



VSAM

GESTION DES FICHIERS

25/01/2021

L'ORGANISATION VSAM

DÉFINITIONS

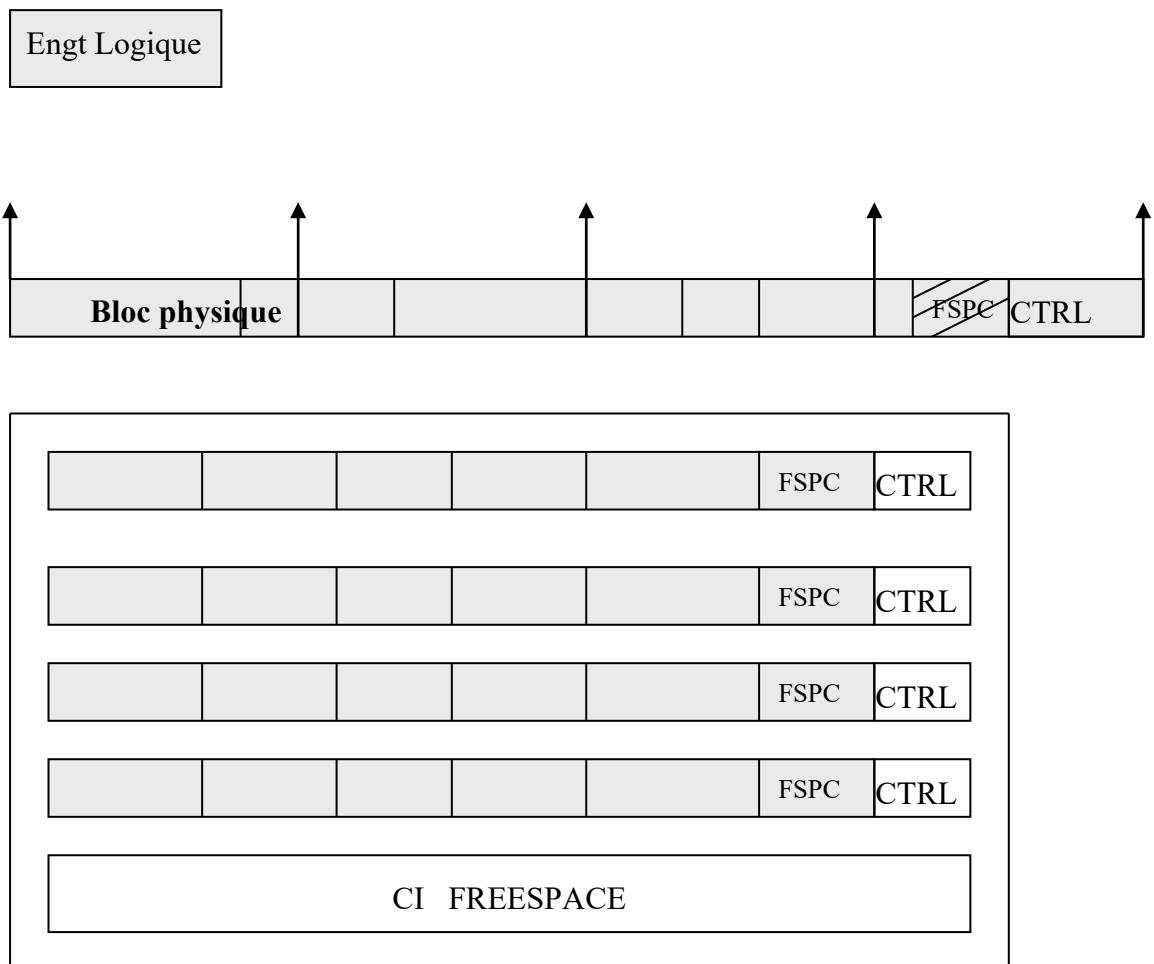
- VSAM
- enregistrement logique
- bloc physique
- Control Interval (CI)
- Control Area (CA)
- composant
- cluster

V.S.A.M

Virtual Storage Access Method

Méthode d'accès permettant de manipuler des fichiers sur disque

Structure logique :



1 Control Area

Control Interval (CI)

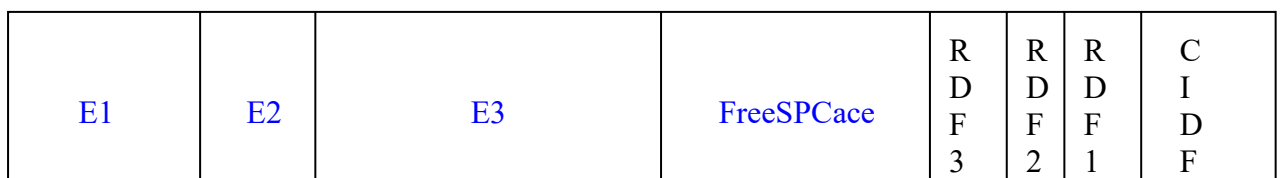
Unité de transfert entre le disque et la mémoire

Contient plusieurs enregistrements.

Les enregistrements "spanned" sont autorisés

CIDF et RDF

- **CIDF** : Control Interval Definition Field
 - Composé de 4 octets pour la gestion de l'espace libre
 - 2 octets : déplacement de l'espace dans le CI.
 - 2 octets : Longueur de l'espace.
- **RDF** : Record Definition Field
 - Composé de 3 octets pour la gestion des enregistrements logiques.



Enregistrement



Informations de contrôles

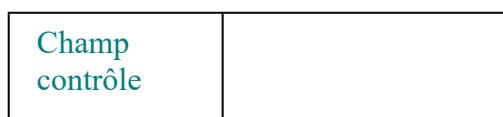
CIDF et RDF

- **CIDF**

- Un CIDF à 0 indique une fin de fichier.
- En ESDS le 1er CI non utilisé contient un CIDF à 0.
- En KSDS ou ESDS le 1er CI dans le CA non utilisé contient un CIDF à 0.

- **RDF** donne des indications sur :

- Les enregistrement spanned
- Les slots vides
- Longueur d'enregistrements ou segment
- Nombre d'enregistrements consécutifs



1 octet

2 octets

Exemple : X'00' : RDF associé à un seul enregistrement

Si plusieurs enregistrements de même longueur :

. X'00' : 1er RDF Nombre d'enregistrements

. X'40' : 2ème RDF Longueur enregistrement

CI et RBA

RBA (Relative Byte Address): adresse relative d'un CI ou d'un record par rapport au début du fichier

Taille d'un CI : $n * 0.5Ko$ ou $n * 2Ko$, avec $1 \leq n \leq 16$

0.5Ko, 1Ko, 1.5Ko, ... 7Ko, 7.5Ko, **8Ko, 10Ko, 12Ko, ... 28Ko, 30Ko, 32Ko**

CONTROL AREA (CA)

Unité d'allocation (d'1 piste à 1 cylindre)

Tous les CAs contiennent le même nombre de CIs.

ORGANISATION KSDS

Key Sequenced Data Set

Deux composants :

- composant data et
- composant index
- Enregistrements peuvent être de longueur variable
- Clé primaire pour l'indexation
même position et même longueur dans tous les enregistrements

FREESPACE

Espace à laisser libre lors du chargement

%Freespace/CI : espace à laisser libre dans chaque CI

%Freespace/CA : CIs libre (FSPC) dans chaque CA

ENREGISTREMENT SPANNED

Paramètre autorisant l'écriture d'un record sur plusieurs CIs

Pas d'autre record dans ces CIs

Pas de "spanned" sur CAs

Clé à l'intérieur du 1er segment

ORGANISATION ESDS

Entry Sequenced Data Set

Composant data uniquement

Pas de freespace

Enregistrement de longueur fixe ou variable

"Spanned" autorisé

Un fichier peut occuper un ou plusieurs volumes de même type.

Les enregistrements son chargés en séquence indépendamment de leur contenu.

ORGANISATION ESDS

- Options de traitement :
 - Récupération séquentielle des enregistrements
 - Accès direct possible
 - Ajout d'enregistrement en fin de fichier
 - Mise à jour d'enregistrement (au même emplacement et sans modifier la longueur)
 - Suppression d'enregistrements interdite
 - Possibilité d'avoir un AIX

ORGANISATION RRDS

Relative Record Data Set

Record de longueur fixe

Enregistrement vide ou plein

Accès par numéro (RRN = Relative Record Number)

ORGANISATION LDS

Linear Data Set

CI sans information de contrôle

Taille du CI : 4Ko.

Accès par fenêtre

RÉSUMÉ

	KSDS	ESDS	RRDS
Rangé par	clé	ordre d'entrée	RRN
Accès par	clé, RBA	RBA	RRN
AIX	oui	oui	non
Réorg.	oui	non	non
Taille d'engt	variable	variable	fixe
Spanned	oui	oui	-
index	oui	non	non
Split CI, CA	oui	non	non
Insertion	n'importe où	en fin	si slot vide
Suppres.	oui	non	oui
Modif. de longueur	oui	non -	-
Nbre d'I/Os	2 à 4	1	1

EXERCICE

1. Citez 5 types d'organisation VSAM :,,
2. Le terme utilisé pour décrire les composants de fichier VSAM est le
3. L'unité de transfert de données entre le disque et la mémoire est le ...
4. Le champ du CI utilisé pour déterminer l'espace restant libre est le et sa longueur est de .. octets.
5. Le champ du CI utilisé pour déterminer la longueur de l'enregistrement est le et sa longueur est de .. octets.
6. Les enregistrements sont rajoutés au CI de la vers la et les informations de contrôle de la vers la
7. Un enregistrement logique occupant plus d'un CI est appelé enregistrement et ses informations de contrôle ont une longueur de .. octets.
8. L'unité de disque occupée par 2 CIs ou plus est appelée ... dont la taille maximum est de
9. L'utilisateur peut définir la taille du .. mais VSAM détermine la taille du ... en fonction de
10. VSAM détermine la position d'un enregistrement en calculant son

DEFINITION DES CLUSTERS

SYNTAXE de la SYSIN

Ne rien écrire en colonne 1

Ne rien écrire après colonne 72

/* */ autour des commentaires

- + : continuation, après le dernier caractère non blanc

+ : continuation en milieu de mot

Séparateurs : blanc, virgule

; : fin de commande

Exemple :

```
DEFINE CLUSTER /*CREATION KSDS*/ -  
  ( NAME(SDJ.VS +  
      AM.KSDS1 )           -  
    CYL(5,1) VOL(VOL222)   -  
  )
```

SYNTAXE

Erreurs de syntaxe fréquentes :

- Caractère de continuation omis ou avant commentaires
- Caractère de continuation sur dernière ligne
- Nom de commande ou de paramètre omis
- Erreur de parenthèses
- Paramètres incompatibles
- Paramètre obligatoire manquant
- Ecriture en colonne 1

Access Method Services

```
//SDJFO1      JOB ...  
//            EXEC PGM=IDCAMS  
//SYSPRINT    DD ...  
//SYSIN       DD *  
...  
...  
/*
```

IDCAMS : nom du programme

SYSPRINT : nom du fichier de messages

SYSIN : fichier contenant les commandes à exécuter

FILE ou DATASET

FILE DDNAME / DATASET DSNAME

FILE : allocation faite par l'initiateur

Peut être requis

DATASET : allocation DYNAMIQUE faite par l'AMS.

CODE RETOUR

Code retourné par l'AMS

Code	Signification
0	Commande exécutée sans erreur
4	Commande exécutée avec "warning"
8	Commande exécutée avec erreur
12	Commande non exécutée
16	Commande non exécutée et arrêt de l'AMS

CATALOGUER UN NON VSAM

- en fin de création :

```
//      JOB      (compte),...
//      EXEC     PGM=IEFBR14
//DD1    DD      DSN=X.Y.Z,VOL=SER=SDJVOL,
//              DISP=(NEW,CATLG),
//              UNIT=3390,SPACE=(TRK,10)
//      ...
```

- en différé :

```
//      JOB      (compte),...
//CREAT    EXEC   PGM=IEFBR14
//DD1      DD     DSN=SDJ.FICH.VSAM,VOL=SER=SDJVOL,
//          SPACE=(TRK,(5,5)),DISP=(NEW,KEEP),UNIT=3390
//          ...
//CATAL    EXEC   PGM=IDCAMS
//SYSPRINT DD     SYSOUT=*
//SYSIN    DD     *
//DEFine    NonVSAM      -
//          (NAME(SDJ.FICH.VSAM)      -
//          DEviceTypes(3390)      -
//          VOLumes(SDJVOL))
/*
```

DEFINE NONVSAM

Sert uniquement à cataloguer un fichier non VSAM.

Syntaxe :

```
DEFine NonVSAM -  
  (NAME(nom)  
  DEviceTypes(type-unité ...)  
  VOLumes(numéro-volume ...)  
  OWNER(propriétaire)  
  TO(date-d'expiration)  
  FOR(nombre-de-jours))
```

Obligatoires: nom, type d'unité, volume

Catalogue : utiliser les alias

Exemple :

```
DEFINE NONVSAM -  
  (NAME(SDJ.NVSAM.FICH1) -  
  DEVT(3390) -  
  VOLUMES(VOL120 VOL130))
```

ALIAS

Alias : autre nom

Il existe 2 types d'alias :

1. alias de catalogue
2. alias de fichier non VSAM

Alias catalogue = 1 ou plusieurs qualifieurs

Alias fichier non VSAM = autre nom qualifié

DEFINE ALIAS

```
DEFine ALIAS  
  (NAME(nom-alias)  
   RELate(nom-référencé))
```

Exemple :

```
DEFINE ALIAS          -  
  (NAME(NVSAM.ALIAS1) -  
   RELATE(SDJ.NVSAM.FICH1))
```

GENERATION DATA GROUP

Ensemble de fichiers non VSAM ayant un nom générique

Exemple :

SDJ.PAIE.G0001V01

SDJ.PAIE.G0002V01

On doit préciser le nombre de versions à conserver et le mode de rotation.

DEFINE GDG

Syntaxe :

```
DEFine GenerationDataGroup  
    (NAME(nom-gdg)  
    LIMit(nombre-gds)  
    EMPty NoEMPty  
    OWNER(propriétaire)  
    SCRatch NoSCRatch  
    TO(date-d'expiration)  
    FOR(nombre-de-jours))
```

Paramètres obligatoires :

- NAME : nom du GDG
- LIMIT : nombre de GDS à cataloguer

Autres paramètres (incidence quand nombre limite atteint)

- EMPTY , NOEMPTY (défaut)
- SCRATCH , NOSCRATCH (défaut)

DEFINE GDG

Exemple :

```
DEFINE GDG      -  
      (NAME(SDJ.PAIE) -  
      LIMIT(10))
```

Quand on crée la 11ème version :
la version 1 est décataloguée,
elle n'est pas détruite;
les versions 2 à 10 sont conservées.

Création d'une nouvelle version :

```
//GDS  DD      DSN=SDJ.PAIE(+1),  
//          DISP=(,CATLG),UNIT=...,  
//          SPACE=...
```

GDS (Generation Data Set) : un des fichiers du GDG

CLUSTER VSAM

Cluster VSAM = entité VSAM

cluster de base = le cluster autour duquel sont construits les autres clusters (AIX)

Un cluster = 1 ou 2 composants

Un composant = un fichier physique = 1 entrée dans la VTOC

Nom du cluster = DSN = nom donné par le programmeur

Nom de composant : nom de fichier contenant les données

Dans la VTOC : nom de composant uniquement

Lien entre nom du cluster et noms des composants dans le catalogue

CLUSTER VSAM

Exemple de nom de cluster :

cluster de base :

SDJ.FICHER.CLIENT

Composant data :

*SDJ.FICHER.CLIENT.**DATA***

Composant index :

*SDJ.FICHER.CLIENT.**INDEX***

cluster AIX :

SDJ.FICHER.CLIENT.AIX

Composant data de l'AIX :

SDJ.FICHER.CLIENT.AIX.DATA

Composant index de l'AIX :

SDJ.FICHER.CLIENT.AIX.INDEX

Path associé :

*SDJ.FICHER.CLIENT.AIX.**PATH***

DÉFINITION D'UN CLUSTER

Syntaxe générale :

DEFine CLuster (paramètres cluster) DATA (paramètres data) IndeX (paramètres index)
--

Les paramètres des composants data et index peuvent être définis au niveau cluster.

Il y a propagation des valeurs des paramètres définis au niveau cluster sur le composant data et/ou sur le composant index.

La partie index n'existe que pour les KSDS.

VALORISATION DU PARAMÈTRE CISZ

Exemples :

1/ DEF CL (NAME(SDJ.FICH.VSAM) ... CISZ(2048))

2/ DEF CL (NAME(A.B.D) ... CISZ(2048)) -
 DATA (CISZ(4096) ...)

3/ DEF CL (NAME(A.B.E) ... CISZ(2048)) -
 DATA (CISZ(8192) ...) -
 IX (CISZ(1024) ...)

4/ DEF CL (NAME(A.B.F) ...) -
 DATA (CISZ(22528) ...)

Taille des CIs des composants des clusters

Fichier	data	index
SDJ.FICH.VSAM	2K	2K
A.B.D	4K	2K
A.B.E	8K	1K
A.B.F	22K	0.5K

PARAMÈTRES OBLIGATOIRES

Pour définir un cluster il faut préciser :

1. le nom
2. les allocations d'espace désirées
3. le(s) volume(s)

ou

1. le nom
2. un modèle

Par défaut on définit un KSDS, KEYS(64,0).

Exemples :

```
DEFine CLuster (NAME(SDJ.CLIENT.EX1) -  
                CYLinders(1,1) -  
                VOLumes(APIWK1))
```

```
DEFine CLuster (NAME(SDJ.CLIENT.EX2) -  
                MODEL(SDJ.CLIENT.EX1))
```

LES NOMS

Le nom du cluster est obligatoire.

C'est le DSN qu'il faudra fournir pour accéder au fichier.

Le nom du/des composant(s) est conseillé.

```
DEFine  
  CLuster (NAME(SDJ.KSDS.CLIENT) ...) -  
  DATA (NAME(SDJ.KSDS.CLIENT.data) ) -  
  IndeX (NAME(SDJ.KSDS.CLIENT.index))
```

Si les noms des composants data et index sont omis, des noms génériques sont utilisés :

hlq.Tbbbbbbb.VIDyyddd.Taaaaaaa

ORGANISATION DU CLUSTER

Indication du type d'organisation désirée :

Défaut : **IndeXeD**

<u>KSDS</u>	DEFine CLuster (NAME(SDJ.KSDS.fich) IndeXeD)
<u>ESDS</u>	DEFine CLuster (NAME(SDJ.ESDS.fich) NonIndeXeD)
<u>RRDS</u>	DEFine CLuster (NAME(SDJ.RRDS.fich) NUMbereD)
<u>LDS</u>	DEFine CLuster (NAME(SDJ.LINEAR.fich) LINear)

DEFINE CLUSTER

[illegible]

DEFINE CLUSTER (INDEX)

DEFINE CLUSTER	ESSDS	KSDS	RDS	z/OS	VSE
INDEX [ATTEMPTS(n 2)] [AUTHORIZATION(point d'entrée)] [CODE(code)] [CONTROLINTERVALSIZE (taille)] [CONTROLPW(mot de passe)] [CYLINDERS (primaire[secondaire]) BLOCKS(primaire[secondaire]) RECORDS(primaire[secondaire]) TRACKS (primaire[secondaire]) [FILE(nom)] [IMBED/ <u>NOIMBED</u>] [MASTERPW(mot de passe)] [MODEL (nom[/mot de passe] (nom cat[/mot de passe]])] [(NAME(entry name)] [<u>ORDERED/UNORDERED</u>] [OWNER(nom propriétaire)] [READPW(mot de passe)] [REUSE <u>NOREUSE</u>] [SHAREOPTIONS(crossregion 1 (cross system 3] [<u>UNIQUE/SUBALLOCATION</u> NOALLOCATION] [UPDATEPW(mot de passe)] [USECLASS(primaire[secondaire])] [VOLUMES(volser[volser...]) <u>DEFAULTVOLUMES</u>] [WRITECHECK <u>NOWRITECHECK</u>]) [CATALOG (nom catalogue [/mot de passe]]	N		N		N
				N	N
				N	N
				N	N

DÉFINITION D'UN KSDS

Exemple :

```
//SYSIN DD *
DEFine CLuster
    (NAME(SDJ.KSDS.CLIENT) -
    INDEXED -
    CYLinders(1,1) -
    VOLumes(APIWK1) -
    FreeSPaCe(10 20) -
    /*----- % FSPC par CI, % CI FSPC/CA */ -
    KEYS(5 0) /*long.pos.*/ -
    RECoRdSiZe(100 100)) -
    /* taille moyenne maximum */ -

    DATA (NAME(SDJ.KSDS.CLIENT.DATA) -
           ControlIntervalSiZe(4096)) -

    INDEX (NAME(SDJ.KSDS.CLIENT.INDEX))
```

KSDS : LES PARAMÈTRES

KEYS précise la **longueur** et la **position** de la **clé** primaire.

Est obligatoire pour un KSDS (défaut (64 0))

RECORDSIZE : taille des enregistrements logiques

CI (dans l'exemple) : 4Ko, pour le composant data fixé par l'AMS, pour composant index

Si non précisé :

Taille CI = $n * \text{taille max enregistrement} + \text{taille infos contrôle}$ $\leq \text{taille du CI}$

Autre paramètre possible : SPaNneD

DÉFINITION D'UN ESDS

Exemple :

```
DEFine CLuster (NAME(SDJ.ESDS.CLIENT) -  
  NONINDEXED -  
  TRacKs(10,5) -  
  VOLumes(VOL123) -  
  ControlIntervaSiZe(4096) -  
  RECORDSiZe(100 200)) -  
  /* moyenne maximum */ -  
  
DATA (NAME(SDJ.ESDS.CLIENT.DATA))
```

DÉFINITION D'UN RRDS

Exemple :

```
DEFine CLuster (NAME(SDJ.RRDS.CLIENT)-  
    NUMBERED -  
    RECords(10000) -  
    VOLumes(VOL123) -  
    ControlIntervaSiZe(4096) -  
    RECordSiZe(200 200)) -  
    /* moyenne=maximum */  
  
DATA (NAME(SDJ.RRDS.CLIENT.DATA))
```

Recordsize obligatoire pour un RRDS

Longueur moyenne d'un enregistrement=

= Longueur max d'enregistrement

DÉFINITION D'UN LDS

Exemple :

```
DEFine CLuster (NAME(SDJ.LDS.CLIENT) -  
    LINEAR -  
    CYLinders(1,1) -  
    VOLumes(VOL123) -  
  
    DATA (NAME(SDJ.LDS.CLIENT.DATA))
```

FICHER TEMPORAIRE

Fichier dont les données ne sont valides que pendant la durée d'un JOB

Non VSAM

```
//      JOB ...  
//STEP1 EXEC ...  
//DD1   DD  DSN=&&TEMP,DISP=(NEW,PASS) ...  
//.....  
//STEPN EXEC ...  
//DD1   DD  DSN=&&TEMP,DISP=(OLD,DELETE)
```

VSAM :

Pas de fichier temporaire

Fichier défini avec le paramètre ReUSe

Lors d'une nouvelle utilisation en écriture High
Used RBA remis à 0

LES LISTES

LISTVTOC (IEHLIST)

liste les composants d'un volume (sans en montrer les liens).

LISTCAT (IDCAMS) fournit les entrées au catalogue

On précise :

- l'entrée ou les entrées à lister.
- le type d'entrée à explorer : cluster, data, index
- les informations désirées.

LISTCAT

L I S T C A T	z/ O S	V S E
(entryname [/mot de passe] [ALTERNATEINDEX] [CLUSTER] [DATA] [INDEX] [PATH] [CATALOG(nom catalog[pswd])] [ENTRIES(entry name[/pswd] ...) volser[...] n LEVEL(level) n [CREATION(jours)] [EXPIRATION(jours)] n [<u>NAME</u> VOLUME ALLOCATION ALL HISTORY] n [NOTUSABLE] [OUTFILE(dname)] n		

LISTCAT

NAME/HISTORY/VOLUME/ALLOCATION/ALL

Paramètres	Renseignements obtenus
NAME	Nom de l'entrée, type de l'entrée, associations
HISTory	Ceux fournis par NAME + propriétaire, dates de création et d'expiration
VOLume	Ceux fournis par HIST + type d'unité, numéros des volumes
ALLOcation	Ceux fournis par VOL + allocation primaire et secondaire, HURBA, HARBA
ALL	Toutes les informations contenues au catalogue

Par défaut : NAME

REPRO : CHARGEMENT

Chargement d'un fichier ?

- par programme (ASM, Cobol, PL1 ...)
- par l'AMS

avec un fichier ou des données in stream

Le fichier à charger doit être défini avant.

Que préciser ?

- Enregistrements à copier ou à sauter

REPRO

R E P R O	z/ O S	V S E
{INFILE(dname[/pswd]) INDATASET(entryname[/pswd])}		N
[ENVIRONMENT (DUMMY) (BLOCKSIZE(taille) [HINDEXEDEVICE(devtype)] [NOLABEL <u>STD</u> LABEL] [NOREWIND REWIND UNLOAD] [PRIMEDATADEVICE(devtype)] [RECORDFORMAT(format)] [RECORDSIZE(taille)]	N N N N N N N	N
[FROMKEY(clé)/ FROMADDRESS(adresse) FROMNUMBER(nbre) SKIP(nbre)		
{OUTFILE(dname[/pswd]) [ENVIRONMENT(*)] OUTDATASET(entryname[/pswd])}	N	N
[TOKEY(clé)/ TOADDRESS(adresse) TONUMBER(nbre) COUNT(nbre)]		
[REPLACE/ <u>NOREPLACE</u>]		
[REUSE/ <u>NOREUSE</u>]		

CHARGER UN KSDS

Exemple:

- le fichier source est séquentiel,
- ses enregistrements sont de longueur fixe,
- la clé primaire du KSDS a été définie par **KEYS(5 0).**

```
// JOB
//TRI EXEC PGM=SORT
//SORTIN DD DSN=SDJ.séquent.fich,
// DISP=OLD
//SORTOUT DD DISP=(,PASS),UNIT=3390,
// SPACE=(CYL,(1,1))
//SYSIN DD *
        SORT      KEYS=(1,5,CH,A)
/*
//SYSOUT DD SYSOUT=*
/*
//CHARGT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SOURCE DD DSN=*.TRI.SORTOUT,
// DISP=(OLD,PASS)
//SYSIN DD *
        REPRO InFILE(SOURCE)
              OutDataSet(SDJ.KSDS.FICH)
/*
```

CHARGER UN CLUSTER D'UN AUTRE TYPE

Dans un ESDS les enregistrements sont écrits dans l'ordre d'arrivée.

Dans un RRDS le 1er enregistrement lu est mis dans le 1er slot,

le 2ème enregistrement lu est mis dans le 2ème slot

...

PARAMÈTRE "REUSE"

On peut recharger un fichier.

```
DEFine CLuster (NAME(SDJ.fich2) -  
        ReUSe/NoReUSe ...)
```

```
REPRO IDS(SDJ.fich1) -  
      ODS(SDJ.fich2) -  
      ReUSe/NoReUSe
```

- DEF RUS, REPRO RUS : avant REPRO fich2 vidé de son contenu (HURBA à 0).
- DEF NRUS, REPRO RUS :
 - fich2 vide : chargement fich1 dans fich2
 - fich2 non vide : ERREUR
- REPRO NRUS : fusion fich1 et fich2 dans fichier2

FUSION

La commande REPRO sert à fusionner des fichiers.

Utiliser le paramètre NoReUSe (défaut)

Dans un ESDS il s'agit toujours d'un ajout en fin de fichier.

KSDS et RRDS : REPlace NoREPlace autorise ou non le remplacement d'enregistrements, si clés identiques en fichiers source et cible.

ECLATEMENT DE FICHIERS

Chargement partiel :

- nombre d'enregistrements à sauter : SKIP(s)
- nombre d'enregistrements à charger : COUNT(c)
 - Eclatement KSDS : FromKEY(a) TOKEY(b)
 - Eclatement ESDS (ou KSDS) :
FromADDRess(lowRBA)
ToADDRess(highRBA)
 - Eclatement RRDS : FromNUMber(RRN1)
ToNUMber(RRN2)
 -

Exemples :

- | | |
|-------------------------------------|--------|
| 1. FNUM(200) COUNT(50) | (RRDS) |
| 2. FKEY(clé1) COUNT(30) TOKEY(clé2) | (KSDS) |
| 3. FADDR(125000) | (ESDS) |

DELETE : SUPPRESSION

DElete	nom
	ALIAS AlternateIndeX
	CLuster PATH

Type facultatif

Contrôle effectué, si demandé

Détruire un cluster détruit tous ses associés :
composant data, index, cluster AIX, path.

RÉORGANISATION DE FICHER

Un KSDS ayant subi de nombreux CI et CA splits doit être réorganisé,
sinon **dégradation des temps d'accès**.

Lors d'un chargement en séquence :

- prise en compte du freespace
- enregistrements classés,
d'où pas de split

Exemple :

DEFine CLuster (NAME(SDJ.ksds.temp) - MODEL (SDJ.ksds.fich))	1
REPRO InDataSet(SDJ.ksds.fich) - OutDataSet(SDJ.ksds.temp)	2
DElete (SDJ.ksds.fich) Cluster	3
DEFine CLuster (NAME(SDJ.ksds.fich) - MODEL (SDJ.ksds.temp)) -	4
REPRO InDataSet(SDJ.ksds.temp) - OutDataSet(SDJ.ksds.fich)	5
DElete (SDJ.ksds.temp) cluster	6

COMMANDES MODALES

On peut exécuter plusieurs commandes avec un appel à l'AMS.

Chaque commande fournit un code retour 0, 4, 8, 12 ou 16.

2 variables sont utilisables :

LASTCC, code rendu par la dernière commande exécutée

MAXCC, plus haut code rendu par les commandes précédemment exécutées

EXÉCUTION CONDITIONNELLE

```
IF test THEN [DO]
    Commande(s)
    [END]
ELSE [DO]
    commande(s)
    [END]
```

DO ... END obligatoires si plusieurs commandes

Si THEN ou ELSE ne sont pas suivis, sur la même ligne, de DO, d'une commande ou d'un caractère de continuation aucun traitement ne sera effectué.

Changer de ligne après DO

END seul sur sa ligne

Changer de ligne pour ELSE

EXÉCUTION CONDITIONNELLE

Test : LASTCC MAXCC opérateur nombre

opérateurs : = ^= = <=

EQ NE GT LT GE LE

nombre : 0 à 16

Jusqu'à 10 niveaux de tests imbriqués

Exemples :

```
IF LASTCC = 4 THEN commande  
  ELSE commande
```

```
IF MAXCC NE 0 THEN -  
  IF LASTCC GE 4 THEN -  
    DO  
      commande1  
      commande2  
    END  
  ELSE  
    Commande x  
  ELSE DO  
    commande3  
    commande4  
  END
```

MODIFICATION CODE RETOUR

A la fin de l'AMS le code retour fourni est le MAXCC.

La commande

SET LASTCC ou MAXCC = nombre

modifie LASTCC ou MAXCC.

EXEMPLE :

```
DEFINE CLuster (NAME(SDJ.fich) ...)
IF LASTCC LE 4 -
  THEN DO
    REPRO IDS(SDJ.sequent) -
      ODS(SDJ.fich)
    IF LASTCC GT 4 -
      THEN DO
        DELETE(SDJ.fich)
        IF LASTCC LE 8 THEN -
          SET MAXCC = 4
      END
    ELSE DO
      DELETE(SDJ.sequent)
      IF LASTCC LE 4 THEN -
        SET MAXCC = 0
    END
  END
```


CLÉ SECONDAIRE

Clé : champ d'enregistrement permettant accès et classement

Clé primaire sur un KSDS : clé précisant l'ordre dans lequel sont stockés les enregistrements.

Clé secondaire (KSDS ou ESDS) : clé permettant de classer les enregistrements selon un autre critère.

Doubles autorisés sur clés secondaires

Clé primaire non modifiable

Clé secondaire modifiable si accès par autre clé

Clé secondaire à l'intérieur du 1er segment

ORGANISATION PHYSIQUE

Alternate index : fichier qui, pour chaque valeur de la clé secondaire, indique

- clé(s) primaire(s) associée(s), si KSDS
- RBA(s) associée(s), si ESDS

Structure d'un AIX : KSDS

Clé primaire AIX = clé secondaire du cluster de base

Présentation VSAM

SDJ Informatique

Présentation VSAM

SDJ Informatique

DEFINE ALTERNATE INDEX

	z/ O S	V S E
INDEX		
[ATTEMPTS(nombre)]		
[AUTHORIZATION(point d'entrée)]		
[CODE(code)]		
[CONTROLINTERVALSIZE(taille)]		
[CONTROLPW(mot de passe)]		
[CYLINDERS(primaire[secondaire]) BLOCKS(primaire[secondaire]) RECORDS(primaire[secondaire]) TRACKS(primaire[secondaire])	n	
[FILE(nom)]		
[IMBED] <u>[NOIMBED]</u>		
[KEYRANGES((clé mini clé maxi)...)]		
[KEYS(long position)]		
[MASTERPW(mot de passe)]		
[MODEL(nom[/mot de passe] [(nom cat[/mot de passe]])]		
[NAME (entryname)]		
[<u>ORDERED</u> <u>UNORDERED</u>]		
[OWNER(nom propriétaire)]		
[READPW(mot de passe)]		
[<u>REPLICATE</u> <u>NOREPLICATE</u>]		
[<u>REUSE</u> <u>NOREUSE</u>]		
[SHAREOPTIONS(crossregion (cross system)]		n
[<u>UNIQUE</u> <u>SUBALLOCATION</u> NOALLOCATION]	n	
[UPDATEPW(mot de passe)]		
[USECLASS(primaire[secondaire])]	n	
[VOLUMES(volser[volser...]) <u>DEFAULTVOLUMES</u>]	n	
[WRITECHECK <u>NOWRITECHECK</u>])		

DEFINE PATH

DEFINE PATH	M V S	V S E
<p>(NAME (nom) PATHENTRY(nom [/mot de passe] [ATTEMPTS(nombre)] [AUTHORIZATION(point d'entrée)] [CODE(code)] [CONTROLPW(mot de passe)] [FILE(nom)] [MASTERPW(mot de passe)] [MODEL(nom[/mot de passe] [(nom cat[/mot de passe]])] [OWNER(nom propriétaire)] [READPW(mot de passe)] [TO(date) FOR(jours)] [UPDATE/NOUPDATE] [UPDATEPW(mot de passe)]) [CATALOG(nom catalog[/mot de passe])]</p>		

DEFINE BLDINDEX

BLDINDEX	z/ O S	V S E
{INDATASET(nom cluster de base[pswd]) /INFILE(nom cluster de base[/pswd])}		n
{OUTDATASET(nom AIX ou PATH[pswd]) /OUTFILE(nom cluster de base[/pswd])}		n
[CATALOG(nom catalogue[mot de passe])]		
[EXTERNALSORT <u>INTERNALSORT</u>]		
[WORKFILES(IDCUT1 IDCUT2)]		n
[WORKVOLUMES(volser)]	n	

LE PATH

Possibilité d'accéder à un cluster à travers un AIX

Le nom du PATH est le DSN à fournir pour accéder au fichier à travers l'AIX.

NOTA : Ordre des définitions.

1. Définir Cluster
2. Définir AIX
3. Définir PATH
4. BLDINDEX

Le cluster, AIX et PATH doivent se trouver dans le même catalogue.

Le DDname du Path doit être le DDname du cluster de base suffixée par 1

Exemple

```
//FCLIENT DD DSN=SDJ.KSDS.CLUSTER,DISP=SHR  
//FCLIENT1 DD DSN=SDJ.KSDS.CLUSTER.AIX.PATH,DISP=SHR
```


CRÉER UN INDEX ALTERNÉ

Exemple

```
DEfINE AlternateIndeX -  
    (NAME(SDJ.ksds.client.aix) -  
    RELate(SDJ.ksds.client) -  
    TRacKs(50,5) -  
    VOLumes(VOL111) -  
    FreeSPaCe(20 10) /*% par ci et % de ci libres par ca*/-  
    KEYS(10 0) /* lg. pos. */ -  
    NonUNiQueKey -  
    UPGrade ) -  
DATA (NAME(SDJ.ksds.client.aix.d) -  
    ControllIntervaSiZe(4096)) -  
INDEX (NAME(SDJ.ksds.client.aix.i))
```

- RELATE indique le nom du cluster de base.
- KEYS précise la position de la clé secondaire.
- Cet AIX fait partie de l'upgrade set.

Si la clé secondaire est unique, les enregistrements de l'AIX sont de taille fixe .

CRÉATION D'UN INDEX ALTERNÉ

La commande BLDINDEX permet de remplir l'AIX.

```
BldIndex InDataSet(SDJ.ksds.client) -  
         OutdataSet(SDJ.ksds.client.aix)
```

Fichier en entrée : cluster de base

Fichier en sortie : index alterné

Le cluster de base doit comporter au moins une entrée.

L'AIX doit être vide ou avoir été créé avec l'option REUSE.

2 sortes de tri :

- interne (défaut), InternalSORT pour fichier de taille raisonnable
- externe, ExternalSORT pour gros fichier nécessitant des fichiers de travail

CONSTRUIRE UN INDEX ALTERNÉ (SUITE)

Utiliser un tri externe

WORKFILE(dd1 dd2) précise les noms des fichiers de travail.

```
//DD1 DD DSN=SDJ.work1,DISP=OLD  
//DD2 DD DSN=SDJ.work2,DISP=OLD
```

ddnames recommandés : IDCUT1, IDCUT2

DÉFINIR UN CHEMIN D'ACCÈS

```
DEFine PATH          -  
    (NAME(SDJ.xsds.client.aix.path) -  
    PathENTry(SDJ.ksds.client.aix) -  
    UPDate)
```

Pour créer un chemin d'accès il faut indiquer

- le nom du chemin d'accès
- l'index alterné à utiliser
- "update" : si l'on veut que lors d'une ouverture en écriture les index alternés de l'upgrade set soient mis à jour.

On peut définir un path sur le cluster de base.

Dans le JCL : DSN = nom de PATH

==> enregistrements du cluster de base présentés dans l'ordre de la clé secondaire

LISTCAT ET DELETE DE AIX

1. LISTCAT

LISTCAT permet d'obtenir la liste des AIX et celle des paths.

On obtient les associations cluster de base, AIX et paths ainsi que les composants data et index de l'AIX.

2. DELETE

Destruction d'un cluster de base

====>destruction des index alternés associés

Destruction d'un index alterné

====>destruction du chemin d'accès associé

UPGRADE SET

La mise à jour d'un AIX peut se faire

- en direct
- en différé

En direct : une mise à jour du cluster de base entraîne la mise à jour correspondante de l'AIX.

Upgrade set : ensemble cluster et AIX automatiquement mis à jour

En différé : aucune mise à jour automatique dans l'AIX

Il faut reconstruire l'AIX avant chaque utilisation.

UTILITAIRES IDCAMS

PRINT

3 formats d'impression

DUMP *Impression sous les 2 formes*

CHARacter *Impression caractères EBCDIC*

HEX *Impression en hexadécimale*

PRINT

Exemple :

```
PRINT InDataSet(SDJ.fich) -  
CHARacter DUMP HEX
```

Impression partielle :

- SKIP(nombre d'enregistrements à sauter)
- COUNT(nombre d'enregistrements à imprimer)
- KSDS : FromKEY(a) TOKEY(b)
- ESDS : FromADDRess(lowRBA)
ToADDRess(highRBA)
- RRDS : FromNUMBER(n1) ToNUMBER(n2)

Exemples :

1. FNUM(1200) COUNT(100)
2. FKEY(100) TOKEY(200) COUNT(10)
3. FADDR(65536)

KSDS sans AIX

On indique le nom du cluster de base.

Impression partielle possible

- par clé (ordres des clés)
- par RBA (ordre des CIs)

Si impression par RBA :

FADDR sur une frontière d'enregistrement

TADDR arrondi à l'enregistrement

On peut imprimer le composant Index.

Exemples :

1. PRINT IDS(SDJ.ksds.fich) -
FKEY(cle1) TOKEY(cle2) CHAR
2. PRINT IDS(SDJ.ksds.fich.index) HEX

KSDS avec AIX

On indique

- soit le nom du cluster de base
- impression dans l'ordre de la clé primaire;
- soit le nom d'un path
- impression dans l'ordre d'une clé secondaire, impression partielle éventuellement.

Impression de l'AIX : On indique son nom, impression partielle éventuellement par clé ou RBA

Exemples:

```
PRINT IDS(SDJ.ksds.fich.aix.path) -  
FKEY(50) TOKEY(80) CHAR
```

```
PRINT IDS(SDJ.ksds.fich.aix) DUMP
```

ESDS sans AIX ou LDS

On indique le nom du cluster de base.

Impression par RBA :

Pour impression partielle :

FADDR sur une frontière d'enregistrement ou de CI

TADDR arrondi à l'enregistrement ou de CI

Exemples:

```
PRINT IDS(SDJ.esds.fich) -  
  FADDR(400) TADDR(4795)
```

```
PRINT IDS(SDJ.lds.fich) -  
  FADDR(2048) TADDR(4095)
```

ESDS avec AIX

On indique

- soit le nom du cluster de base
- impression dans l'ordre de la clé primaire;
- soit le nom d'un path
- impression dans l'ordre d'une clé secondaire, impression partielle éventuellement.

Impression de l'AIX

: nom de l'index alterné

impression partielle éventuellement par clé ou
RBA

Exemples :

```
PRINT IDS(SDJ.esds.fich.aix.path) -  
FKEY(30) TOKEY(70) CHAR
```

```
PRINT IDS(SDJ.esds.fich.aix) DUMP
```

RRDS

On indique le nom du cluster de base.

Possibilité d'impression partielle par numéro d'enregistrement

Exemple:

```
PRINT IDS(SDJ.rrds.fich) -  
      FNUM(300) TNUM(400)
```

Sauvegarder : EXPORT

E X P O R T	z/ O S	V S E
entryname[/pswd] [INFILE(dname)] {OUTDATASET(entryname) OUTFILE(dname [ENVIRONMENT ([BLOCKSIZE(taille)] [NOLABEL <u>STDLABEL</u>] [NOREWIND REWIND UNLOAD] [PRIMDATADEVICE(devtype)])]) [CIMODE <u>RECORDMODE</u>] [ERASE <u>NOERASE</u>] [INHIBITSOURCE <u>NOINHIBITSOURCE</u>] [INHIBITTARGET <u>NOINHIBITTARGET</u>] [PURGE <u>NOPURGE</u>] [TEMPORARY <u>PERMANENT</u>]}	n	n

Sauvegarder : EXPORT

LDS : sauvegarde en mode CI

Autres organisations : sauvegarde en mode record ou CI

(RECORDMode CIMode)

Exporter séparément les AIX

Protection source possible

Fichier de backup : préciser nom, unité et volume

Blocksize par défaut 2Ko

La copie exportée peut être sur bande.

Sauvegarde temporaire ou permanente

Sauvegarde temporaire : le fichier source est conservé.

Sauvegarde permanente : destruction du fichier source

TEMPorary PERManent (défaut PERM)

(T) : On peut continuer à utiliser le fichier après sauvegarde.

(P) : Sauvegarde AIX avant cluster de base (voir destruction)

(P) : Lien entre cluster de base et AIX perdu

IMPORT : Restaurer

I M P O R T	z/ O S	V S E
{INTDATASET(entryname) INFILE(dname [ENVIRONMENT ([BLOCKSIZE(taille) [NOLABEL <u>STDLABEL</u> [NOREWIND REWIND UNLOAD] [PRIMDATADEVICE(devtype)])])	n	n
{OUTFILE(dname[/pswd]) OUTDATASET(entryname[/pswd])		n
[CATALOG(catname[/pswd])] [ERASE NOERASE] [INTOEMPTY]		
[OBJECTS(entryname [FILE(dname)] [KEYRANGES((lowkey highkey)...)] [NEWNAME(nom)] [ORDERED UNORDERED] [USECLASS(primaire[secondaire])] [VOLUMES(volser [...]) DEFAULTVOLUMES]]	n	
[OUTPW(pswd)]		
[PURGE NOPURGE]	n	

Fichier en entrée : créé par EXPORT

On peut changer nom,...

Restaurer le cluster de base avant les AIX, à importer séparément.

ALTER : Modification d'attributs

Il faut préciser le nom de l'entrée à modifier:

- cluster de base,
- composant data,
- composant index ou
- cluster AIX.

Voir les tables des modifications autorisées pour chaque type d'entrée

ALTER : Modification d'attributs

A L T E R	z/ O S	V S E
ENTRYNAME [/mot de passe] [ADDVOLUMES(volser[volser ...])] [ATTEMPTS(number)] [AUTHORIZATION(entry point[string])] [BUFFERSPACE(taille)] [CATALOG(nom catalog[/pswd])] [CODE(code)] [CONTROLPW(mot de passe)] [ERASE NOERASE] [FILE(nom)] [FREESPACE(%CI[%CA])] [KEYS(long position)] [MASTERPW(mot de passe)] [NEWNAME(nom)] [NULLIFY [AUTHORIZATION(MODULES STRING))] [CODE] [CONTROLPW] [MASTERPW] [OWNER] [READPW] [RETENTION] [UPDATEPW] [OWNER(nom propriétaire)] [READPW(mot de passe)] [RECORDSIZE(taille moyenne maxi)] [REMOVEVOLUMES(volser[volser ...])] [SHAREOPTIONS(crossregion (cross system))] [TO(date) FOR(jours)] [UNINHIBIT INHIBIT] [UNIQUEKEY NOUNIQUEKEY] [UPDATE NOUPDATE] [UPDATEPW(mot de passe)] [UPGRADE NOUPGRADE] [WRITECHECK NOWRITECHECK]		n

Commande ALTER

ALTER nom du composant
paramètres à modifier

Quelques paramètres modifiables :

- nom d'un composant
- freespace
- update
- upgrade
- UNiQueKey NonUNiQueKey

Certaines modifications sont possibles sur des fichiers vides.

Existe également la possibilité de conversion ESDS (CI 4Ko.) en LDS

EXAMINE

EXAMINE	NAME(SDJ.ksds.fich) <u>IndexTEST</u> <u>NODataTEST</u> ErrorLIMIT(12)
---------	--

Pour vérifier la structure interne des composants
index ou data d'un KSDS ou d'un AIX

VERIFY

V E R I F Y	M V S	V S E
{FILE(dname[/pswd]) dataset(entryname[/pswd])}		n

Pour vérifier les informations de fin de fichier dans le catalogue

Mise à jour automatique de ces informations en cas d'anomalie

A utiliser, par exemple, en cas d'écriture suivi d'unabend (fichier non fermé)