1.4 VALG '3' OG '4' . . . . .	29
4.1.5 VALG '5' '6' OG '7' . . . . .	30
4.1.6 VALG '8' . . . . .	33
4.1.7 VALG 'E' . . . . .	34
4.1.8 VALG 'F' . . . . .	35
4.2. 'TIPS AND HINTS' . . . . .	36
4.2.1 BLANKE FOR 'TUSEN' . . . . .	36
4.2.2 'PRIKKING' . . . . .	36
4.2.3 'LESETIDFORKORTEELSE' . . . . .	37
4.2.4 OPPRETNING PÅ TABINN . . . . .	38
5. PLANLEGGING . . . . .	39
6. HUSKELISTE . . . . .	40

## 1. GENERELT

Printertab lager trykkeferdige tabeller ut fra egne filer. En fil med tekst, en fil med redigeringsparametre, en med databeskrivelsen, og selve datafilen. Printertab har muligheter for å utføre de fleste beregninger både på kolonne- og linjenivå. Det finnes også flere rutiner for å ivareta de enkelte tabellers utseende.

Selve kjøringen (eksekveringen) utføres i to steg. Det første steget setter de forskjellige delene som tabellen består av sammen til en eller flere tabeller. Dette steget kalles Preprocessor. Det andre steget henter tallene inn, bearbeider disse og setter resultatet inn i tabellen der de hører hjemme. Dette er selve tabellprogrammet.

Ressursforbruket i tabellprogrammet er avhengig av datafilens struktur. Ligger dataene på det minste nivå og aggregering skjer til et vesentlig høyere nivå vil dette legge beslag på unødige ressurser. Erfaringen viser at en forhåndsbearbeiding ved hjelp av f.eks. Easytrieve Plus vil senke kjøretiden ned til under det halve.

Det vil bli laget datasett med eksempler på de forskjellige tabelltyper og konstruksjon. Hvordan dette skal organiseres er ikke fastlagt så langt, men dette vil bli offentliggjort så snart disse foreligger. Der vil det ligge inne medlemmer til flere forskjellige tabelltyper. Det forutsettes at disse er så selvforklarende som mulig med fylldige kommentarer.

Denne veiledningen vil ta for seg de forskjellige filene, deres bruk, organisering og oppbygging. Det er lagt vekt på praktiske eksempler som middel til raskere forståelse. Det er i første rekke en innføring i tabellprogrammet som det tas sikte på her. Det er å håpe at denne enkle innføring øker forståelsen av, og ønsket om å ta i bruk, Printertab.

Til slutt vil det følge en liten innføring i enkelte rutiner som gjør at programmet kan skreddersys den enkeltes behov.

## 1.1 INNFORING

Printertab er utviklet i COBOL og ivaretar Byråets spesifikke måte for oppbygging av tabeller. For lettere å kunne kombinere deler av tabeller, er disse splittet slik at de ligger lagret på forskjellige datasett/membere. Hver av disse kan derfor kopieres, editeres eller fjernes uten at dette innvirker på de øvrige tabelldelene.

For å kjøre printertab må derfor de forskjellige tabelldelene tas inn i programmet. Dette skjer ved å spesifisere kun medlemnavn til de forskjellige delene, i og med at de forskjellige delene ligger på predefinerte datasett. Disse er faste og printertab leter derfor ikke på andre datasett etter medlemene som oppgis.

Printertab legger beslag på forholdsvis store ressurser på IBM ved kjøringen. Det er derfor nødvendig med planlegging både av selve tabellens utseende og strukturen i datafilen (der tallene finnes). Når det gjelder tabellens utforming vises det til 'SSBs REGLER FOR PRODUKSJON AV TABELLER I PUBLIKASJONER' som kan bestilles fra tekstbehandlingskontoret eller fra administrasjonskontoret.

I og med at det kan lages forskjellige tabelldeler hver for seg, er fleksibiliteten stor. Tabeller som f.eks. har forskjellige forspalter men ellers er lik trenger derfor kun ny forspalte og ny overskrift. De øvrige delene brukes 'om igjen' til alle tabellene.

Dette innebærer at når man en gang har laget et medlem med f. eks. alle kommunene i forspalten, så kan man ta den i bruk til alle andre tabeller på kommunenivå. Dette medfører at man som regel har kortere produksjonstid ettersom biblioteket vokser. Dette vil også spare plass, idet man ikke har mer på lager enn strengt tatt nødvendig.

## 1.2 VIRKEMÅTE

De fleste kontorer har i dag PC med regnearkprogram, og for dem som har kjennskap til dette er Printertab lett tilgjengelig. Cellenes adressering er ikke ulikt det vi finner i regneark. Beregning foregår med adresser eller pekere til en eller en serie med celler.

De begrensninger som ligger inne her er at antall kolonner ikke kan overstige 50, kolonne 51 inneholder en form for 'tally' det vil si at en her finner en total for linjen. Høyden på et ark vil normalt ikke overstige en trykkeklar side. En deltabell kan ikke overstige 99 linjer med celler (tall). Innenfor disse marginer kan man foreta beregninger med celler som grunnlag, konstanter eller en kombinasjon av disse.

På samme måte som enkelte regneark kan man beregne flere ganger til en celle, det gjelder her å holde tunga rett i munnen. Så snart et resultat er skrevet ut på en celle, vil det som sto i cellen være borte.

Den avgjørende forskjellen mellom Printertab og de forskjellige regneark er måten man bygger opp en tabell på. I regneark legger man tekst, formler og data inn på samme fil. For å imøtekomme en større fleksibilitet har Printertab's opphavsmann valgt å legge de forskjellige deler av tabellen ut på forskjellige filer. Den mest iøynefallende fordelingen med dette er at har man f.eks. først laget en datafil til en publikasjon, eller del av en publikasjon, slipper man å legge tallene inn i hver tabell idet datafilen leses inn når Printertab kjøres.

Printertab utfører beregninger på realtall. Dette vil si at vi kan bruke datafiler med aggregerte tall og ikke antall forekomster (F.eks. en linje pr. person). Vi kan derfor lage egne datafiler som tar liten plass og bruker lite ressurser.

Printertab har altså plukket med seg det beste fra regnearkprogram og kombinert dette med mulighetene med å operere med flere filer som ligger under batch på IBM/Comparex.

### 1.3 BRUK

Formålet med Printertab er at det skal være så enkelt som mulig i bruk. Hele konseptet er gjennomført med dette for øyet at bruker skal ha full kontroll uten å måtte sette seg inn i IBMs mangesidige gjøren og laden. Det er derfor lagt opp nærmest som et regneark, hvor brukeren legger inn funksjoner til hver 'celle' som forteller hva som skal utføres av beregninger.

### 1.4 FILENE

- Tekstfilen inneholder hoder, forspalter, fotnoter og selektering. Den kan også inneholde beregninger der dette er ønskelig (se pkt.1).
- Redigeringsfilen inneholder hvilke hoder som skal stå over hvilke forspalter idet tekstfilen inneholder alle hoder, forspalter osv. Videre forteller redigeringsfilen hvor mange kolonner vi skal ha, og kan også brukes til beskrive hvor disse skal starte/ende.
- Databeskrivelsen inneholder navn på variablene i datafilen, samt første og siste posisjon på disse.
- Datafilen inneholder de tall som skal forekomme i tabellen(e). Er filen stor og inneholder små gruperinger, kan en spare tid ved å først utføre de aggregeringer, selektering og event. beregninger /avledete variabler. Dette må i alle tilfelle gjøres hvis data finnes på flere filer.

## 2. DE ENKELTE FILER

Vi skal her ta for oss en enkel tabell for å vise oppstillingen for de forskjellige delene av printertab. For de datasett som brukes vises til pkt. 3.

### 2.1 TEKSTFILEN

Nedenfor ser vi en hel tabell med de mest vanlige elementene. Denne tabellen er på 1 side. Tabeller over flere sider vil inneholde '02' med 'Tabell 1 (forts.)'. Folketall i ...osv.' for de neste sidene. Dette vil bli grundigere forklart i tekstene under det enkelte element. Det er her viktig å legge merke til at alle elementhenvisninger har store bokstaver, '02' og 'f5' blir altså ikke godtatt!

```

***** ***** TOP OF DATA *****
000001 01
000002 Tabell 1. Folketall i byregioner pr 1/1 1970, 1975, 1980 og 1984.
000003 H1
000004 -----
000005
000006                                Befolkning
000007 Byregion -----
000008                                pr. 1/1    pr. 1/1    pr. 1/1    pr. 1/1
000009                                1970      1975      1980      1984
000010
000012 TAB-----+-----+-----+-----+-----+-----
000013 -----
000014 Hele landet      .      3866468    3997521    4078900    4134353
000015 F1
000016 00                Alle byregioner .....
000017 01
000018 0101             Sarpsborg-
000019 0101 BY =01      Fredrikstad .....
000020 0102 BY =02      Oslo .....
000021 0103 BY =03      Drammen .....
000022 0104 BY =04      Tønsberg .....
000023 0105             Skien-
000024 0105 BY =05      Porsgrunn .....
000025 0106 BY =06      Kristiansand .....
000026 0107             Stavanger-
000027 0107 BY =07      Sandnes .....
000028 0108 BY =08      Bergen .....
000029 0109 BY =09      Trondheim .....
000030 K1
000031 01                SUMBE70
000032 02                SUMBE75
000033 03                SUMBE80
000034 04                SUMBE84
000035 05(1)=04*K1000/K4134353
000036 06(0)=04*K100/01
000037 N1
000038 -----
000039 Kilde: Statistisk sentralbyrå
***** ***** BOTTOM OF DATA *****

```



### 2.1.1 OVERSKRIFTEN

000001 01

000002 Tabell 1. Folketall i byregioner pr 1/1 1970, 1975, 1980 og 1984.

Overskriftene markeres med '0' etterfulgt av et siffer fra 1 - 999. Sifferet er ikke systemavhengig. Dette gjør det mulig å lage eget system. En tabell over 6 oppslagssider kan derfor nummereres som f. eks.

```

-----
! 011  ! 012  !      ! 021  ! 022  !      ! 031  ! 032  !
!      !      !      !      !      !      !      !      !      osv.
!      !      !      !      !      !      !      !      !
!      !      !      !      !      !      !      !      !
-----

```

Neste linje inneholder selve teksten i tabelloverskriften og kan inneholde flere linjer. Det er viktig at alle startmarkeringene '01', 'H1' osv. starter i første posisjon (helt til venstre).  
HUSK STORE BOKSTAVER!

For å få luft rundt tabellene ved utkjøring til printer er det her lurt å legge på blanke linjer. F.eks. 10 blanke linjer hvis du kun legger disse inn i overskriften, eller 5 blanke linjer hvis du legger disse inn både i overskrift og etter fotnoten.

000001 01

000002

000003

000004

000005

000006

000007 Tabell 1. Folketall i byregioner pr 1/1 1970, 1975, 1980 og 1984.

Hvis du vil legge inn kommentarer eller huskeliste for hvilke deler som hører med til tabellen, kan du legge disse inn som en overskrift som du ikke bruker når tabellen settes sammen.

000001 099

000002 referanser : test01

000003 databeskrivelse : testfil

000004 datafil : PXnnn.Snnnn.ini.TESTFIL

000005 01

000006 Tabell 1. Folketall i byregioner pr 1/1 1970, 1975, 1980 og 1984.

På denne måten er det greit å ta igjen tabellene når neste publikasjon skal ut, eller etterfølgeren i stillingen skal kjøre tabellene.

## 2.1.2 HODET

```

000003 H1
000004 -----
000005
000006
000007 Byregion
000008
000009
000010
000011
000012 -----
000013 hele landet
000014 TAB-----+-----+-----+-----+-----+-----

```

	Befolkning			
	pr. 1/1 1970	pr. 1/1 1975	pr. 1/1 1980	pr. 1/1 1984
hele landet	3866468	3997521	4078900	4134353

Hodet markeres med 'H' etterfulgt av et siffer med verdi fra 1 til 999 og skal komme på første ledige linje etter overskriften (husk på å venstreorientere alle markeringene). I hodet legges malen for tabellen, og overskrift, fotnote og redigering utføres etter denne mal. Hodet omsluttet av streker som går i hele tabellens lengde. Høyden på hodet er ikke kritisk og kan utformes som ønsket.

I dette eksemplet er det lagt inn en konstant til tabellen. For å lette arbeidet med datafilen til tabellen er denne lagt inn 'for hånd' som en del av hodet. Dette kan brukes hvis den filen vi har ikke inneholder tallene.

```

000014 TAB-----+-----+-----+-----+-----+-----

```

Denne linjen er ikke en del av teksten. Den forteller hvor de enkelte kolonner slutter. Hvis vi bruker 'COLS' i edit vil vi se:

```

=COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+
000014 TAB-----+-----+-----+-----+-----+-----

```

Forspalten slutter her i posisjon 20, første kolonne slutter i posisjon 29, andre i pos. 40 osv. Vi kan med andre ord plassere forspalten og kolonnene der hvor de hører hjemme uten å telle posisjoner! Hvis en eller flere kolonner er feilplassert er det bare å flytte merket '+' og feilen er rettet.

### 2.1.3 FORSPALTEN

Forspalten består av flere elementer som hver for seg utformer den ferdige tabellen. Først ser vi på hele forspalten for siden å plukke den fra hverandre for å se nærmere på de enkelte elementer og deres funksjon:

```

000015 F1
000016 00           Alle byregioner .....
000017 01
000018 0101        Sarpsborg-
000019 0101 BY   =01   Fredrikstad .....
000020 0102 BY   =02   Oslo .....
000021 0103 BY   =03   Drammen .....
000022 0104 BY   =04   Tønsberg .....
000023 0105        Skien-
000024 0105 BY   =05   Porsgrunn .....
000025 0106 BY   =06   Kristiansand .....
000026 0107        Stavanger-
000027 0107 BY   =07   Sandnes .....
000028 0108 BY   =08   Bergen .....
000029 0109 BY   =09   Trondheim .....

```

000015 F1

Forspalten markeres med 'F' etterfulgt av et siffer med verdi fra 1 til 99 og skal stå på første ledige linje etter hodet (husk venstreorientering!). Som de øvrige markeringene er ikke sifferet systemavhengig, og kan derfor velges fritt men i stigende rekkefølge.

```

000016 00           Alle byregioner .....

```

Dette er en totallinje. Det vil si at printertab ikke legger sammen alle linjene som forekommer i tabellen, men summerer alle records i datafilen! Hvis den filen som blir brukt inneholder bare det materialet som tabellen skal inneholde, kan denne linje med fordel også brukes som summeringskontroll. Denne linje eller andre kan fjernes ved 'd' (delete) i edit på den ferdige tabellen. Hvis andre summeringer ønskes kan dette gjøres i kolonneredigeringen (pkt.3.1.4). Denne '00'-linje er ikke påkrevet og kan sløyfes.

000017 01

Denne linje markerer at her begynner deltabell nr. 01. Alle linjer i denne deltabellen skal derfor starte med '01nn'. Deltabellen kan som de øvrige elementer ha verdien fra 01 til 99. Forskjellen her er at verdien må starte på 01, neste skal være 02, så 03 osv.

000018 0101 Sarpsborg-  
000019 0101 BY =01 Fredrikstad .....

Dette er deltabell nr. 01 og første linje '01'. Også her er linjenr. jevnt stigende. Ut for hver ny linje kommer en selektering. Her er valgt byregioner som selektering for de enkelte linjer i tabellen. Hver byregion er i denne datafilen gitt en kode med 2 siffer. Samme linjenr (0101) forekommer her to ganger. Dette er gjort fordi teksten i forspalten tar så mye plass at tallene i tabellen vil bli for sammentrengt. Antall linjer med tekst for hvert linjenr. er ikke systemavhengig og kan derfor velges fritt. Variabelnavnet i selekteringen er på maksimum 5 posisjoner. Det samme variabelnavnet må defineres i filen for databeskrivelse! Tallene fra datafilen skrives alltid inn på siste linjenr., hvor også eventuell selektering finner sted. Inneholder datafilen flere records (linjer) med den samme selekteringsverdien, vil printertab aggregere (summere) automatisk.

000020 0102 BY =02 Oslo .....

Deltabell 01, linje 02 med ny selekteringskode og ny tekst. Punktene ut for siste linje er gjort for lang. Lengden på denne bestemmes i redigeringsfilen. Pass derfor på at teksten ikke blir 'hugget' av i redigeringen. Lange tekster kan med fordel deles over flere linjer.

000021 0103 BY =03 Drammen .....  
000022 0104 BY =04 Tønsberg .....  
000023 0105 Skien-  
000024 0105 BY =05 Porsgrunn .....

Husk at hvis tekst deles over flere linjer skal samme linjenummeret skrives hver gang, og at selekteringen skrives ut på siste linje.

000025 0106 BY =06 Kristiansand .....  
000026 0107 Stavanger-  
000027 0107 BY =07 Sandnes .....  
000028 0108 BY =08 Bergen .....  
000029 0109 BY =09 Trondheim .....

Eksemplet her viser et seleksjon på en bestemt verdi. Printertab har selvfølgelig også muligheter for 'større enn', 'mindre enn' og 'ulik' som selekteringskriterie. f.eks.:

000029 0109 BY <09 'BY' med lavere kode enn 9.  
000029 0109 BY >09 'BY' med høyere kode enn 9.  
000029 0109 BY U09 'BY' med kode ulik 9.

### 2.1.4 KOLONNEREDIGERING

Under kolonneredigeringen bestemmes hvilke selekteringskriterier som skal gjelde for hver av kolonnene. Videre kan det foretas beregninger på materialet. Legg her nøye merke til posisjonene for de forskjellige elementene i kolonneredigeringen. Feil, manglende tall/deler av tabellen kan skyldes feil posisjoner.

000030 K1

'K1' betyr kolonneredigering nr. 1, og starter som før til venstre. Her kan fritt velges siffer fra 1 til 999.

000031 01 SUMBE70

'01' er her kolonne nr. 1. I dette tilfellet skal befolkningen i 1970 summeres. Variabelnavnet (BE70) er på forhånd definert i filen for databeskrivelse og etterfølger kommandoordet 'SUM'. Sum betyr at her skal det skrives ut summen av alle records (linjer) i datafilen som har det angitte variabelnavn. Se også tilslutt i pkt. 2.1.4.

000032 02 SUMBE75

Samme som over, men dette er kolonne 2 og variabelen er befolkningen i 1975 med variabelnavnet 'BE75'.

000033 03 SUMBE80

000034 04 SUMBE84

Det kan her legges til at uten 'SUM' vil kun opptelling skje! Det innebærer at innholdet i variabelen 'BExx' ikke legges sammen, men tallet som framkommer er antall tilslag (records) på selekteringen!

000035 05(1)=04\*K1000/K4134353

I kolonne 5 skal det her foretas en beregning. Parentesen i linjer med beregning inneholder antall desimaler og skal alltid forekomme. Videre skal '=' etterfølges av selve formelen. I dette tilfellet er det befolkningen i 1984 (hentes fra kolonne 04) i promille av hele landets befolkning (4 134 353). Beregninger på variable hentes altså fra en kolonnebetegnelse mens faste eller konstante tall i formelen alltid starter med en 'K' (konstant).

000036 06(0)=04\*K100/01

Her er det en beregning av befolkningen i 1984 i forhold til 1970 uttrykt som indeks. '(0)' forteller at det ikke være noen desimaler i kolonne 6. Printertab foretar eventuell forhøyning automatisk.

I dette tilfellet er det kun 1 kriterium for valg av variabler til kolonnene. Hvis vi f.eks. hadde en kjønnsfordeling av befolkningen ville vi kunne fordele dette slik:

```
000031    01KJ    =1    SUMBE70
000032    02KJ    =2    SUMBE70
000033    03      =      SUMBE70
```

Dette ville ha gitt alle menn i 1970 i første kolonne, alle kvinner 1970 i andre kolonne og total befolning 1970 i tredje kolonne.

Tabellen kan inneholde maksimum 50 kolonner (det finnes 51. se under). Selv om tabellen sjelden inneholder mer enn ca. 10 kolonner, kan de øvrige fritt nyttes som hjelpefelt for forskjellige beregninger.

En nyttig detalj er at kolonne 51 inneholder totalen for hele linjen og det er derfor enkelt å prosentuerer ved hjelp av denne. Eks.:

```
000031    11          SUMMENN          Alle menn
000032    12          SUMKVINN        Alle kvinner
000033    01(1)=11*K100/51          Menn*k100/(menn+kvinner)
000034    02(1)=12*K100/51          Kvinner*k100/(menn+kvinner)
000035    03(1)=K100                Konstanten 100,0
```

Her er tallgrunnlaget lagt 'utenfor' tabellen (kol. 11 og 12) for siden å bli trukket med i beregningen sammen med summen av linjen (kol. 51). Resultatet blir lagt i kolonne 1 og inneholder en desimal. I kolonne 3 er det lagt inn en konstant. Dette gjøres ofte når det regnes i prosent/promille, idet summetallet ikke alltid blir lik 100/1000. Dette har sammenheng med reglene for forhøyning.

Hvis du skal legge sammen flere etterfølgende kolonner, bruk:

```
000036    13(0)=13+TIL22          Alle kolonner fra og med 13
                                          til og med 20 summeres
```

Det er også muligheter for beregninger på linjer på samme måte som på kolonner. Du kan også kombinere dette.

```
000038 0111 (0)=0111+0112+0113+0114  Alle linjer fra og med 0111
                                          til og med 0114 aggregeres
```

```
000039 011105(1)=04*K100/51          Alle linjer fra og med 0111
000040 TIL                            til og med 0118, beregning
000041 011805(1)=04*K100/51          av prosent til kolonne 05
```

### 2.1.5 FOTNOTEN

000037 N1

Fotnotene starter som de øvrige deler av tekstfilen med bokstav og siffer. Bokstaven 'N' etterfølges av siffer med verdien fra 1 til 999 hvor verdien ikke er systemavhengig.

000038 -----

000039 Kilde: Statistisk sentralbyrå

Fotnoten til denne tabellen er en avsluttende linje og egenreklame. For tabeller som går over flere sider brukes kort linje. Hvis det skal kjøres ut mange tabeller samtidig, vil det være fornuftig å putte inn 5 - 10 blanke linjer for lettere å kunne skille disse fra hverandre.

000037 N1

000038 -----

000039 Kilde: Statistisk sentralbyrå

000040

000041

000042

000043

000044

000045 N10

000046 -----

000047 Kilde: Statistisk sentralbyrå

000048

000049

000050

000051

000052

Det vises ellers til regler for bruk av linjer/fotnoter i tabellregulverket.

## 2.2 REDIGERINGSFILEN

Nedenfor ser vi en komplett redigeringsfil til tabelleksemplet som er brukt her. I denne filen bestemmer vi hvilke hoder, forspalter etc. som skal kombineres; videre linjeavstander i de forskjellige elementene i den endelige tabellen og hvor mange kolonner den inneholder, samt plasseringen av disse i forhold til forspalten. Vær nøye med posisjoner!

Under punkt 2.1.2 viser vi hvordan redigeringen forenkles vesentlig ved hjelp av 'TAB' kommandoen. Dette oppsettet vil derfor bare være av interesse for dem som 'arver' gamle program som skal settes i produksjon med ny årgang data.

```
***** ***** TOP OF DATA *****
000001 01H1F1K1N1
000002 RED   1320419 09111111
***** ***** BOTTOM OF DATA *****
```

```
000001 01H1F1K1N1
```

Her forteller vi at overskrift 1, hode 1, forspalte 1, kolonne 1 og fotnote 1 skal kombineres til en tabell. Linjen starter helt til venstre og alle parametre kommer fortløpende (ingen åpen plass for 2. siffer)

```
000002 RED   1320419 09111111
```

Selve redigeringen starter alltid med 'RED'. Deretter kommer en serie på 3 siffer hvor linjeavstanden bestemmes. Første siffer er avstanden mellom hver linje i tabellen. Andre siffer er avstanden mellom deltabellene og tredje siffer er avstanden fra hode til forspalten (første linje i tabellen). De to neste sifferne '04' viser hvor mange kolonner det er i tabellen. De to siste sifferne i denne kombinasjonen forteller hvor mange posisjoner forspalten skal ha (pass på at du får med all tekst).

Den neste serien med siffer er antall posisjoner mellom hver kolonne (siste posisjon i kolonnen er den som oppgis). Her slutter altså første kolonne 9 plasser etter forspalten (to siffer; derfor '09'), andre kolonne 11 plasser etter første kolonne osv.



000002 RED Skal alltid være med  
 000002 RED 132 Linjeavstand (linje,hode,deltab.)  
 000002 RED 13204 Antall kolonner  
 000002 RED 1320419 Lengden på forspalten

OBS! Kolonneredigering er ikke  
 OBS! nødvendig ved bruk av funksjonen  
 OBS! 'TAB' i hodet på tekstfilen!

000002 RED 1320419 09 Kolonneredigering:  
 000002 RED 1320419 09 Siste siffer 9 plasser til høyre  
 000002 RED 1320419 0911 Siste siffer 11 plasser til høyre  
 000002 RED 1320419 091111 Siste siffer 11 plasser til høyre  
 000002 RED 1320419 09111111 Siste siffer 11 plasser til høyre  
 000002 RED 1320419 09111111 for siste angitte posisjon!

OBS! Husk; alltid 2 posisjoner!

Skal flere deltabeller med i kjøringen, legges en ny redigerings-  
 enhet umiddelbart etter den første. eks.:

000001 01H1F1K1N1  
 000002 RED 1320419 09111111  
 000003 02H1F2K1N1  
 000004 RED 1320419 09111111  
 000005 02H1F3K1N10  
 000006 RED 1320419 09111111

Linje 000003 og 000004 er deltabell 2. Denne har ny overskrift  
 (Tabell 1 (fors.). Folketall i ....), samme hode, ny forspalte,  
 samme kolonne og fotnote.

Linje 000005 og 000006 er deltabell 3. Denne har samme overskrift  
 som deltabell 2, samme hode, ny forspalte, samme kolonne men fot-  
 noten er ny.

På denne måten kan man sette sammen tabellene slik som ønsket. Det  
 gjelder bare å holde tungen rett i munnen. Ved å se nøye igjennom  
 tekstfilen og redigeringsfilen (bruk gjerne papir og blyant), kan  
 man spare seg for mye unødig tidsbruk. Printertab bruker som sagt  
 noe ressurser under eksekvering.

## 2.3 DATABESKRIVELSEN

Databeskrivelsen ligger som egen fil i printertab; dette fordi samme filen ofte brukes til hele serier med tabeller eller et helt NOS-hefte. Den består helt enkelt av variabelnavn, start og stopposisjon samt en forklaring/beskrivelse av dataene. Alle felter som skal inn i tabellen kan bare inneholde siffer!

```
***** ***** TOP OF DATA *****
000001 BY      0001-0002  Kode for byregion
000002 BE70    0003-0009  Befolkning 1970
000003 BE75    0010-0016  Befolkning 1975
000004 BE80    0017-0023  Befolkning 1980
000005 BE84    0024-0030  Befolkning 1984
***** ***** BOTTOM OF DATA *****

000001 BY      0001-0002  Kode for byregion
```

Her er første variabelen som skal brukes 'BY' (datafilen kan inneholde felter som ikke skal brukes til tabellproduksjon, og disse skal heller ikke defineres her). Den starter på posisjon 1 '0001' og ender på posisjon 2 '0002'. Deretter følger en beskrivelse av feltet. Dette er gjort fordi variabelnavnet ikke kan inneholde mer enn 5 posisjoner (plassproblemer) og ikke alltid gir et klart bilde av hvilke data en jobber med.

Er det filer som fra før er dokumentert på IBM kan man kopiere filebeskrivelsen over til seg selv fra 'SSB1.DOKUMENT.RECORD' for deretter å legge inn egne variabelnavn (det øvrige er alt på plass). Beskrivelsen er ikke nødvendig for Printertab, men for deg selv eller den som skal overta etter deg.

```
000001 BY      Variabelnavnet (maks 5 posisjoner)
000001 BY      0001      Startposisjon i datafilen
000001 BY      0001-    Mellomtegn
000001 BY      0001-0002  Siste posisjon i datafilen
000001 BY      0001-0002  Kode for..  Beskrivelse av variabelen
```

Det er ikke nødvendig at alle felter i filen skal defineres/beskrives. Dette innebærer at man slipper å lage egen fil for Printertab, man definerer kun de felt som er av interesse i sammenhengen (det vil si: KUN TALL).

## 2.4 DATAFILEN

Datafilen er enten en rådatafil (en hvilken som helst fil på IBM), eller en tilrettelagt fil for printertab. Tilrettelegging (agg-regering, selektering og event. nye variable (avledete)) gjøres lett med Easytrieve Plus på IBM. Hvis filen er stor er dette en fordel. Printertabkjøringen krever tildels store ressurser ved søking i store databaser, så her kan man virkelig korte ned på produksjonstiden. Det kan være enklere å lage 3 - 4 små tabelldatafiler enn å legge mye arbeid i en stor og uoversiktig tabelldatafil. Filen til vårt eksempel kan se slik ut:

```
***** ***** TOP OF DATA *****
000001 010105293010973901093740109990
000002 020789866080720908127260819466
000003 030101507010806801095380112496
000004 040062092006396700654260066677
000005 050087804008980200927730092999
000006 060073682007904800820130084254
000007 070125886013546501444940152457
000008 080242772025317302538280256754
000009 090148479015905301620060163393
***** ***** BOTTOM OF DATA *****
```

Som vi ser stemmer dette med den beskrivelsen av datafilen (pkt.2.3)

000001 01	Kode for byregion (0001-0002)
000001 010105293	Befolkning 1970 (0003-0009)
000001 0101052930109739	Befolkning 1975 (0010-0016)
000001 01010529301097390109374	Befolkning 1980 (0017-0023)
000001 010105293010973901093740109990	Befolkning 1984 (0024-0030)

Som tidligere bemerket, er det ikke nødvendig å lage egen fil hvor alle felt er i bruk. Finnes fra før en høvelig datafil så bruk denne. Databeskrivelsen avgjør hvilke og hvor mange felt som skal med i tabellen(e).



### 3.1 TEKSTER

Her lagres tekstfilen til tabellene. Navnet på disse bør i størst mulig utstrekning følge tabell-/prosjektnavnet. For å lette oversikten bør hver tabell lagres som et eget member. Datasettet har en fast postlengde (linjelengde) på 150 bytes (posisjoner). Dette fordi tabellen som regel nedfotograferes før trykking.

### 3.2 REFERAN

Referan inneholder referansene til tabellen. Denne inneholder alle filer med redigeringsparametrene til de forskjellige tabeller. Referansesfilen bør ha samme navnet som tabellen/tabellserien. Postlengden her er på 80 bytes.

### 3.3 DATABESK

Her finner vi beskrivelsen av de forskjellige datafilene som brukes til produksjonen av tabellene. Det bør også her brukes navnsetting som korresponderer med de øvrige filer til tabellen(e). Postlengden er her på 80 bytes.

### 3.4 TABELLER

I dette datasettet legges/ligger de ferdige tabellene. Du bestemmer selv navnet på hver enkelt tabell. Det kan være lurt å sette disse i system allerede fra starten. Eks.: forbrx01 forbrx02 osv. er bedre enn tabell1 tabell2 osv. Membernavnet er altså memo-navnet på hver enkelt tabell. Postlengden er her 132 bytes.

Etter at tabellen(e) er kjørt kan man ta en tabell inn i edit for 'finpuss' som justering av kolonner, overskrift etc., sette inn/ta ut linjer eller flytte linjer fra en side til neste. En ting du må gjøre i edit er der hvor tabellen går over to sider og forspalten skal stå som 'etterspalte' (lengst til høyre). Se ellers punkt 3.5.4 hvor dette også tas opp.

### 3.5 DATASETT ELLERS

Under selve kjøringen brukes to datasett i tillegg til dem vi bruker til utformingen av tabellene. Dette er datasett vi vanligvis ikke er i kontakt med, men de er aktive og vi kan dra fordel av en av disse.

```
PPkkk.S0000.ini.MELDING
PPkkk.S0000.ini.TABINN(member)
```

Og for dem som har kjørt/kjører korreksjonsprogrammet, finnes også:

```
PPkkk:S0000.ini.KORREK(member)
```

Til slutt har vi selve kjernen i alle tabeller, dataene, som ligger på en egen fil.

#### 3.5.1 DATAFIL

I datafilen ligger selve datadelen, det vil si de tall vi bygger tabellen på. Det kan være en råfil, enten fra vårt eget kontor eller importert, eller en tabellfil spesielt tilrettelagt for tabellproduksjon. Det mest vanlige formatet på denne er:

```
PXkkk.Snnnn.InnnAnAn.Gnnnn.Vnn
```

De fem første posisjoner er opplysninger om filen. De fem neste er statistikknr. Deretter følger identitetsdelen, generasjonsangivelse, og tilslutt versjonnr. Datafilens maksimale lengde er 2000 bytes (2kb), og den må være sekvensiell (ikke member). Videre må feltene være numeriske. Det er altså ikke tillatt med blanke i de variabler som skal settes inn i tabellen. Et kjennetegn for selektering kan derimot være alfanumerisk (skal kun brukes til dette). eks.:

```
000010 F1
000012 00          Landsdelene
000013 00          i alt .....
000014 0101  IND  =1          Under 200 .....
000015 0102  IND  =2          200-499 .....
000016 0103  IND  =3          500-999 .....
000017 0104  IND  =4          1000-1999 .....
000018 0105  IND  =5          2000-4999 .....
000019 0106  IND  =6          5000-9999 .....
000020 0107  IND  =7          10000-19999 .....
000021 0108  IND  =8          20000-49999 .....
000022 0109  IND  =9          50000 og over .....
000023 0110
000024 0110  IND  =0          Periferi .....
000025 0111  IND  =X          Uplassert .....
000025 0111  IND  =X          Uplassert .....
```

Indeksen IND har her verdien X, som er alfanumerisk og derfor ikke kan beregnes på!

### 3.5.2 MELDING

Dette datasettet inneholder ingen medlemmer. Her legges det ut eventuelle feilmeldinger etter at preprosessoren er ferdig. Man går inn og leser meldingene direkte fra menyen i printertab. Dette er et 'skyggedatasett' som ikke brukes på vanlig måte. Men settet må opprettes før kjøring! Postlengden her er 132 bytes.

### 3.5.3 TABINN

I TABINN er det to medlemmer 'KLADD' og 'TABINN'. Under 'KLADD' har printertab opprettet et kladdefelt for å ta vare på sine egne parametre for den ferdige tabellen og brukes derfor av preprosessoren. OBS! Gå ikke inn i 'KLADD' for å se hva som ligger der, det er likevel ikke mulig å forstå! under 'TABINN' lagres den komponerte tabellen klar for innsetting av tall i selve tabellprogrammet. Se under punkt 4.2 om hvordan denne kan benyttes. Datasettet har 160 bytes postlengde.

### 3.5.4 KORREK

Korreksjonsprogrammet er en enkel rutine for å omredigere tabeller. Det brukes enkle feltdefinisjoner (parametre) og kommandoer. Dette krever god kjennskap til hvor de enkelte elementer befinner seg i tabellen(e). Det vil i de langt fleste tilfeller være raskere og enklere og foreta redigering av tabeller i editoren 'EDIT' (pkt. 2 i hovedmenyen). Har du først gjort en feil er den uopprettelig! I 'EDIT' kan man trekke seg ut ved å kansellere jobben (COMMAND ==> can).

I korreksjonsfilen legger man inn parametre for korrigeringer som er nødvendige for å få den endelige tabell trykkeklar. Denne filen har ikke så stor aktualitet i dag i og med den sterke editoren som IBM har. Her finnes parametre for listing av filene, innsetting/fjerning av linjer og retting av linjer. En tabell over 2 sider (oppslag), har som regel forspalten eller deler av denne bakerst i tabellen. Siden printertab ikke kan legge alfanumeriske verdier andre steder enn i forspalten, vil en redigering være nødvendig.

```
***** ***** TOP OF DATA *****
000001 RETT 00001-001001;13;70/13;80;1
***** ***** BOTTOM OF DATA *****
```

Her flyttes alle de første 13 posisjoner til posisjon 70 og utover, deretter flyttes fra posisjon 13 og 80 posisjoner inn til posisjon 1. Dette gjøres fra linje 1 til og med linje 100.

000001 RETT 00001-001001;13;70/13;80;1

Vi skal se nærmere på det enkelte element og forklare disse:

RETT	forteller at noe skal rettes
RETT 00001	linjenummer(kan stå alene)
RETT 00001-	til(hvis fler enn en linje)
RETT 00001-00100	linjenummer
RETT 00001-001001	startposisjonen på feltet(start- posisjonen kan være 3 lang).
RETT 00001-001001;	semikolon forteller at et nytt parameter starter etter.
RETT 00001-001001;13	feltet er 13 posisjoner langt
RETT 00001-001001;13;	skilletegnet igjen
RETT 00001-001001;13;70	startposisjonen hvor feltet skal flyttes til
RETT 00001-001001;13;70/	skilletegn som forteller at flere instruksjoner følger
RETT 00001-001001;13;70/13	et felt som starter på pos. 13
RETT 00001-001001;13;70/13;	skilletegn
RETT 00001-001001;13;70/13;80	og er 80 lang
RETT 00001-001001;13;70/13;80;	skilletegn igjen
RETT 00001-001001;13;70/13;80;1	skal flyttes slik at det starter på posisjon 1 (og er 80 lang fremdeles)

Det er her viktig å huske at man ikke legger noe over et annet felt i tabellen som inneholder tall - DETTE BLIR DA OVERSKREVET! Hvis eksemplet over blir utført i den motsatte rekkefølge vil forspalten bli overskrevet (ødelagt) og de første 13 posisjonene (med befolkningstall) bli flyttet bakerst. Det er 200 posisjoner til rådighet slik at ved store ommøbleringer er det mulig å opprette hjelpe-felt som ligger utenfor den opprinnelige tabellen.

Det er mulig å legge inn tekst på eksisterende linje(r), samt på nye linjer som opprettes. Det er imidlertid lettere å ta den ferdige tabellen inn i 'edit' og redigere der.



#### 4. KJØRING

Produksjonen av en ferdig tabell skjer i to trinn. Den første, preprosessen, plukker ut de tabelldelene som skal settes sammen og sjekker at alle parametre stemmer med de faktiske forhold. Den legger ut et 'mellomresultat' som siden tas inn i selve tabellprogrammet. Den ferdige tabellen legges deretter ut på datasettet 'PPkkk.S0000.ini.TABELLER' med medlemnavn lik tabellnavn.

##### 4.1 MENYBILDE

, , , , , , , , , , , , , , , ,

P R I N T E R T A B - menybilde

VALG ==>

\_\_\_\_\_ Navn på mottaker av lister

STATISTIKKNR

VELG EN AV FØLGENDE RUTINER:

-----member-----	
1	TEKSTER _____ Lage/endre tekster og tabellparametre
2	REFERAN _____ Ekstrahere tabellkomponenter
3	DATABESK _____ Lage/endre databeskrivelse
4	KORREK _____ Lage/endre rettinger
-----	
5	INTERPAR Kjøre preprosessor
6	INTERTAB Kjøre tabellprogram
7	INTERKOR Kjøre korreksjonsprogram
8	SDSF Se job rapporter
E	INTERPAR Kjøre preprosessor(ny rutine)
F	INTERTAB Kjøre tabellprogram(ny rutine)

PF1/PF13=HJELP

PF3/PF15=AVSLUTT(END)

, , , , , , , , , , , , , , , ,

## 4.1.1 VALG '0'

```

, , , , , , , , , , , , , , ,
          P R I N T E R T A B - menybilde
VALG ==> 0

```

Navnet på den som skal ha listene og statistikknr. fylles ut. Er det første gangen du kjører printertab overhodet, velges '0' fra denne meny (står ikke på menybildet) og alle datasett som er nødvendig blir allokert.

```

-----filnavn----- --recordlengde-
1. Tekster : PPkkk.S0000.ini.TEKSTER(member)      150
2. Referan : PPkkk.S0000.ini.REFERAN(member)       80
3. Databesk: PPkkk.S0000.ini.DATABESK(member)      80
4. Melding : PPkkk.S0000.ini.MELDING                132
5. Tabinn  : PPkkk.S0000.ini.TABINN                 160
6. Tabeller: PPkkk.S0000.ini.TABELLER(member)      132
7. Korrek  : PPkkk.S0000.ini.KORREK(member)        200

```

Etter at 'enter' er trykket, flyttes kursor ned i venstre billedkant og blir stående der til du får melding om at jobben er kjørt.

Hvis du velger '8' (SDSF) fra menyen og punkt 'h' (DISPLAY JOBS...) vil følgende bilde komme fram etter 's' (SELECT) forran jobben:

```

***** TOP OF DATA *****
      J E S 2  J O B  L O G  -- S Y S T E M  I P O 1  -- N O D E  S S X

13.07.33 JOB 8709 TSS701I Kkkkini LAST-USED 01 JUL 88 10:03 SYSTEM=IPO
13.07.33 JOB 8709 TSS702I COUNT=07035 MODE=FAIL LOCKTIME=NONE NAME=XX
- - - o. s. v. - - -
13.07.33 JOB 8709 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SB
13.07.33 JOB 8709 -Kkkkinin ALOK 00 0 .00 .0
- - - o. s. v. - - -
----- JES2 JOB STATISTICS -----
01 JUL 88 JOB EXECUTION DATE
  37 CARDS READ
  95 SYSOUT PRINT RECORDS
   0 SYSOUT PUNCH RECORDS
   5 SYSOUT SPOOL KBYTES
- - - o. s. v. - - -
  7 //TEKSTER DD DSN=PPkkk.S0000.ini.TEKSTER,
      DISP=(NEW,CATLG),UNIT=SYSDA
- - - o. s. v. - - -
  8 //MELDING DD DSN=PPkkk.S0000.ini.MELDING,
- - - o. s. v. - - -
  9 //REFERAN DD DSN=PPkkk.S0000.ini.REFERAN,
- - - o. s. v. - - -
 10 //DATABESK DD DSN=PPkkk.S0000.ini.DATABESK,
 11 //TABINN DD DSN=PPkkk.S0000.ini.TABINN,
 12 //TABELLER DD DSN=PPkkk.S0000.ini.TABELLER,
 13 //KORREK DD DSN=PPkkk.S0000.ini.KORREK,
- - - o. s. v. - - -

```



## 4.1.2 VALG '1'

, , , , , , , , , , , , , , , ,

VALG ==&gt; 1

Petter Smart\_\_\_\_\_ Navn på mottaker av lister  
 nnnn STATISTIKNR  
 VELG EN AV FØLGENDE RUTINER:

```
----- -member- -----
1  TEKSTER  test01__  Lage/endre tekster og tabellparametre
2  REFERAN  _____  Ekstrahere tabellkomponenter
```

, , , , , , , , , , , , , , , ,

Her er membertavnet kalt 'test01' og dette member leses inn i edit.  
 Er det et nytt member vil bildet se slik ut:

, , , , , , , , , , , , , , , ,

```
EDIT ---- PPkkk.S0000.ini.TEKSTER(TEST01) 01.00 ----- COLUMNS 001 072
COMMAND ==>                                         SCROLL ==> HALF
```

```
***** ***** top of data *****
```

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

, , , , ,

```
***** ***** bottom of data *****
```

, , , , , , , , , , , , , , , ,

Så er det bare å skrive inn hvordan tabellen(e) skal se ut med utgangspunkt i kap. 2.1 tekstfilen.

Husk å sette inn et navn (medlemnavn) i feltet første gang du er inne på de forskjellige datasett, ellers vil det komme en feilmelding om at datasettet er tomt (meldingen er kryptisk).

Etter å ha åpnet datasettet første gangen kan følgende tips spare en for mye ergelser.

- skriv 'profile' etter 'COMMAND ==>'
- skriv 'recovery on' i det samme felt
- skriv deretter 'reset' (i samme felt)

Det tar ofte lang tid å lage en tekstfil for store tunge tabeller. Er du så heldig å bli sittende for lenge i en telefon eller blir heftet på annen måte, vil skjermen din 'falle ut', og du står der du var da du åpnet datasettet. Jeg har ved to anledninger mistet ca. en halv dags telling av posisjoner for å få alt på riktig plass slik den forrige publikasjonen var.

'Recovery on' vil bringe deg tilbake til der du var da skjermen falt ut. Dette tipset kan med hell brukes til alle datasett du måtte ha, ikke bare i Printertab!

#### 4.1.3 VALG '2'

VALG ==> 2

Petter Smart \_\_\_\_\_ Navn på mottaker av lister

nnnn STATISTIKKNR

VELG EN AV FØLGENDE RUTINER:

```
----- -member- -----
1  TEKSTER  _____  Lage/endre tekster og tabellparametre
2  REFERAN  test01__  Ekstrahere tabellkomponenter
```

Her er medlemnavnet det samme, og det kan være greit at alle komponentene til en og samme tabell har samme navn.

Editeringen er som for tekster. Det kan være greit å bruke et gammelt member som mal. Finn et passende member som i grove trekk inneholder det du trenger. Åpne først det nye memberet i EDIT, skriv deretter:

COMMAND ==> copy glmember

hvor 'glmember' er navnet på det memberet vi ønsker å hente inn og bruke som mal.

#### 4.1.4 VALG '3' OG '4'

Det er ingen forskjell fra de to foregående valg. Har du f.eks. en filbeskrivelse i et Easytrieve-program, kan du kopiere dette over ved kommandoen 'COPY'. Nederst på bildet 'EDIT - COPY' kan du angi linjenummer (f.o.m.-t.o.m.) du vil ha med. Nummertype er alltid 'standard' (NUMBER TYPE ==> s). Etter kopieringen kan du justere beskrivelsen slik at den stemmer med oppsettet under kap. 2.3 (s. 17).

## 4.1.5 VALG '5' '6' OG '7'

Disse prosedyrene er til for de som har kjørt printertab på den gamle måten, altså uten PP-datasett. Endringene til den nye varianten (som vi holder på med) innebærer enklere utfylling og kjøring. For å vise forskjellen, samt de som holder på gamle-måten, tar med den gamle menyen.

, , , , , , , , , , , , , , , ,

## P R I N T E R T A B - menybilde

VALG ==&gt; 5

```
-----
5 INTERPAR           Kjøre preprocessor
6 INTERTAB          Kjøre tabellprogram
```

, , , , , , , , , , , , , , , ,

Før en kjører preprosessoren må de tre øverste (tekster, referan og databesk) inneholde navn på medlemmer (disse må også eksistere og inneholde de parametre som er nødvendig for å lage en tabell). Vi får følgende skjermbilde fram:

, , , , , , , , , , , , , , , ,

## KJØRE PREPROSESSOR

KOMMANDO

```
____ - Statistikknummer
_     K - Kjøre preprocessor
_     S - Se meldinger
```

```

                fil-identer:
                Project  Library  Type      Member
Tekster        Kkkkini  test____  filer____ member__
Referanser     Kkkkini  test____  diverse_  member__
Databesk       Kkkkini  test____  diverse_  member__
```

BRUK END (PF3/PF15) FOR Å GÅ TILBAKE TIL FORRIGE MENYBILDE

, , , , , , , , , , , , , , , ,

Her finner vi igjen navnet på medlemmet som skrevet inn i forrige menybilde. For å kjøre preprosessoren skrives 'k' i feltet under statistikknummeret, deretter 's' for å se på meldingene.

Etter at preprosessoren er kjørt og meldingene viser at alt er ok, velger vi '6' fra hovedmenyen for å starte selve tabellprogrammet.

KJØRE TABELLPROGRAM

KOMMANDO

\_\_\_\_\_ - Statistikknummer  
 - K - Kjøre tabellprogrammet  
 - S - Se den ferdige tabellen

Fil-ident datafil:

\_\_\_\_\_

Navn på tabellsettet:

\_\_\_\_\_

BRUK END (PF3/PF15) FOR Å GÅ TILBAKE TIL FORRIGE MENYBILDE

Her skrives foruten navnet på selve tabellen (medlemnavn) også filidenten på datafilen. Deretter skrives statistikknummer og 'k' for å kjøre tabellprogrammet.

For å se/redigere den ferdige tabellen skrives 's'. Tabellen hentes da inn i EDIT slik at du her kan foreta de justeringer som måtte være ønskelig. Under EDIT finnes det kommandoer som tar vare på det meste du måtte ønske å gjøre med en tabell.

Boundaries 'bnds' sammen med copy/move og overlay gjør det forholdsvis enkelt å f. eks. bytte om på kolonner eller sette forspalten bakerst (til høyre) i tabellen.

For at du skal få tak i alle finessene i EDIT kan du bruke noe tid med HELP-funksjonen når du er i EDIT. Hvis du logger deg inn med en PC, har du muligheten til å få en utskrift av det bilde du har på skjermen ved trykke 'Print Scrn'.

Du kan finne mye interessant i en manual som heter 'ISPF 2 - 3 EDIT, EDIT MACROS' med undertittel 'SC34-4121-00'. Kap. 3 og 9 inneholder endel fikse kommandoer som kan lette arbeidet med redigering av filer og tabeller. Manualen får du fra BSS, og IBM's tittel på denne er 'Interactive System Productivity Facility/Program Development Facility, Version 2 Release 3, Edit and Edit Macros, MVS', kort og fyndig!



Til slutt kan korreksjonsprogrammet kjøres (hvis ikke tabellen rettes opp i 'edit'). Etter valg '7' i hovedmenyen vises:

, , , , , , , , , , , , , , , ,

KJØRE KORREKSJONSPROGRAM

KOMMANDO

\_\_\_\_\_ - Statistikknummer \_\_\_\_\_ Lokal printer id

- K - Kjøre korreksjonsprogrammet  
 S - Se den ferdige tabellen  
 U - Utskrift av den ferdige tabellen

Navn på tabellsettet(Membernavn): \_\_\_\_\_

Navn på korreksjoner(Membernavn): \_\_\_\_\_

Navn på korrigert tabellsett(membernavn): \_\_\_\_\_

PF1/PF13=HJELP \*\*\* ENTER=SEND JOB \*\*\* PF3/PF15=AVSLUTT(END)

, , , , , , , , , , , , , , , ,

Den gamle og den nye tabellen legges til det samme datasett og må derfor ha forskjellige navn! Husk å fylle inn printer-id (eks.: rmt5) før du printer ut. Statistikknummeret skal alltid med.

Et lite tips til disse tre:

For å få melding om at jobben er kjørt må 'enter' trykkes. Dette vil sende jobben en gang til hvis vi ikke setter et annet tegn enn de lovlige (K, S og U). Setter du f. eks. et punktum istedet, kan du trykke enter (du får feilmelding om at K, S og U ikke er brukt).

Den andre muligheten er å bruke splitscreen (PF2) og veksle mellom disse med swap (PF9) til SDSF (se 4.1.6 på neste side).

#### 4.1.6 VALG '8'

Alle som har kjørt jobber i batch vil kjenne SDSF. Her vil alle jobrapportene havne etter at jobben er kjørt. Et eksempel på en rapport finner du under punkt 4.1.1.

For å kunne åpne en rapport velges først '8' fra hovedmenyen. Deretter skrives 'h' etter 'command input' og en liste kommer opp over alle jobber som er kjørt. For å velge en av jobbene flyttes kursor i feltet 'np' og skriver 's' for 'select'. Jobben vil nå komme opp på skjermen som vist under punkt 4.1.1.

Det vil, ettersom man kjører tabeller, bygge seg opp en lang liste med jobbrapporter. For egen og andres skyld; fjern alle jobbrapporter som ikke lenger er aktuelle. Disse legger beslag på plass i noe som heter 'JES2' og her bestemmer plassen bl. a. for hvor mange som kan være logget på IBM/COMPAREX samtidig. Det er derfor en god regel å holde listen over jobbrapporter så kort som mulig!

De lovlige kommandoer i SDSF er:

DA - Viser aktive brukere av systemet  
 I - Viser jobber som ligger i inn-kø  
 O - Viser jobber som ligger i ut-kø  
 H - Viser jobber som er ferdig kjørt  
 WHO - Viser bruker-ID, logon-prosedyrenavn, og terminal-ID  
 END/PF3 - Avslutter SDSF  
 PF1 - Hjelp, informasjon på engelsk

Kommandoene under I, O og H er som i BROWSE. I tillegg har du mulighet til å bla opp og ned i de forskjellige deler av jobben med 'n' - next og 'p' - previous.

Er du usikker på hvordan kommandoene skal brukes så kan det lønne seg å bruke litt tid i HELP. Her kan du se gjennom deler eller hele avhandlingen om SDSF (dette kan ta litt tid).

#### 4.1.7 VALG 'E'

Dette er den versjonen som bør brukes idet denne ivaretar de nye forandringer som er lagt til, bl. a. PP-datasettene.

, , , , , , , , , , , , , , , ,

#### KJØRE PREPROSESSOR

KOMMANDO

\_\_\_\_\_ - Statistikknummer  
 - K - Kjøre preprossessor versjon 1 el. 2 \_  
 - S - Se meldinger

FILNAVN	KONTORNR.	INITIALER	MEMBER
Tekster	_____	_____	_____
Referanser	_____	_____	_____
Databeskrivelse	_____	_____	_____

PF1/PF13=HJELP \*\*\*\*\* ENTER=SEND JOB \*\*\*\*\* PF3/PF15=AVSLUTT(END)

, , , , , , , , , , , , , , , ,

Her finner du igjen to muligheter (versjoner) å velge mellom. Versjon 1 er den grunnleggende versjonen. Versjon 2, som de fleste bruker, har noen rutiner som kan være verdt å titte på.

- Max antall kolonner er utvidet til 50 (51 = SUM)
- Mulighet for å legge inn blanke ved tusen (eks.: 4 207 311)
- Mulighet for 'prikking' maskinelt.

Det henvises til punkt 4.2 hvor dette blir gjennomgått.

**4.1.8 VALG 'F'**

Her er det program som skaper tabellene vi har komponert. Som beskrevet tidligere setter preprosessoren tabelldelene sammen til en eller flere tabeller. Selve tabellprogrammet tar for seg datafilen, utfører de beregninger som er kodet, for så å legge tallene inn i tabellen(e).

For å starte tabellprogrammet velges 'F' fra hovedmenyen og det siste bildet i sekvensen ser vi her:

, , , , , , , , , , , , , , , ,

**KJØRE TABELLPROGRAM**

KOMMANDO

```

_____          _____          Lokal printer id
-   - Statistikknummer         Jobbklasse
-   K - Kjøre tabellprogram versjon 1 el. 2 _
    S - Se den ferdige tabellen
    U - Utskrift av den ferdige tabellen

```

Fil-ident datafil:

\_\_\_\_\_

Navn på tabellsettet:

\_\_\_\_\_

PF1/PF13=HJELP \*\*\* ENTER=SEND JOB \*\*\* PF3/PF15=AVSLUTT(END)

, , , , , , , , , , , , , , , ,

Her legger du inn hvilken printer du ønsker at tabellene dine skal komme ut på. Du bestemmer også hvilken jobbklasse du vil kjøre under. Husk; kjør under samme versjon som preprosessoren.

Navnet på tabellsettet bør velges med omhu, her kan du gi opplysninger om innholdet. Navn som 'tabella', 'tabell2' o.s.v. er ingen god løsning. Tabelldatasettet fylles fort opp og det gjelder å holde god orden allerede fra starten. Er du ikke fornøyd med navnevalgene, er det mulig å bruke 'R' (rename) under punkt 3.1 (LIBRARY UTILITY) i hovedmenyen.

#### 4.2. 'TIPS AND HINTS'

Det hender at enkelte ting ønskes utført maskinelt da tiden er knapp og alle maser. Derfor er det opp gjennom årene utviklet endel rutiner som tar vare på slike ønsker. Vi skal ikke her lage en komplett liste i det dette vil føre for langt (og heller forvirre enn forklare!). Disse hjelperutiner vil foruten å forenkle etterarbeidet også hjelpe maskinen (ressurssparing på alle bauger og kanter med andre ord).

##### 4.2.1 BLANKE FOR 'TUSEN'

Noen tabeller har relativt god plass og/eller store tall. Slike tabeller gjør seg utvilsomt bedre med mellomrom ved hver tusen. Tabellen blir også mer lettlest (eks.: 4123456 mot: 4 123 456). For å rette på dette blir en rutine satt igang ved å utelate 'D' i RED slik:

```
000001 01H1F1K1N1
000002 RE      13204
```

Alle tall i tabellen vil med denne rutine ta noe større plass enn tidligere, husk å ta hensyn til dette under oppstillingen!

##### 4.2.2 'PRIKKING'

I byråets publikasjoner er det endel standardtegn. Bl. a. skal felt i tabellene som er null skrives med strek (-) mens der hvor tall ikke kan forekomme skriver prikk (.). For å lette dette arbeidet noe er det også her laget en liten rutine!

Det bør her presiseres at rutinen ikke kan finne alle steder der prikking skal forekomme, kun der hvor det logisk skal forekomme!

I 6. posisjon i 'RED-kortet' setter vi inn bokstaven 'P' for 'prikking' og alle steder der tall ikke kan forekomme settes det inn en prikk. Redigeringsfilen ser da slik ut:

```
000001 01H1F1K1N1
000002 RE      P13204
```

Her blir både blanke ved tusen samt prikking utført. Det er selvfølgelig mulig å bruke bare en av mulighetene.

### 4.2.3 'LESETIDFORKORTEELSE'

For å forkorte lesetiden på datafilen, dette gjelder i særlig grad store datafiler, kan det legges inn koder. Dette kan fortelle at når data forekommer bør den ikke lete etter flere forekomster eller fortelle at en variabel må overstige/være lik/ikke overstige en konstant for at recorden skal leses (være med).

For å ivareta disse funksjoner bruker vi redigeringsfilen hvor disse seleksjonene kodes. Vi skal først sette kriterier for om recorden skal leses eller ikke. Dette foregår i posisjon 71-80 i første linje i redigeringsparametrene. Posisjon 71-74 inneholder startposisjonen i datafilen vi søker igjennom. Posisjon 75 forteller hvor langt feltet er (maks 4). Posisjon 76 gir kriteriet og posisjon 77-80 angir verdien seleksjonen skal rette seg etter. Bruk PF10/11 (PF22/23) for å bla 'sidelengs'.

```
***** ***** TOP OF DATA *****
=COLS> ----2----+----3----+----4----+----5----+----6----+----7----+----8
000001                                     00453<163
000002 09060906090609
***** ***** BOTTOM OF DATA *****
```

Vi ser her at i datafilen finnes et felt som starter i posisjon 45, feltet er 3 posisjoner langt og skal ha verdi lavere enn 163.

```
=COLS> : 7----+----8
000001 : 0045          Startposisjon på datafilen: 45
000001 : 00453        Lengden på feltet: 3
000001 : 00453<      Seleksjonskriteriet: skal være mindre enn
000001 : 00453<163   Seleksjonsverdien: 163
```

Seleksjonskriteriet kan være en av følgende:

```
= (er lik)          > (større enn)
< (mindre enn)     U (ulik)
```

I og med at det kun er tre posisjoner igjen (78 til 80) kan ikke seleksjonsverdien være større enn 999. I de tilfelle vi har større verdier som f. eks.: opp til 999 999, kan vi ved å endre startposisjonen på datafilen selektere på hele tusen (100 = 100 000).

Det finnes tre koder for å fortelle tabellprogrammet at en data-record (en linje i datafilen) ikke skal forekomme mer enn en gang.

```
000001 01H1F1K1N1          Hver datarecord kan bare forekomme
000002 REDU 13204          i en kolonne

000001 01H1F1K1N1          Hver datarecord kan bare forekomme
000002 RED X 13204        en gang i hver deltabell

000001 01H1F1K1N1          Hver datarecord kan bare forekomme
000002 RED Y 13204        i en deltabell
```

Disse koder kan redusere lese- eller søketiden med opp til 50 prosent. Dette vil spare ressurser når en kjører store tabeller eller serier av tabeller til f.eks. NOS. CPU-tid er noe vi alle har for lite av, så dette kan være ditt bidrag i så måte.

#### 4.2.4 OPPRETNING PÅ TABINN

Ved små feil oppdaget av preprosessoren, er det her mulig å gå direkte inn i 'TABINN' for retting. Dette krever kanskje en del kjenskap til printertab, men man kan spare mye tid ved å benytte denne snarveien ved store/kompliserte tabeller.

Alle navn som refererer til datafilbeskrivelsen er her byttet ut med posisjonsangivelser. Dette innebærer at kommune som starter i første posisjon i datafilen ikke heter:

0101 KOMM =0402                   Kongsvinger .....

Men derimot:

0101 00014=0402                  Kongsvinger .....

Som betyr at i første posisjon starter et felt som er 4 lang og skal selekteres om verdien = 0402.

Det samme finnes i kolonneberegningene. Der vil:

01                   SUMBEF80

Bli til:

01                   SUM00177

Som betyr at vi skal legge inn summen av et felt på datafilen som starter i posisjon 17 og er 7 lang i første kolonne.

Er du i tvil om hvordan du skal takle denne utfordringen, kan det lønne seg å lage en liten treningstabell hvor du kan trene på de forskjellige kommandoer, parameterstyringer o.s.v. Eller du kan kontakte Halvor Strømme eller Per H. Strand, Kontor for befolkningsstatistikk, Kongsvinger.

## 5. PLANLEGGING

For å gjennomføre en publikasjon med trykkeklare tabeller kreves planlegging før programmeringen starter. Er publikasjonen tidligere utgitt, spares mye arbeid med oppbyggingen av tabellene. Utforming av en tabell må skje i nøye samsvar med tabellregelverket, i tillegg skal den være lett å lese. Dette kan innebære endel testkjøring (bruk tall fra forrige publikasjon, fordi resultatet kan sjekkes). Derved får du også et bilde av hvordan datafilen(e) bør bygges opp.

Som nevnt tidligere er oppbyggingen av datafilen viktig. I størst mulig utstrekning bør det skreddersys datafiler enten til en tabell eller en tabellserie. Der det er mulig bør en 'sammenslåing' av flere datafiler finne sted. Til dette arbeidet er Easytrieve Plus erfaringsmessig det lettest tilgjengelige program for denne form for 'fil-massering'. Det er bedre å lage ti småfiler enn en stor og uoversiktlig fil. Det er også viktig at disse aggregeres opp til det nivå tabellene har som grunnenhet. Dette vil spare kjøretiden betraktelig når man samtidig bruker de kodene som nevnt under punkt 4.2.4.

Det bør også kjøres ut listinger eller testkjøringer på en eller flere av de momentene som inngår i tabellene for å kontrollere selve bearbeidingen av tallene i tabellprogrammet. Dette kan gjøres samtidig med opprettingen av datafilene. Foruten å teste selve data-grunnlaget kan man lettere finne fram til eventuelle feil i tabellparametrene.

Selve tekstdelen av tabellene bør korrekturleses og godkjennes så langt det er mulig før disse settes i produksjon. Det er også mulig å trekke tekstbehandlingskontoret inn i denne del av produksjonen.

Siden tabellprogrammet legger beslag på store maskinressurser, vil det lønne seg å legge kjøringene til tidspunkter hvor belastningen på maskinen er minimal. I enkelte tilfeller kan det være på plass å vurdere overtidsarbeide i forbindelse med kjøringene.



## 6. HUSKELISTE

De fleste vil få feilmeldinger i starten, og det er ikke alltid like lett å finne årsaken. Det er ikke engang sikkert at du får feilmelding, tabellen er tom, eller den er ikke der i det hele tatt. Her kommer derfor en liste med momenter som er verdt å huske under arbeidet med de forskjellige filer.

- STORE bokstaver på de deler som ikke er tekst til tabellen.
- Husk å sjekke linjenummer, stigende rekkefølge, tekst over flere linjer skal ha samme linjenummer og seleksjonen SKAL ligge på den siste av disse linjer. Har du også husket å forandre deltabelnr. (to første posisjoner i linjenr.)?
- Se etter at redigeringsfilen ikke inneholder seleksjoner som fulgte med når du kopierte over fra et annet member. (pkt. 4.2.4).
- Har du gitt riktig medlemnavn på de forskjellige delene?
- Har du riktig navn på datafilen?
- Stemmer datafilens navnsetting med de som brukes i tekstfil? (eks. kommune er KOMM i tekst og KOM i beskrivelse)
- Er posisjonene i databeskrivelsen i overenstemmelse med datafilen?
- Inneholdet i datafilen - er tallene riktige? - er det tall der i det hele tatt? (test med f.eks. Easytrieve Plus)
- Beregningene - 'K' for konstanter ved f.eks. prosent.
  - Er rekkefølgen på beregningene i en linje/kolonne riktig?
  - Er det riktig linje/kolonne det beregnes på?
  - Står de forskjellige parametre på rett plass?
- Er det riktig forspalte/-seleksjon som er definert i redigeringsfilen?

Disse momentene er erfaringsmessig der det stopper opp for de fleste. Ved å teste de forskjellige momenter som kan skape problem dekker denne lista minst 90% av tilfellene.

Til slutt må det opplyses at produktet printertab er utviklet ved Byrået av førstekonsulent Halvor Strømme, Kontor for befolkning. Han er også behjelpelig ved problemer etc.