

Interne notater

STATISTISK SENTRALBYRÅ

88/10

13. mai 1988

EN KORT INNFORING I IBM'S JCL (JOB CONTROL LANGUAGE)

AV

LIV DAASVATN

INNHOOLD:

	side
Generelt om JCL-oppsett	1
Eksempel på et enkelt JCL-oppsett	3
Noen ord om JES2-kort	5
Eksempel på et litt større JCL-oppsett	7
Om å opprette en fil i batch	8
Om å hekte sammen datasett	12

FORORD

Etter at jeg i flere år har holdt Byråskolens kurs i SAS og forøvrig hjulpet mange EDB-brukere i nød, er det min erfaring at JCL er den store bøygen for mange IBM-brukere. Som et resultat av dette, har brukere i stor stil 'arvet' gamle JCL-oppsett og/eller kopiert hverandres. Det sier seg selv at JCL-oppsettene da kan være lite hensiktsmessige (av og til riv ruskende gale ...).

Jeg håper at disse få sidene kan klare litt opp i en (tilsynelatende) innviklet JCL-verden.

Kommentarer mottas med takk!

Liv Daasvatn
Systemkontoret, Oslo

GENERELT OM JCL-OPPSETT:

Når du skal kjøre en batch-job på IBM, trenger du et språk for å kommunisere med maskinen. Du må jo oppgi navnet på input-filene, hvor programmet er, hvor resultatene skal legges osv. Det er dette vi bruker Job Control Language (JCL) til.

JCL består i hovedsak av 3 forskjellige instruksjoner ('kort') med tilhørende tilleggsopplysninger (parametre). De 3 instruksjonene er JOB-kortet, EXEC-kort og DD-kort. (Egentlig er det 8 forskjellige 'kort', men de 5 andre er ikke så mye brukt som disse 3.)

JCL har sin egen syntaks, hvor disse reglene er viktige:

- * Alle linjer må begynne med // i kolonne 1 og 2.
- * Små bokstaver er forbudt! (Unntatt i tekststrenger.)
- * Parametre adskilles med komma, ikke blanke tegn.
- * Det som står i kolonne 72 og utover blir ignorert.
- * Hvis en instruksjon skal fortsette på neste linje, må dette markeres med et komma som siste tegn på linjen. Fortsettelseslinjen må da ikke ha mer enn 13 blanke tegn mellom // i kolonne 1 og 2 og selve fortsettelsen.

I JOB-kortet gir du blant annet opplysninger om hvem du selv er (altså brukeridenten din), hvilket statistikknummer jobben skal føres på (for eventuelt regnskap over ressurs-forbruk), om jobben krever bånd-montering (hvis f.eks input-data ligger lagret på tape), osv. JOB-KORTET SKAL STÅ ALLER FØRST I JOBBEN, OG DET MÅ ALLTID VÆRE MED. Skjelettet til et JOB-kort ser slik ut:

```
//jobident JOB parametre
```

I EXEC-kortet oppgir du navnet på programmet eller prosedyren du skal bruke. Det må alltid være med minst ett EXEC-kort. DET MÅ VÆRE MED ETT EXEC-KORT FOR HVER 'AKTIVITET', DVS HVERT 'JOB-STEP' DU HAR I JOBBEN. EXEC-kortet til det første steppet kommer rett etter JOB-kortet. Skjelettet til et EXEC-kort ser slik ut:

```
//stepnavn EXEC parametre
```

I DD-kortene gir du opplysninger om alle filer (datasett) som er involvert i jobben. DET MÅ VÆRE ETT DD-KORT FOR HVER FIL SOM BRUKES. I en jobb med flere job-step (og dermed flere EXEC-kort), må en passe på at filene som brukes i hvert enkelt job-step får DD-kortene sine på rett sted (altså etter riktig EXEC-kort).

Skjelettet til et DD-kort ser slik ut:

```
//DD-navn DD parametre
```

De andre 5 JCL-kortene er disse:

```
//*           Kommentar-kort.
//           Avslutter en jobb (ofte unødvendig).
/*           Markerer slutten på innskutte data.

//..... PROC       Start på en JCL-prosedyre.
//..... PEND       Slutt på en JCL-prosedyre.
```

Manualen som beskriver JCL heter MVS/370 JCL (GC28-1350-2). Denne manualen er svært oversiktlig og lett å finne frem i, i motsetning til mange av de andre IBM-manualene. Det som imidlertid (selvfølgelig) ikke står i denne manualen, er hva vi her i Byrået har satt som default-verdier på visse parametre. (Defaultverdien vil si den verdien parameteren settes til dersom du utelater den helt). Defaultverdiene settes av systemprogrammererne.

EKSEMPEL PÅ ET ENKELT JCL-OPPSETT:

Eksempelet viser et JCL-oppsett for å kjøre SAS. (SAS er et av Byråets programmeringsspråk/analyseverktøy.)

```
//0414LDAA JOB 1361,'Liv, tlf 2068',NOTIFY=0414LDA,
//          CLASS=A,MSGCLASS=X,MSGLEVEL=(2,0)
//STEP1    EXEC SAS,OPTIONS='GEN=0 PS=46 NOCENTER'
//SASFIL   DD DSN=TS414.S1361.LDA.HELSEFIL,DISP=OLD
//SYSIN    DD *
```

```
DATA aids;
  SET sasfil.syke;
  IF diagnose = 27;
  .....
```

→ Dette er selve SAS-programmet, altså ikke JCL.

I dette eksemplet er det 1 JOB-kort, 1 EXEC-kort og 2 DD-kort. JOB-kortet går over 2 linjer, derfor avsluttes første linje med et komma. Alle linjer i JCL-oppsettet starter med // .

Kommentarer til JOB-kortet:

Det som står rett ut for // i første linje, er jobidenten. Hos oss er jobidenten lik brukeridenten din, pluss ett (fritt valgt) tegn. Så kommer JOB, som sier at dette er et JOB-kort. Deretter kommer JOB-kortets parametre, hvorav den første er jobbens statistiknummer (1361). Neste parameter er en tekststreng hvor en oppgir navn el.l. på den som er ansvarlig for jobben (nesten alltid deg selv).

I NOTIFY-parameteren oppgis til hvilken brukerident meldingen om jobben skal sendes (som regel din egen brukerident). Meldingen om jobben kommer opp på skjermen når jobben er ferdig. Den sier om jobben har gått bra eller ikke.

CLASS-parameteren angir jobbens jobbklasse. I jobbklassen oppgir vi om jobben krever båndmontering eller om den krever ekstra mye CPU-tid:

```
CLASS=T    Jobben krever båndmontering.
CLASS=D    Jobben krever mer enn et halvt CPU-minutt.
CLASS=H    Som CLASS=D, men jobben kan kjøres om kvelden/natten.
CLASS=A    Alle andre jobber.
```

Neste parameter i eksempelet er MSGCLASS-parameteren. Den kan hos oss være enten X eller A. MSGCLASS=X betyr at rapporten maskinen lager om jobben (kjører rapporten) legges i en kø (i pkt.8:h i ISPF) der vi kan se på den (lese feilmeldinger, blant annet!) og eventuelt ta den ut på papir. Setter vi imidlertid MSGCLASS=A, vil kjører rapporten bli sendt direkte til printer. Dette frarådes!!!

I MSGLEVEL-parameteren angir vi hvor mye opplysninger vi vil ha i kjører rapporten. Kjører rapporten inneholder all slags informasjon og statistikk om selve jobben: CPU-forbruk, generert JCL, allokeringer osv osv. (Den blir ofte fryktelig lang!) MSGLEVEL=(0,0) gir et minimum informasjon: kun JOB-kortet blir skrevet ut. Med MSGLEVEL=(2,0) blir også EXEC- og DD-kort skrevet ut. MSGLEVEL=(2,0) vil som oftest gi en passende mengde opplysninger. MSGLEVEL=(1,1) gir maksimal informasjon. MSGLEVEL=(1,1) gir sidevis med informasjon en ofte ikke greier å dechifrere, men det er denne verdien som er default.

NYTTIG HINT!

En nyttig parameter til JOB-kortet er TYPRUN=SCAN. Hvis du bruker denne parameteren vil JCL-oppsettet ditt bli sjekket for syntaksfeil. Ved å bruke TYPRUN=SCAN når du er usikker på om JCL-oppsettet er riktig, vil du kunne unngå ressurskrevende feilkjøring. Men husk å fjerne TYPRUN=SCAN når JCL'en er feilfri og du skal kjøre jobben!!

Kommentarer til EXEC-kortet:

Det som står rett utfor // i EXEC-kortet kalles step-navnet. Det velges fritt, og er nyttig dersom man har en jobb med flere aktiviteter (og dermed flere EXEC-kort). Går en jobb galt, vil man få vite hvilket step som har gått galt, og debuggingen (feilsøkingen) blir lettere. (Har man bare ett step i jobben, er det unødvendig, men helt uskadelig, med et steplavn). Etter step-navnet kommer EXEC, som angir at dette er et EXEC-kort. SAS er en katalogisert prosedyre, derfor er det nok å skrive EXEC SAS. (At SAS er en katalogisert prosedyre, betyr at det JCL-oppsettet som trengs for å kjøre selve SAS'en ligger ferdig lagret på et prosedyrebibliotek.) Hadde SAS istedet vært lagret som et program i et programbibliotek, hadde en måttet skrive EXEC PGM=SAS istedenfor bare EXEC SAS .

Det som står etter EXEC SAS i eksempelet på side 3, er spesifikasjoner (valg, options) som er spesielle for SAS og som kan variere for hver enkelt kjøring. Med andre prosedyrer eller program vil det være noe

annet (eller ingenting) som står her.

Kommentar til DD-kortene:

DD-kortene beskriver alle filer (datasett) som er involvert i jobben. DD står for Data Definition. Det som står rett utfor // i et DD-kort, kalles DD-navnet til den filen som DD-kortet beskriver. DD-navnet er et slags forkortet navn på filen. Det brukes som bindeledd mellom programmet og JCL'en, så du i programmet skal slippe å skrive et fullstendig filnavn hver gang du skal referere til denne filen. DD-navnet til en fil kan du så godt som alltid velge selv. DD-navnet må starte med en bokstav, men kan ellers bestå av både bokstaver og tall. Det må ikke være mer enn 8 tegn langt. (Æ, Ø eller Å er ikke tillatt i DD-navn.) Det er kun i spesielle tilfelle at en fil må ha et bestemt DD-navn. SYSIN og SYSOUT er slike spesielle DD-navn. SYSIN er reservert program eller innskutte data. I eksempelet står det en * etter DD. Stjernen betyr at programmet følger umiddelbart etter SYSIN-kortet.

Det første DD-kortet i eksempelet på side 3, har foruten DD-navnet og DD også med parametrene DSN= og DISP=. I DSN-parameteren oppgir man det fullstendige navnet på filen. I DISP-parameteren gir man opplysninger om filen: om den eksisterer, om den skal slettes når jobben er ferdig, om man kun har lov til å lese den, om man også har lov til å skrive på den osv. I eksempelet står det DISP=OLD. Det betyr at filen eksisterer, at vi både kan lese fra den og skrive til den, og at filen skal beholdes når jobben er ferdig.

NOEN ORD OM JES2-KORT:

Hittil har vi sagt at alle JCL-kort starter med // i 1. og 2. kolonne. Det er strengt tatt riktig, men i svært mange JCL-oppsett vil en støte på kort som har /* i 1. og 2. kolonne (og dette er ikke JCL-kort som markerer slutten på innskutte data). Det dreier seg om, er JES2-kort. JES2 er den delen av operativsystemet som kontrollerer jobbenes input og output. Det vil i de aller fleste tilfelle bare være et par JES2-kort som er aktuelle for oss: /*ROUTE PRINT-kortet og /*JOBPARM-kortet. Disse to kortene skrives rett etter JOB-kortet. Syntaksreglene til JES2-kortene er de samme som til JCL-kortene.

I /*ROUTE PRINT-kortet angir vi hvilken printer vi vil at utskriften fra jobben skal sendes til. Kortet vil som regel se slik ut:

```
/*ROUTE      PRINT RMTnnn
```

RMTnnn er 'adressen' til den aktuelle printeren (nnn er et tall). Under følger en liste over hvilken slik 'adresse' de forskjellige IBM-printerne har pr idag:

I OSLO:

'ADRESSE':	HVOR STAR DEN?	BRUKES AV:
RMT6	Rom 628/630	EDB-avdelingen
RMT7	Rom 538 B	Intervjukontoret
RMT8	Rom 716	3. kontor
RMT10	Driftskontoret	Alle
RMT12	Rom 402	4. kontor
RMT15	Driftskontoret	Alle
RMT16	Biblioteket	Biblioteket

PÅ KONGSVINGER:

'ADRESSE':	HVOR STAR DEN?	BRUKES AV:
RMT4	Rom C 301	
RMT5	Rom C 201	
RMT9	Driftskontoret	Alle
RMT11	Rom C 101	
RMT15	Tekstbeh.kontor	Tekstbeh.kontor

RMT10 er en skjønnskriver, og RMT15 er en laserskriver. Hvis man ikke har med noen opplysninger om hvilken printer som skal brukes, vil eventuelle utskrifter komme på hovedprinteren som står på Driftskontoret.

Dersom du er pålogget Kongsvinger-maskinen, men ønsker at utskriften skal sendes til en printer i Oslo, må du skrive N331 foran printeradressen (f.eks N331.RMT8). Er du pålogget Oslo-maskinen, men vil ha utskriften på Kongsvinger, må du istedenfor N331 skrive N330.

I /*JOBPARM-kortet har vi mulighet til å oppgi hvor mange linjer vi vil ha pr side i utskriften fra jobben (LINECT-parameteren). Dersom utskriften er på mer enn 10 000 linjer, må dette angis i en LINES-parameter (angis i 1000 linjer). /*JOBPARM-kortet kan f.eks se slik ut

```
/*JOBPARM      LINECT=46,LINES=20
```

EKSEMPEL PÅ ET LITT STØRRE JCL-OPPSETT:

Eksempelet under viser JCL-oppsett for å kjøre Easytrieve+. Easytrieve+ er et verktøy som brukes mye til filbehandling.

```
//0414KU1A JOB 8019,'Kurs i EZT+',MSGLEVEL=(0,0),MSGCLASS=X,
//          CLASS=D,NOTIFY=0414KU1,TIME=1,REGION=2048K
/*ROUTE      PRINT RMT6
//EZTSTEP    EXEC PGM=EZTPA00
//STEPLIB    DD DSN=EZT.LINKLIB,DISP=SHR
//ALDER      DD DSN=TK414.S8019.EZTKURS.DATAFIL,DISP=SHR
//KOMMTXT    DD DSN=TK414.S8019.EZTKURS(KOMMTXT),DISP=SHR
//EZTVFM     DD UNIT=WORK,SPACE=(4096,(100,200),,ROUND)
//SORTWK01   DD UNIT=WORK,SPACE=(4096,500,,ROUND)
//UTFIL      DD DSN=TK414.S8019.EZTKURS.UTFIL,DISP=OLD
//SYSIN      DD DSN=TK414.S8019.EZTKURS(PROGRAM),DISP=SHR
//SYSPRINT   DD SYSOUT=*
//
```

Kommentarer til dette JCL-oppsettet:

I JOB-kortet er MSGLEVEL-parameteren satt til (0,0). Det gir minimum informasjon. I dette eksempelet er det også med en TIME-parameter. Den angir at jobben maksimalt vil ta 1 CPU-minutt. Dette er årsaken til at CLASS-parameteren er satt til D (CLASS=A som i det forrige eksempelet er kun for jobber som bruker mindre enn et halvt CPU-minutt). I JOB-kortet er det også med en REGION-parameter, som sier hvor mye plass i maskinens internminne jobben vil kreve. Vanligvis vil default-verdien for REGION-parameteren være tilstrekkelig for jobben.

/*ROUTE PRINT-kortet er med for å styre output til en bestemt printer, nemlig RMT6.

I EXEC-kortet står det EXEC PGM=EZTPA00. Dette fordi Easytrieve+ ikke er lagt inn i et prosedyrebibliotek, men foreligger som et program. Av samme grunn må vi ha med et STEPLIB-kort. STEPLIB-kortet refererer til det bibliotek der programmets load-modul ligger. (Load-modulen er den kompilerte og linkete versjon av programmet.) STEPLIB-kortet må komme rett etter EXEC-kortet. STEPLIB er et reservert DD-navn.

De to DD-kortene som følger etter STEPLIB-kortet (DD-navn ALDER og KOMMTXT), tilhører data som skal behandles i programmet. Disse data har programmet kun lesetillatelse til (DISP=SHR).

De to neste DD-kortene (DD-navn EZTVFM og SORTWK01) allokerer midlertidig plass på et arbeidsområde. Hvordan man gjør dette (i de tilfellene det er nødvendig), er som oftest beskrevet i manualene som gjelder for de enkelte programpakker. I slike tilfelle kan det også ofte være snakk om reserverte DD-navn.

Det neste DD-kortet i eksempelet gjelder en fil med DD-navn UTFIL. Av DISP-parameteren kan vi se at filen eksisterer fra før, og at programmet både har lese- og skrive-tillatelse på den. Utfra JCL-oppsettet er det da nærliggende å tro at dette er en fil som Easytrieve-programmet bruker til å legge resultater ut på.

Kortet med DD-navn SYSPRINT gjelder utskrift fra programmet (SYSPRINT er et reservert DD-navn i Easytrieve+). SYSOUT=* betyr at utskriften styres til kjørerapporten.

I dette JCL-oppsettet leses programmet inn fra en fil, det ser vi av SYSIN-kortet.

Den siste JCL-linjen som kun inneholder // avslutter JCL-oppsettet.

OM Å OPPRETTE EN FIL I BATCH:

Dersom man skal opprette en fil i batch, må man foruten DD-navn, datasettnavn og DISP-parameter ha med 3 parametre til: DCB-parameteren, SPACE-parameteren og UNIT-parameteren.

Først litt mer om DISP-parameteren:

DISP-parameteren har 3 underparametre. De sier noe om filens status ved job-stepets start, hva som skal skje med den ved job-stepets

slutt, og hva som skal skje med den dersom job-stepet aborterer. Altså slik:

DISP=(start,slutt,abort) .

1.underparameter kan være NEW, SHR, OLD eller MOD. NEW er default, det vil si at dersom man ikke har med 1.underparameter, vil maskinen tro at filen ikke eksisterer og at den derfor skal opprettes. Hvis filen allerede eksisterer, kan en bruke SHR, OLD eller MOD som 1.underparameter. DISP=SHR betyr at man kun kan lese fra filen. DISP=OLD innebærer at man både kan lese fra og skrive til filen (oppdatere den). DISP=MOD betyr også at man både kan lese fra og skrive til filen, men med DISP=MOD blir det man skriver ut lagt til på slutten av filen. Filen vokser altså istedenfor å bli oppdatert.

2.underparameter har KEEP som default, altså at filen beholdes etter job-stepets (vellykkete) slutt. Ønsker man dette, kan man derfor kutte ut 2.underparameter. De andre mulighetene man har er DELETE, CATLG, UNCATLG og PASS. PASS betyr at filen skal brukes i et etterfølgende job-step.

3.underparameter har samme muligheter som 2.underparameter, så nær som PASS. DEFAULT-verdiene for 3.underparameter er avhengig av om filen allerede eksisterer eller om den skal opprettes. Hvis filen allerede eksisterer, er KEEP default. Hvis filen derimot ikke eksisterer, er DELETE default.

Dersom man velger defaultene for 2. og 3.underparameter, kan man la være å skrive dem. Dette er gjort i JCL-eksempelet på side 3. Har man bare med 1.underparameter, kan man også sløyfe parenteser rundt. I DISP-parameteren har du altså disse mulighetene til å angi en filstilstand (default-verdiene er understreket):

DISP=(start-tilstand,normal-slutt,unormal-slutt)

<u>NEW</u>	<u>KEEP</u>	<u>DELETE</u>
OLD	DELETE	<u>KEEP</u>
SHR	CATLG	CATLG
MOD	UNCATLG	UNCATLG
	PASS	

Dersom man ønsket å opprette en fil og katalogisere den (det er vanlig), kunne en skrevet dette slik: DISP=(,CATLG) . Dette ville

altså være det samme som å skrive `DISP=(NEW,CATLG,DELETE)` .

I DCB-parameteren (DCB=Data Control Block) kan man blant annet oppgi filens logiske recordlengde (LRECL), dens blokkstørrelse (BLKSIZE) og recordformatet (RECFM).

Blokkstørrelsen (BLKSIZE) er for filer med fast recordlengde et multiplum av recordlengden. (Er recordlengden variabel, kan blokkstørrelsen være et multiplum av den lengste recordlengden + 4.) For filer som skal lagres på disk av type 3375 (våre SSB-og BATCH-disker) lønner det seg å ha en blokkstørrelse så nær opptil 17300 som mulig. (Datasett som skal brukes i batch-kjøringer skal lagres på SSB- eller BATCH-diskene.) En fil med fast recordlengde har F som første tegn i recordformatet (RECFM). Er recordlengden derimot variabel, er første tegn i recordformatet V. Er filen blokket, skal andre tegn i recordformatet være en B. En vanlig, sekvensiell fil vil som regel ha recordformat fast og blokket (RECFM=FB). Dersom du skal lage en utskriftsfil som skal ha styrekarakterer for printer (for å angi side- og linje-skift), skal siste tegn i recordformatet være en A. En blokket fil med fast recordlengde som skal ha printer-styrekarakterer skal altså angis med RECFM=FBA.

Noen vanlige recordformat:

```
FB      (fast recordlengde, blokket)
FBA     (fast recordlengde, blokket, med styrekarakter)
VB      (variabel recordlengde, blokket)
VBA     (variabel recordlengde, blokket, med styrekarakter)
```

En DCB-parameter kan altså f.eks se slik ut:

```
//FEILFIL DD DSN=TK414.S6911.LDA.FEILFIL,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//          .....
```

I SPACE-parameteren oppgir du hvor mye plass som skal avsettes (allokeres) til filen. Første underparameter angir hvilken størrelsesenhet du vil angi plassen i. Enheten kan være et track (TRK), en sylinder (CYL) eller et antall tegn (som regel blokkstørrelsen). Det er som oftest mest hensiktsmessig å bruke blokkstørrelsen som enhet. Etter å

ha oppgitt størrelsesenheten, oppgir du to tall: 1. og 2. allokeringsmengde. Det er nemlig slik at når det avsettes plass til en fil, får du først én porsjon: 1. allokeringsmengde. Dersom dette er for lite til at filen får plass, får du én porsjon til (2. allokeringsmengde). Dersom ikke dette heller er nok, kan du få flere porsjoner å 2. allokeringsmengde (totalt 15 ganger).

SPACE-parameteren kan f.eks se slik ut: SPACE=(17292,(10,5),RLSE)

Her er allokeringseenheten 17292 tegn, og 1. allokeringsmengde er 10. Det vil si at det først avsettes 10 ganger 17292 tegn. Dersom dette ikke er nok, vil 2. allokeringsmengde (5*17292 tegn) avsettes inntil 15 ganger til. Denne filen vil altså maksimalt få $10*17292 + 15*5*17292$ tegn avsatt. RLSE som står på slutten av SPACE-parameteren, betyr at allokert plass som ikke er benyttet, skal frigjøres til andre formål. (Bruk RLSE-underparameteren! Det kan jo tenkes at filen ikke 'fyller ut' avsatt plass 100%.)

I UNIT-parameteren kan man angi om filen skal lagres på disk eller kassett. Man kan eventuelt oppgi navnet på en spesiell disktype eller disk. F.eks UNIT=CAS, UNIT=SSB eller UNIT=SSB003.

Et JCL-oppsett for å opprette en fil i batch kunne altså se slik ut:

```
//NYFIL DD DSN=TK414.S3497.LDA.BANKFIL,
//      DISP=(,CATLG),
//      DCB=(RECFM=VBA,LRECL=137,BLKSIZE=141),
//      SPACE=(141,(500,100),RLSE),
//      UNIT=SSB
```

Noen ganger kan det lønne seg å opprette temporære datasett. At et datasett er temporært, angis i DD-kortet med && foran datasett-navnet. En temporær fil vil kun vare så lenge kjøringen varer. Når man skal opprette en temporær fil, må man også ha med SPACE-, DCB- og UNIT-parametere. Et DD-kort for en temporær fil vil kunne se slik ut:

```
//TEMPFIL DD DSN=&&TEMP,UNIT=WORK,
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//      SPACE=(3120,(20,10),RLSE)
```

OM Å HEKTE SAMMEN DATASETT:

Av og til vil en ha behov for å hekte datasett etter hverandre (konkatenerer datasett). Dette kan for eksempel være i det tilfelle der data for én årgang ligger på én fil, og vi ønsker å behandle mange årganger under ett. Vi vil da lese inn alle filene (=årgangene) som om de utgjorde én stor fil. JCL-oppsett for å konkatenerer filer på denne måten kan for eksempel se slik ut:

```
//STORFIL DD DSN=P4243.S3432.I545A4A1.G8400.V00,DISP=SHR
//          DD DSN=P4243.S3432.I545A4A1.G8500.V00,DISP=SHR
//          DD DSN=P4243.S3432.I545A4A1.G8600.V00,DISP=SHR
//          DD DSN=P4243.S3432.I545A4A1.G8700.V00,DISP=SHR
```