

HERCULES: Création de DASD

Table of Contents

- [HERCULES: Création de DASD](#)
 - [Utilisation d'images DASD prédéfinies](#)
 - [Création, formatage et chargement de volumes DASD](#)
 - [Création d'un volume DASD vide](#)
 - [Formater le volume DASD vide](#)
 - [Chargement du nouveau volume DASD](#)
 - [Construction d'un volume DASD à partir de fichiers PDS non chargés](#)
 - [Fichier de contrôle](#)
 - [Exemples](#)
 - [Autres utilitaires DASD](#)
 - [DASDLS - Liste des jeux de données sur le volume](#)
 - [DASDCAT - Afficher les membres du PDS](#)
 - [DASDPDSU - Télécharger les membres PDS](#)

Cette page décrit différentes manières de créer et de charger des volumes DASD à utiliser avec Hercules.

Utilisation d'images DASD prédéfinies

IBM distribue les systèmes OS/390 et z/OS préconfigurés sur deux packages de CD-ROM différents:

- Le CD de développement d'applications OS/390 et z/OS (ADCD) disponible uniquement pour les membres d'IBM PartnerWorld for Developers
- OS/390 et z/OS DemoPkg disponible uniquement pour les employés IBM et les partenaires commerciaux IBM qualifiés.

Ces deux packages contiennent des fichiers d'image DASD prédéfinis qui doivent simplement être décompressés sur votre disque dur. Les images décompressées peuvent être lues directement par Hercules.

Création, formatage et chargement de volumes DASD

Création d'un volume DASD vide

Le programme `dasdinit` doit d'abord être exécuté à partir de l'invite du shell Unix pour créer un fichier contenant un volume DASD vide.

Le format de la commande `dasdinit` est le suivant:

```
dasdinit [-options] nom_fichier devtype [-model] [volser] [taille]
```

où:

- **-v** affiche les informations de version et l'aide
- **-z** construit un fichier image dasd compressé à l'aide de zlib
- **-bz2** construit un fichier image dasd compressé avec bzip2
- **-0** construire un fichier image dasd compressé sans compression
- **-lfs** construit un gros fichier dasd (non compressé) (si supporté)
- **-a** construit un fichier image dasd incluant des cylindres alternatifs (option ignorée si la taille est spécifiée manuellement)
- **-r** construit le fichier image 'raw' dasd (pas de piste VOL1 ou IPL)
- **-linux** Les images de piste null ressemblent aux images linux dasdfmt'ed (Type d'appareil 3390 uniquement)
- **nom_fichier** nom du fichier image dasd à créer
- **devtype** CKD: 2305, 2311, 2314, 3330, 3340, 3350, 3375, 3380, 3390, 9345 FBA: 0671, 3310, 3370, 9313, 9332, 9335, 9336
- **model** modele (implique size) (opt)
- **Numéro de série du volume** volser (1-6 caractères) (spécifié uniquement si l'option '-r' n'est pas utilisée)
- **taille** nombre de cylindres CKD ou secteurs FBA de 512 octets (obligatoire si le modèle n'est pas spécifié, sinon facultatif)

La taille [*] peut être spécifiée, sinon la taille par défaut est celle du premier modèle répertorié.



Volumes supérieurs à 2 Go

Pour les volumes CKD dépassant 2 Go, tels que 3390-3 et pour lesquels le paramètre `-lfs` n'est pas spécifié, le programme DASDINIT créera plusieurs fichiers en ajoutant les caractères `1`, `_2`, `_3`, etc. au nom de fichier indiqué sur la ligne de commande. Ces caractères sont insérés avant le premier point (`.`) Après la dernière barre oblique (`/`). S'il n'y a pas de point, les caractères sont ajoutés à la fin du nom. Chaque fichier contient un nombre entier de cylindres. Le support Hercules CKD reconnaît les fichiers comme appartenant à un seul volume logique. Indiquez uniquement le nom complet du premier fichier du fichier de configuration Hercules (par exemple, " nomfichier _1").

Le programme DASDINIT ne peut pas créer de volumes FBA dépassant 2 Go sauf si le paramètre `-lfs` est spécifié et si la taille de fichier importante est prise en charge sur votre plate-forme.

Par exemple, pour créer un volume CKD 3390 modèle 3 (triple densité) de 3339 cylindres avec le numéro de série du volume WORK03:

```
dasdinit triple.a88 3390-3 work03
```

Étant donné que ce volume dépasse 2 Go, DASDINIT créera deux fichiers avec `triple1.a88` contenant les cylindres 0-2518 et `triple2.a88` contenant les cylindres 2519-3339. Si vous spécifiez

```
dasdinit -lfs triple.a88 3390-3 work03
```

DASDINIT créera alors un fichier triple.a88 contenant tous les cylindres. Votre plate-forme doit prendre en charge les fichiers volumineux pour spécifier l'option -lfs .

Formater le volume DASD vide

Après avoir créé un volume DASD, on peut le formater avec un programme tel que IBCDASDI ou ICKDSF autonome.

Voici un exemple d'instructions de contrôle IBCDASDI requises pour initialiser un volume 3330:

```
WORK01 JOB  'INITIALIZE 3330 WORK VOLUME '  
MSG      TODEV=1052,TOADDR=009  
DADEF    TODEV=3330,TOADDR=151,IPL=NO,VOLID=WORK01,BYPASS=YES  
VLD      NEWVOLID=WORK01,OWNERID=HERCULES  
VTOCD    STRTADR=1,EXTENT=5  
END
```

Pour exécuter IBCDASDI, placer les instructions ci-dessus dans un fichier appelé init3330.txt et démarrer Hercules en mode S/370 avec un fichier de configuration contenant les instructions suivantes:

```
CPUSERIAL 001234  
CPUMODEL 3145  
MAINSIZE 2  
CNSLPORT 1052  
ARCHMODE S/370  
0009      1052  
000A      1442      ibcdasdi.rdr  
000C      1442      init3330.txt  
0151      3330      work01.151
```

Après IPLing à partir du périphérique de lecteur de carte 00A, connecter un client Telnet au port 1052 et appuyer sur Entrée. À l'invite IBCDASDI, entrer:

```
input=1442 00c
```

Chargement du nouveau volume DASD

Ensuite, il faut créer un fichier de vidage de volume complet sur l'ordinateur central et le convertir au format AWSTAPE à l'aide du `tapeconv.jcl` du répertoire source Hercules. Le fichier AWSTAPE peut ensuite être téléchargé au format binaire sur le PC où il peut être défini comme un lecteur de bande virtuelle dans le fichier de configuration Hercules.

Un programme autonome peut désormais être IPLé dans Hercules pour restaurer l'image de volume à partir de la bande virtuelle sur le volume DASD virtuel formaté.

Construction d'un volume DASD à partir de fichiers PDS non chargés

Le programme `dasdload` peut être exécuté à partir de l'invite du shell Unix pour créer un nouveau fichier image DASD et le charger avec les données des fichiers PDS non chargés.

Le format de la commande `dasdload` est le suivant:

```
dasdload [options] ctlfile outfile msglevel
```

où

[options]

- **-z** construire un fichier image dasd compressé avec zlib
- **-bz2** construire un fichier image compressé dasd en utilisant bzip2
- **-0** construire un fichier image dasd compressé sans compression
- **-lfs** construire un grand fichier image dasd (peut dépasser 2G)
- **-a** construire un fichier image dasd incluant des cylindres alternatifs
- **ctlname** est le nom du fichier de contrôle qui spécifie les jeux de données à charger sur le volume nouvellement créé
- **outfile** est le nom du fichier image DASD à créer
- **msglevel** est un nombre compris entre 0 et 5 qui contrôle le niveau de détail des messages émis lors du chargement.

Fichier de contrôle

Le fichier de contrôle requis par le programme `dasdload` est un fichier texte ASCII constitué d'une instruction de volume suivie d'une instruction de jeu de données pour chaque jeu de données à créer.

Le format de la déclaration de volume est:

```
volser devtype [-model] [cyls [ipltext ]]
```

où:

- **volser** est le numéro de série du volume nouvellement créé
 - **devtype** correspond au type de périphérique émulé (2311, 2314, 3330, 3340, 3350, 3375, 3380 ou 3390) du nouveau volume. Les types de périphériques FBA ne sont pas pris en charge par le programme `dasdload`. Le modèle peut être spécifié comme `dasdinit` ci-dessus.
 - **cyls** est la taille du nouveau volume en cylindres. Si `cyls` est codé *, 0 ou s'il est omis, la taille par défaut du type d'appareil et du modèle est utilisée. Cylinders est ignoré pour les périphériques compressés.
 - **ipltext** est un paramètre facultatif spécifiant le nom d'un fichier contenant le texte IPL qui sera écrit sur le volume. Le fichier doit être sous la forme d'un jeu d'objets contenant des enregistrements EBCDIC de 80 octets de longueur fixe dans le même format que celui

attendu par IBCDASDI ou ICKDSF.

Le format d'une instruction de jeu de données est le suivant:

```
dsname method units pri sec dir dsorg recfm lrecl blksize keylen
```

où:

- **dsname** est le nom du jeu de données
 - **method** est la méthode de chargement de l'ensemble de données qui peut être l'une des suivantes:
 - **XMIT filename** le jeu de données est chargé à partir d'un PDS non chargé créé par la commande TSO XMIT
 - **SEQ filename** le jeu de données séquentiel est chargé à partir d'un fichier binaire . La traduction ascii / ebcdic n'est pas prise en charge pour le moment. De plus, le nom de serveur doit être PS ou DA et recfm doit être F ou FB .
 - **EMPTY** le jeu de données est initialisé avec un enregistrement de fin de fichier (si DSORG est PS) ou un répertoire PDS vide (si DSORG est PO)
 - **DIP** l'ensemble de données est initialisé avec un enregistrement d'en-tête LOGREC
 - **CVOL** le jeu de données est initialisé en tant que système d'exploitation SYSCTLG contenant le nombre minimal d'entrées nécessaire pour IPL sur un système OS / 360
 - **VTOC** spécifie la taille et l'emplacement de la VTOC. Un nom de jeu de données doit être codé sur cette instruction, bien qu'il ne soit pas utilisé. Si aucune instruction VTOC n'est présente, la VTOC sera placée après le dernier jeu de données sur le volume et la taille de la VTOC correspondra au nombre minimum de pistes nécessaires.
 - **units** sont les unités d'allocation d'espace: TRK ou CYL .
 - **pri** est la quantité primaire d'allocation d'espace
 - **sec** est la quantité secondaire d'allocation d'espace
 - **dir** est le nombre de blocs de répertoire
 - **dsorg** est l'organisation du jeu de données: PS , PO , DA ou IS ,
 - **recfm** est le format d'enregistrement: F , FB , FBS , V , VB , VBS ou U
 - **lrecl** est la longueur de l'enregistrement logique
 - **blksize** est la taille du bloc
 - **keylen** est la longueur de la clé

Tous les paramètres, à l'exception de dsname et de la méthode, sont facultatifs. Des valeurs par défaut de zéro sont fournies pour les paramètres DCB. Pour les ensembles de données chargés avec la méthode XMIT, les paramètres DCB sont extraits du PDS non chargé et l'allocation d'espace minimale requise pour charger l'ensemble de données est utilisée sauf si une quantité supérieure est spécifiée. Si l'allocation d'espace est omise, la valeur par défaut est TRK 1 0 0. Si CYL est spécifié sans quantité primaire, l'allocation d'espace par défaut est égale à 1 cylindre ou au nombre minimal de cylindres requis pour charger le jeu de données, selon la valeur la plus grande.

Exemples

[1] Pour créer un volume 2314 dans un fichier appelé sysres.230 à l'aide du fichier de contrôle sysres.plf avec le niveau de message 2:

```
dasdload sysres.plf sysres.230 2
```

Un exemple de fichier de contrôle est présenté ci-dessous:

```
#
# Fichier de mise en page pour le volume de résidence du système MFT
#
sysres 2314 * ieaip100.rdr
sys1.parmlib xmit /cdrom/os360/reslibs/parmlib.xmi
sys1.imagelib xmit /cdrom/os360/reslibs/imagelib.xmi
sysctlg cvol trk 1 0 0 ps f 256 256 8
sysvtoc vtoc trk 5
sys1.logrec dip trk 1 0 0
sys1.nucleus xmit /cdrom/os360/reslibs/nucleus.xmi cyl
sys1.svclib xmit /cdrom/os360/reslibs/svclib.xmi cyl
sys1.sysjobqe empty cyl 2 0 0 da f 176 176 0
sys1.dump empty cyl 10 0 0 ps u 0 3625 0
```

[2] Pour créer un volume 3390-3 compressé dans un fichier appelé linux.500 contenant un système linux amorçable pour l'installation de linux / 390 à l'aide du fichier de contrôle linux.prm :

```
dasdload -z linux.prm linux.500
```

Un exemple de fichier de contrôle est présenté ci-dessous:

```
#
# Construire un disque Linux amorçable
# les noms de jeu de données (sys1.linux. ...) sont codés en dur dans
# linuxipl.obj et ne peut pas être modifié sans le reconstruire
#
linux 3390-3 * linuxipl.obj
sys1.linux.parmfile SEQ images / redhat.prm trk 1 0 0 ps fb 1024 1024
sys1.linux.tapeipl.ikr SEQ images / kernel.img trk 200 0 0 ps fb 1024 1024
sys1.linux.initrd SEQ images / initrd.img trk 200 0 0 ps fb 1024 1024
```

Autres utilitaires DASD

Ces programmes peuvent être utilisés pour extraire des données d'images CKD DASD à l'aide de commandes exécutées à l'invite du shell Unix.

DASDLS - Liste des jeux de données sur le volume

DASDLS, écrit par Malcolm Beattie, est une commande permettant de répertorier les noms des jeux de données contenus dans les images de disque.

Le format de la commande est:

```
dasdls ckdfilename [sf=shadow-file-name] ...
```

où

- **ckdfilename** est le nom d'un fichier Unix contenant un volume CKD
- **nom-fichier-image (facultatif)** est le nom du fichier ombre associé.

DASDCAT - Afficher les membres du PDS

DASDCAT est une commande permettant de lire des jeux de données à partir d'images de disque.

Le format de la commande est:

```
dasdcat -i ckdfilename [sf=shadow-file-name] dsname1 dsname2 ... -i ckdfilename2 dsname10 ...
```

où

- **ckdfilename** est le nom d'un fichier Unix contenant un volume CKD,
- **shadow-nom-fichier (facultatif)** est le nom du fichier ombre associé
- **dsname** peut être un nom de fichier brut (non partitionné) (qui n'est actuellement pas géré) ou de la forme pdsname / memname où memname peut être:
 - Nom du membre PDS (automatiquement mis en majuscule), éventuellement suivi de ":" et indicateurs "a" ou "c".
 - "c" signifie © ard des images et transforme un membre PDS avec une taille de bloc multiple de 80 en plusieurs lignes séparées de nouvelle ligne de 72 caractères avec EBCDIC converti en ASCII et avec des numéros de séquence coupés.
 - "a" signifie (a) scier le membre (mais ne coupez pas les numéros de séquence ou ne faites pas l'image de carte).
 - ? entre quotes pour éviter que le shell ne le supprime) pour lister les noms de tous les membres de PDS au lieu d'afficher leur contenu.
 - (encore une fois, entre quotes ou retourné pour éviter que ce soit un glob) pour afficher tous les membres du PDS au lieu d'un seul nommé. Cela peut éventuellement être suivi de deux points, puis, comme ci-dessus. Chaque membre est précédé d'une ligne "> Member: memname" et, si le "c" des indicateurs de carte-images est utilisé, chaque ligne du contenu des membres est précédée de "|" afin de garantir sa distinction.

DASDPDSU - Décharger les membres PDS

DASDPDSU est une commande qui décharge les membres PDS d'une image disque et copie chaque

membre dans un fichier nom_mod. Mac dans le répertoire de travail en cours.

Le format de la commande est:

```
dasdpdsu ckdfilename [sf=shadow-file-name] pdsname [ascii]
```

où

- **ckdfilename** est le nom d'un fichier Unix contenant un volume CKD,
- **shadow-nom-fichier (facultatif)** est le nom du fichier ombre associé
- **pdsname** est le nom d'un PDS sur ce volume. Si le mot clé ascii facultatif est spécifié, les membres seront déchargés sous forme de fichiers texte de longueur variable ASCII. Sinon, les membres sont déchargés sous forme de fichiers binaires EBCDIC de longueur fixe.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=ibm:hercules-390-dasd>

Last update: **2025/02/19 10:59**

