JCL (Job Control Language)

Carte JOB

Définition de l'environnement général d'un job

Nom du JOB, du programmeur, classe d'exécution, classe de sortie, niveau des messages d'erreur, durée d'exécution, conditions d'exécution, conditions de reprise... :

```
//jobname JOB (compte),'programmeur',
//   CLASS=classe-d'entrée,
//   MSGCLASS=classe-de-sortie,
//   MSGLEVEL=(jcl,messages),
//   TIME=(mn,ss)|1440|NOLIMIT
//   REGION=nK|nM,
//   NOTIFY=userid|&SYSUID,
//   RESTART=(*[.step][.procstep]),
//   COND=[()](rc,opérateur)[,EVEN|
ONLY)],
//   TYPRUN=COPY|HOLD|JCLHOLD|SCAN,
//   GROUP=groupe-racf,
//   SECLABEL=label-racf,
//   USER=userid,
//   PASSWORD=(mdp[,nouveau-mdp]),
//   BYTES=(n[,CANCEL|DUMP|WARNING]),
//   CARDS=(n[,CANCEL|DUMP|WARNING]),
//   LINES=(n[,CANCEL|DUMP|WARNING]),
//   PAGES=(n[,CANCEL|DUMP|WARNING]),
//   RD=R|RNC|NR|NC,
//   PRTY=priorité,
//   PERFORM=n,
//   ADDRSPC=VIRT|REAL
```

Complément

jobname : Commence en colonne 3 par un caractère alphabétique pour une longueur max de 8 digits. Comporte souvent le userid.
compte : Dépend du site.
Programmeur : Nom du développeur et/ou information sur le job.

Un job comporte au maximum 255 steps.

Carte EXEC

Exécution de programme

```
//JOBLIB DD(1) DSN=loadlib,DISP=SHR(4)
//stepname EXEC PGM=nom-pgm,
//   [PARM='paramètre1',
//   REGION=nnK | nnM,
//   TIME=(mn,ss),
//   ACCT=compte,
//   COND=[()](rc,opérateur[,step])[,EVEN
|ONLY)],
//   ADDRSPC=VIRT|REAL
//   DPRTY=(nn[,nn]),
//   DYNAMNBR=nn,
//   PERFORM=n,
//   RD=R|RNC|NR|NC]
//STEPLIB DD(1) DSN=loadlib,DISP=SHR(4)
```

Execution de procédure

```
//[name] JCLLIB ORDER=(proclib[,...])(2)
//stepname EXEC [PROC]=nom-proc,
//   conste=valeur[,...]
```

Complément

stepname : Commence en colonne 3 par un caractère alphabétique. Longueur max: 8 digits.
nom-pgm : Nom du programme à exécuter (8 digits max). Pgm compilé, link-édité et dont le load est disponible en JOBLIB ou STEPLIB.
nom-proc : Nom de la procédure appelée.(3)

Carte PROC et PEND

Définition d'une procédure

```
//nom-proc PROC [conste=valeur[,...]]
//* ...
//   PEND
```

Carte SET

Définition de la valeur d'une constante

```
//[name] SET conste=valeur[,...]
```

Carte INCLUDE

Insertion d'un jcl dans un job ou une procédure

```
//[name] INCLUDE MEMBER=membre
```

Complément

membre : doit être présent dans le dataset déclaré en carte JCLLIB.(2)

Carte JCLLIB

Définition des PDS de PROC et INCLUDE

```
//[name] JCLLIB ORDER=(proclib[,...])
```

Carte DD (Data definition)

Définition des données (fichiers)

1 / Données en ligne (80 caractères)

```
//ddname DD *|DATA,DLM=$$
Ligne 1
Ligne n ...
/*|$$
```

2 / Données dans un fichier physique

```
//ddname DD DSN=datasetname[(membre)],
//    DISP=SHR|MOD|OLD|...(4)
//    autre-paramètres

autre-paramètres : SPACE, DCB, RECFM,
LRECL, BLKSIZE, LIKE, MGMTCLAS,
DATACLAS, STORCLAS,  REF, UNIT, VOLUME,
AVGREC, LABEL...
```

4 / Données sur une imprimante (Queue)

```
//ddname DD SYSOUT=classe,
//   DEST=imprimante,
//   COPIES=number,
//   HOLD=YES|NO,OUTLIM=lignes,
//   SEGMENT=pages,
//   FREE=CLOSE|END,SPIN=UNALLOC|NO
```

5 / Sans données

```
//ddname DD DUMMY
```

Concaténation

```
//ddname DD fic1(1)
//       DD fic2(1)
//       [...]
```

Override de fichier dans un jeu d'appel

```
//proc-stepname.ddname DD fic1(1)
```

Paramètre DISP

Disposition des fichiers

```
      NEW  KEEP    KEEP
      OLD  DELETE  DELETE
DISP=(SHR, CATLG,  CATLG)
      MOD  UNCATLG UNCATLG
                   PASS
        ❶     ❷      ❸
```

❶ Utilisation du fichier
❷ Disposition en fin normale
❸ Disposition en fin anormale

NEW : Nouveau fichier créé au step
OLD : Fichier existant dont on s'assure l'exclusivité
SHR : Fichier existant utilisable par d'autres JOB
MOD : Fichier auquel on peut ajouter des enreg.
KEEP : Fichier conservé après le step
DELETE : Fichier supprimé après le step
CATLG : Fichier catalogué après le step
UNCATLG : Fichier décatalogué mais pas supprimé
PASS : La disposition finale du fichier est déterminée par le step suivant qui utilise ce fichier.

Disposition par défaut

DISP absent : DISP=(NEW,DELETE,DELETE)
DISP=NEW : DISP=(NEW,DELETE,DELETE)
DISP=SHR : DISP=(SHR,KEEP,KEEP)
DISP=OLD : DISP=(OLD,KEEP,KEEP)
DISP=(NEW,PASS) : DISP=(NEW,PASS,DELETE)
DISP=(SHR,PASS) : DISP=(NEW,PASS,KEEP)

Complément

La disposition DELETE est uniquement possible si la période de rétention du fichier à supprimer est atteinte.

Légende

JOB : Carte
CLASS : Paramètre
mn : Valeur variable de paramètre
NOLIMIT : Valeur fixe de paramètre
conste : Constante
Ligne 1 : Données
| : Ou
[] : Facultatif
(1) voir carte DD
(2) voir carte JCLLIB
(3) voir carte PROC et PEND
(4) voir paramètre DISP

Utilitaires

IDCAMS

Utilitaire général

JCL d'appel

```
//stepname EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=classe
//ddname1 DD fic1
[//ddname2 DD fic2]
//SYSIN DD *
```

Supprimer un dataset

```
DEL[ETE] (datasetname[(membre)][,..])+
   ERAS[E]|N[O]ERAS[E]                +
   P[U]RG[E]|N[O]P[U]RG[E]            +
   F[O]RC[E]|N[O]F[O]RC[E]            +
   SCR[ATCH]|N[O]SCR[ATCH]
```

Lister les catalogues

```
LISTC[AT] ENTRIES(datasetname[,...]) +
   NAME|HISTORY|ALLOCATION|VOLUME|ALL+
   [O[UT]FILE(ddname1)]              +
   [CREATION(nombre-jours)]          +
   [EXPIRATION(nombre-jours)]
```

Définir un GDG

```
DEF[INE] GDG (
   NAME(nom-gdg)                     +
   LIM[IT](nombre-de-générations)    +
   EMP[TY]|N[O]EMP[TY]               +
   SCR[ATCH]|N[O]SCR[ATCH]           +
   [OWNER(texte)                     +
   TO(SSAAJJJ)|FOR(nombre-jours)])
```

Définir un dataset

```
DEF[INE] N[ON]VSAM (
   NAME(datasetname)                 +
   DEV[ICE]T[YPE](type[,...])        +
   VOL[UMES](volume[,...])           +
  [F[ILE]SEQU[E]N[CENUMBERS](n[,...])+
   TO(SSAAJJJ)|FOR(nombre-jours)]    +
   RCTLG|NRCTLG)
```

Imprimer un fichier SAM ou VSAM

```
PRINT I[N]FILE(ddname1)              +
   CHAR[ACTER]|DUMP|HEX              +
   [O[UT]FILE(ddname2)]
```

Conversion, réorg et copie de fichier

```
REPRO I[N]FILE(ddname1)              +
   O[UT]FILE(ddname2)                +
   REP[LACE]|N[O]REP[LACE]           +
   R[E]US[E]|N[O]R[E]US[E]
```

Commandes modales

```
SET MAXCC|LASTCC = nombre

IF LASTCC|MAXCC opérateur nombre
   THEN commande|DO liste-de-cmd END
  [ELSE commande|DO liste-de-cmd END]

CANCEL /* Halts processing */
```

opérateur : =, EQ, ^=, NE, >, GT, <, LT, >=, GE, <=, LE
nombre : 0, 4, 8, 12 ou 16

IEBGENER

Utilitaire de copie et reformatage

```
//stepname EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=classe
//SYSUT1 DD données-en-entrée
//SYSUT2 DD données-en-sortie
//SYSIN DD données-des-commandes

données-des-commandes :

GENERATE MAXNAME=n,
         MAXFLDS=n,
         MAXLITS=n,
         MAXGPS=n
[MEMBER NAME=(membre,[alias,...])]
[RECORD[IDENT=(l,'valeur',pos),]
    [FIELD=(l,pos1,conv,pos2),...]
    [FIELD=(l,'valeur',conv,pos2),...]]
```

l : Longueur
pos : Position
pos1 : Position en entrée
pos2 : Position en sortie
conv : Conversion, PZ (packé à étendu) ZP (étendu à packé)

SORT

Utilitaire de copie, tri, extraction et fusion

```
//stepname EXEC PGM=SORT
//SYSOUT DD SYSOUT=classe
//SORTINnn DD données-en-entrée
//SORTOUT DD données-en-sortie
//SYSIN DD données-des-commandes

données-des-commandes :

SORT|MERGE FIELS=
     (p,l,f,s)|p,l,s),FORMAT=f|COPY
    ,SKIPREC=n
    ,STOPAFT=n

INCLUDE|OMIT COND=
      (p,l,f,cond,
       p,l,f|constante[,AND|OR,...])

SUM FIELDS=(p,l,f[,...])|NONE
```

p : Position
l : Longueur
f : Format CH|ZD|PD|BI|FI...
s : Sens (Ascendant ou Descendant) A|D
cond : Condition EQ|NE|GT|GE|LT|LE
constante : n pour « occurrences de »
nX (n blanc), nC'...' (n char), nX'...' (n hex), nZ (n x'00')

IEBCOPY

Utilitaire de copie, fusion et compression de PDS

```
//stepname EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=classe
//SYSUT2 DD fic-de-travail
//SYSUT3 DD fic-de-travail
//ddname1 DD fic1
//ddname2 DD fic2
//SYSIN DD *
COPY OUTDD=ddname1,INDD=((ddname2[,R]),...)
[SELECT MEMBER=((nom[,newnom[,R]]),...) ]
[EXCLUDE MEMBER=(nom,...)]
```

On utilise soit SELECT soit EXCLUDE.

Légende

ERASE : Mot clé
NOERASE : Mot clé par défaut
datasetname : Valeur variable de paramètre
PZ : Valeur fixe de paramètre
IEBGENER : Nom programme
[] : Facultatif
| : Ou

Guillaume VENTRE
Http://documentation-mainframe.net
V1.1

Mémento JCL

Bibliographie :

- z/OS MVS JCL Reference – Publication No. SA22-7597-07
- z/OS DFSORT: Getting Started – Publication No. SC26-7527-00
- z/OS DFSORT Application Programming Guide – Publication No. SC26-7523-02
- DFSMS/MVS Version 1 Release 5 – Utilities – Publication No. SC26-4926-03