SDisk2

Carte d'émulation de disquette 5,25" pour Apple II — Guide d'utilisation

Ce guide correspond au firmware **v6.7 et supérieur** (lecture *et* écriture).

PRÉPARATION DE LA CARTE SD

Exigences de formatage

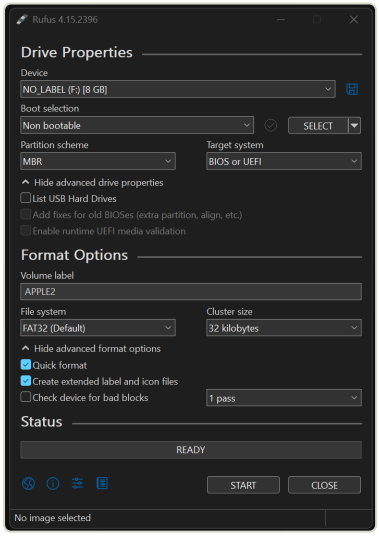
PARAMÈTRE	VALEUR REQUISE
Système de fichiers	**FAT16** ou **FAT32**
Taille de cluster	**4 Ko minimum** (32 Ko recommandé pour la fiabilité)
Table de partition	MBR (une seule partition primaire)
Cartes supportées	SD standard, SDHC

> **Attention :** exFAT et NTFS ne sont **pas** supportés.

Windows (Windows 10 / 11)

La méthode la plus simple est **Rufus**. Utilisez les réglages ci-dessous :

- **Schéma de partition :** MBR
- **Système cible :** BIOS ou UEFI
- **Système de fichiers :** FAT32
- **Taille d'unité d'allocation :** 32 kilo-octets
- **Formatage rapide :** coché



Configuration de formatage Rufus pour SDisk2

Autre option, l'outil officiel **SD Card Formatter** de la SD Association : → https://www.sdcard.org/downloads/formatter/

En ligne de commande avec une taille de cluster précise (invite de commandes en administrateur) :

```
:: FAT32 avec clusters de 32 Ko (remplacer D: par la lettre de votre carte SD)
format D: /FS:FAT32 /A:32K /Q
```

```
:: FAT16 avec clusters de 32 Ko (uniquement pour les cartes ≤ 2 Go)
format D: /FS:FAT /A:32K /Q
```

macOS

```
# Identifier le disque (ex. disk2)
diskutil list

# Partitionner et formater en FAT32 avec clusters de 32 Ko (64 secteurs × 512 o)
diskutil partitionDisk /dev/disk2 MBR FAT32 SDISKII 0b
diskutil unmountDisk /dev/disk2
newfs_msdos -F 32 -c 64 /dev/rdisk2s1
```

Remplacer `32` par `16` pour FAT16, et ajuster `64` (blocs par cluster) si besoin.

Linux

```
# Identifier le périphérique (ex. /dev/sdb, PAS /dev/sdb1)
lsblk

# Créer une table MBR + une partition FAT32
parted /dev/sdb mklabel msdos
parted /dev/sdb mkpart primary fat32 1MiB 100%

# Formater avec des clusters de 32 Ko (64 secteurs × 512 o)
mkfs.fat -F 32 -s 64 /dev/sdb1
```

Remplacer `-F 32` par `-F 16` pour FAT16.

Après le formatage

Copiez vos fichiers `.nic` sur la carte SD. Vous pouvez créer des sous-dossiers (profondeur illimitée). Le fichier `SDISKII.CFG` n'est pas créé automatiquement : il est généré à la demande depuis le menu de configuration (option **Save config**) — voir Fichier de configuration.

FICHIERS IMAGES DISQUE

Seul le format **NIC** (`.nic`) est supporté. Les images DSK/PO doivent être converties au préalable. Les convertisseurs Python sont fournis à la **racine de la carte SD** :

SCRIPT	ENTRÉE	SORTIE
`dsk2nic.py`	`.dsk` / `.do` (ordre secteurs DOS 3.3)	`.nic`
`po2nic.py`	`.po` (ordre secteurs ProDOS)	`.nic`
`nic2dsk.py`	`.nic`	`.dsk` (ou `.po` avec `--prodos`)

```
python3 dsk2nic.py JEU.dsk       # → JEU.nic
python3 nic2dsk.py JEU.nic       # retour au .dsk, pour relire l'image sur PC
```

Aucune dépendance : Python 3 suffit. Des exécutables autonomes (utilisables sans Python installé) se construisent avec `build.bat` (Windows) ou `build.sh` (Linux / macOS) — voir le README fourni à côté des convertisseurs.

> AppleCommander permet de préparer/éditer les images source `.dsk` / `.po` elles-mêmes ; les convertisseurs ci-dessus gèrent l'étape `.nic` spécifique au SDISK II.

Taille d'un fichier NIC : exactement 286 720 octets (280 Kio) = 35 pistes × 16 secteurs × 512 octets.

> **Noms de fichiers :** les **noms longs sont pris en charge** (VFAT). Vous pouvez utiliser des minuscules, des espaces et des noms bien au-delà de l'ancienne limite 8.3 — jusqu'à ~38 caractères sont affichés sur l'OLED, les noms plus longs étant tronqués par `…`. L'extension `.nic` / `.NIC` est masquée à l'affichage (l'icône disque indique déjà le type de fichier) : `Lode Runner.nic` apparaît donc simplement `Lode Runner`.

Organisation recommandée de la carte SD :

```
Racine de la carte SD/
├── SDISKII.CFG          ← optionnel, créé via le menu de configuration
├── dos33/
│   ├── jeux/
│   │   ├── LODERUNNER.NIC
│   │   └── CHOPLIFTER.NIC
│   └── MASTER.NIC
└── prodos/
    └── UTILITAIRES.NIC
```

BOUTONS ET COMMANDES

Les trois boutons sont disposés de gauche à droite sur la carte :

```
[ DOWN ]   [ ENTER ]   [ UP ]
```

BOUTON	UTILISATION NORMALE	DANS LES MENUS
DOWN (gauche)	Ouvrir le menu de réglages	Naviguer vers le bas / diminuer
ENTER (milieu)	Ouvrir le navigateur de fichiers	Confirmer la sélection
UP (droite)	Permuter vers le disque précédent	Naviguer vers le haut / augmenter

La **barre inférieure** de l'écran OLED affiche en permanence les trois icônes correspondant aux actions disponibles.

SÉQUENCE DE DÉMARRAGE

1. Mise sous tension → logo SDisk2 affiché brièvement
2. Détection de la carte SD
   - Pas de carte : message `No SD inserted`
   - Carte présente : initialisation du système de fichiers
3. Deux cas de figure :
   - Si `SDISKII.CFG` existe (créé au préalable via **Save config**, voir Menu de configuration) : la dernière image NIC montée est rechargée automatiquement et l'écran principal s'affiche directement.
   - Sinon (premier usage, ou fichier de configuration supprimé) : SDisk2 démarre directement sur l'**explorateur de fichiers** de la carte SD, prêt à sélectionner une image — inutile d'appuyer sur ENTER pour y accéder.
4. Une fois une image montée, l'écran principal affiche :
   - Barre supérieure : libellé `NIC` + type de système de fichiers (`FAT16` / `FAT32`)
   - Icône disque + nom du fichier monté, sur deux lignes si le nom est long
   - Icône **cadenas** à droite du nom si l'image est protégée en écriture (voir Écriture et protection en écriture)
   - Numéro de piste courant (mis à jour en temps réel lors des lectures de l'Apple II)

- Barre inférieure : icônes des boutons

SÉLECTIONNER UNE IMAGE DISQUE

1. Vérifiez que l'Apple II **ne lit pas** (LED éteinte, moteur du drive arrêté)
2. Si l'explorateur de fichiers n'est pas déjà affiché (un disque est monté et l'écran principal est visible), appuyez sur **ENTER** pour l'ouvrir. Au premier démarrage, ou tant qu'aucune image n'a été montée, l'explorateur est déjà actif.
3. Dans l'explorateur de fichiers :
   - Les dossiers apparaissent sous la forme `[NOMDOSSIER]`
   - Les fichiers NIC sont affichés avec leur nom
   - Utilisez **DOWN** / **UP** pour défiler
   - Appuyez sur **ENTER** sur un dossier pour y entrer
   - Appuyez sur **ENTER** sur un fichier NIC pour le monter
   - Appuyez sur **DOWN** sur `[..]` pour remonter d'un niveau
4. L'image sélectionnée est montée immédiatement et l'Apple II peut démarrer dessus

> **Conseil :** la liste est triée par ordre alphabétique. Jusqu'à 200 fichiers sont affichés par répertoire.

PERMUTER ENTRE DEUX DISQUES

Si `SDISKII.CFG` est présent, SDisk2 mémorise l'image NIC **précédente**.

Appuyez sur **UP** (bouton droit) pour permuter instantanément entre le disque courant et le précédent. Utile pour :

- Les jeux multi-disques (insérer « Disque 2 face B », etc.)
- Copier des fichiers d'une image à l'autre

ÉCRITURE ET PROTECTION EN ÉCRITURE

Le firmware v6.x prend en charge l'**écriture** depuis l'Apple II (`SAVE`, `INIT`, copie de fichiers, etc.). Les secteurs écrits par l'Apple II sont réinjectés directement dans le fichier `.nic` de la carte SD : il n'y a aucune étape de sauvegarde manuelle, et le fichier est modifié en place.

> **Filet de sécurité :** chaque secteur capturé est vérifié (validation GCR) avant d'être écrit sur la carte. Si la capture est douteuse, l'écriture est **annulée** et le message `Write checksum error !` s'affiche à l'écran. L'image `.nic` n'est donc jamais corrompue par une capture incorrecte.

Protéger une image en écriture

Une image est automatiquement protégée dès que le fichier `.nic` porte l'attribut **lecture seule** sur la carte SD :

- **Windows :** clic droit sur le fichier → Propriétés → cocher « Lecture seule »
- **macOS / Linux :** `chmod -w MONDISK.NIC`

Au montage, une **icône cadenas** apparaît à côté du nom du fichier et l'Apple II voit la disquette comme protégée : DOS refuse alors l'écriture, exactement comme avec une vraie disquette dont l'encoche est masquée.

Bascule manuelle

L'entrée **Enable W-Protect** / **Disable W-Protect** du menu de configuration force l'état inverse de l'état courant ; le libellé indique toujours l'action qui sera effectuée, et elle prend effet immédiatement.

> **Attention :** cette bascule est **temporaire**. L'état de protection est relu depuis l'attribut du fichier à chaque montage d'image — y compris à la sortie du menu, qui remonte l'image courante. Pour une protection permanente, utilisez l'attribut lecture seule décrit ci-dessus.

MENU DE CONFIGURATION

Appuyez sur **DOWN** (bouton gauche) pour ouvrir le menu de configuration. Options disponibles :

OPTION	DESCRIPTION
Set SD delay	Compensation de la vitesse de la carte SD (1 = rapide, 10 = lent). Défaut : 5. Augmentez en cas d'erreurs de lecture, surtout avec des images ProDOS. La valeur boucle : au-delà de 10 elle repart à 1.
Set contrast	Luminosité de l'écran OLED (0–255). Défaut : 175. L'effet est visible en direct pendant le réglage.
Save config	Écrit le délai SD et le contraste dans `SDISKII.CFG`, en créant le fichier s'il n'existe pas. Affiche `Done!` si l'écriture a réussi, `SD full / no space` s'il n'a pas été possible d'allouer le fichier.
Delete config	Marque `SDISKII.CFG` comme supprimé. Les réglages reviennent aux valeurs par défaut au prochain démarrage.
Enable W-Protect / **Disable W-Protect**	Force ou lève la protection en écriture de l'image montée. Le libellé indique l'action qui sera effectuée. Effet immédiat mais temporaire — voir Écriture et protection en écriture.
Exit	Retourne à l'affichage principal du disque (l'image courante est remontée).

Naviguez avec **DOWN** / **UP**, confirmez avec **ENTER**. Le menu ne se referme que par **Exit**.

Les valeurs de *Set SD delay* et *Set contrast* prennent effet immédiatement, et sont écrites dans `SDISKII.CFG` à la confirmation **si ce fichier existe déjà**. Au premier usage — tant qu'aucun `SDISKII.CFG` n'a été créé — les réglages ne survivent pas à l'extinction : utilisez **Save config** pour créer le fichier et y enregistrer les deux valeurs.

FICHIER DE CONFIGURATION (SDISKII.CFG)

Le fichier `SDISKII.CFG` est stocké dans le **répertoire racine** de la carte SD. Il n'est **pas** créé automatiquement au démarrage : il faut passer par **Save config** dans le menu de configuration pour le générer.

Paramètres sauvegardés :

- Dernière image NIC montée (répertoire + identifiant de l'entrée FAT)
- Image NIC précédente (pour la fonction de permutation)
- Délai de vitesse de la carte SD
- Valeur du contraste OLED

Tant que le fichier existe, les deux premières entrées sont mises à jour **automatiquement à chaque montage** d'une image — c'est ce qui permet le rechargement au démarrage et la permutation entre deux disques.

La protection en écriture n'y est **pas** enregistrée : elle est relue à chaque montage depuis l'attribut lecture seule du fichier `.nic`.

Si le fichier est supprimé, les réglages reviennent aux valeurs par défaut (délai SD 5, contraste 175) et le démarrage repart sur l'explorateur de fichiers.

MESSAGES D'ERREUR

MESSAGE	SIGNIFICATION	SOLUTION
`No SD inserted`	Aucune carte SD détectée	Insérer une carte
`SD can't idle`	Initialisation SD échouée (étape 1)	Essayer une autre carte ou diminuer le délai
`SD can't init`	Initialisation SD échouée (étape 2)	Idem
`Boot sector`	Impossible de lire le secteur de boot / MBR	Reformater la carte

MESSAGE	SIGNIFICATION	SOLUTION
`Not FAT16/32`	Système de fichiers non supporté	Formater en FAT16 ou FAT32
`Cluster < 32kB`	Taille de cluster trop petite (< 4 Ko)	Reformater avec une taille de cluster plus grande
`Missing MBR`	Aucune table de partition valide trouvée	Reformater avec une table MBR
`SD Block ERROR`	Erreur de lecture/écriture	Vérifier les contacts, essayer une autre carte
`FAT overflow`	Le fichier NIC est fragmenté en trop de clusters	Défragmenter ou reformater + recopier les fichiers proprement
`Write checksum error !`	Une écriture venue de l'Apple II a été rejetée par le contrôle GCR ; l'image `.nic` n'a **pas** été modifiée	Vérifier le câblage et les contacts de la carte ; le message reste affiché jusqu'au montage suivant
`SD full / no space`	*Save config* n'a pas pu créer `SDISKII.CFG` (aucun cluster libre)	Libérer de l'espace sur la carte SD
`Nothing found here!`	Le répertoire ouvert ne contient ni image `.nic` ni sous-dossier	Revenir au niveau supérieur avec `[..]`

Après la plupart des erreurs, l'appareil attend l'éjection de la carte SD avant de reprendre.

AVERTISSEMENTS

- **Ne jamais brancher/débrancher à chaud** -- ne branchez et ne débranchez jamais le SDisk2 lorsque l'Apple II est sous tension. Éteignez toujours l'ordinateur avant d'insérer ou de retirer le SDisk2, afin d'éviter tout risque de dommage matériel ou de corruption des données.
- **Carte contrôleur Disk II requise** (non incluse) -- le SDisk2 vient se connecter sur une carte contrôleur Disk II existante, il ne la remplace pas.
- Produit destiné à être intégré dans un Apple II -- conçu pour un usage hobby et rétro-informatique.
- **Fonctions d'écriture** (SAVE, INIT, etc.) -- nécessitent une carte SD correctement formatée en FAT16 ou FAT32, table de partition MBR, avec des clusters de 32 Ko (voir Préparation de la carte SD).
- Les performances peuvent varier selon la qualité de la carte SD utilisée -- c'est pourquoi une carte SD testée est fournie avec le SDisk2.

Des questions ? **contact@sed-p.net** -- **www.sed-p.net**

SDisk2

5.25" Floppy Disk Emulator for Apple II — User Guide

This guide covers firmware **v6.7 and later** (read *and* write).

SD CARD PREPARATION

Format requirements

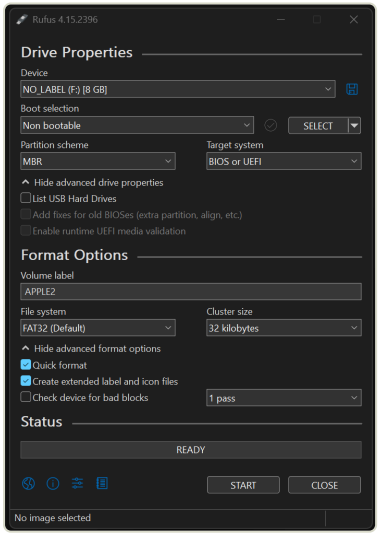
PARAMETER	REQUIREMENT
Filesystem	**FAT16** or **FAT32**
Cluster size	**4 KB minimum** (32 KB recommended for reliability)
Partition scheme	MBR (single primary partition)
Supported types	Standard SD, SDHC

> **Note:** exFAT and NTFS are **not** supported.

Windows (Windows 10 / 11)

The easiest method is **Rufus**. Use the settings shown below:

- **Partition scheme:** MBR
- **Target system:** BIOS or UEFI
- **File system:** FAT32
- **Cluster size:** 32 kilobytes
- **Quick format:** checked



Rufus format configuration for SDisk2

Alternatively, the official **SD Card Formatter** from the SD Association: → https://www.sdcard.org/downloads/formatter/

For command-line formatting with a specific cluster size (open an elevated `cmd`):

```
:: FAT32 with 32 kB clusters (replace D: with your SD card letter)
format D: /FS:FAT32 /A:32K /Q
```

```
:: FAT16 with 32 kB clusters (only works for cards ≤ 2 GB)
format D: /FS:FAT /A:32K /Q
```

macOS

```
# Identify the disk (e.g. disk2)
diskutil list

# Partition and format as FAT32 with 32 kB clusters (64 sectors × 512 B)
diskutil partitionDisk /dev/disk2 MBR FAT32 SDISKII 0b
diskutil unmountDisk /dev/disk2
newfs_msdos -F 32 -c 64 /dev/rdisk2s1
```

Replace `32` with `16` for FAT16, and adjust `64` (blocks per cluster) as needed.

Linux

```
# Identify the device (e.g. /dev/sdb, NOT /dev/sdb1)
lsblk

# Create MBR partition table + one FAT32 partition
parted /dev/sdb mklabel msdos
parted /dev/sdb mkpart primary fat32 1MiB 100%

# Format with 32 kB clusters (64 sectors × 512 B)
mkfs.fat -F 32 -s 64 /dev/sdb1
```

Replace `-F 32` with `-F 16` for FAT16.

After formatting

Copy your `.nic` image files to the SD card. You can create subfolders (any depth). The `SDISKII.CFG` file is not created automatically: it is generated on demand from the Setup Menu (**Save config** option) — see Configuration File.

DISK IMAGE FILES

Only **NIC format** (`.nic`) is supported. DSK/PO images must be converted first. The Python converters are supplied at the **root of the SD card**:

SCRIPT	INPUT	OUTPUT
`dsk2nic.py`	`.dsk` / `.do` (DOS 3.3 sector order)	`.nic`
`po2nic.py`	`.po` (ProDOS sector order)	`.nic`
`nic2dsk.py`	`.nic`	`.dsk` (or `.po` with `--prodos`)

```
python3 dsk2nic.py GAME.dsk      # → GAME.nic
python3 nic2dsk.py GAME.nic      # back to .dsk, to read the image on a PC
```

No dependencies — Python 3 is enough. Standalone executables (usable without Python installed) can be built with `build.bat` (Windows) or `build.sh` (Linux / macOS) — see the README supplied next to the converters.

> AppleCommander can prepare/edit the source `.dsk` / `.po` images themselves; the converters above handle the SDISK II-specific `.nic` step.

NIC file size: exactly 286,720 bytes (280 KiB) = 35 tracks × 16 sectors × 512 bytes.

> **File names: long file names are supported** (VFAT). You can use lowercase, spaces and names well beyond the old 8.3 limit — up to ~38 characters are shown on the OLED, longer names are truncated with `…`. The `.nic` / `.NIC` extension is hidden on the display (the disk icon already indicates the file type), so `Lode Runner.nic` simply appears as `Lode Runner`.

Recommended file placement:

```
SD card root/
├── SDISKII.CFG          ← optional, created via the Setup menu
├── dos33/
│   ├── games/
│   │   ├── LODERUNNER.NIC
│   │   └── CHOPLIFTER.NIC
│   └── MASTER.NIC
└── prodos/
    └── UTILITIES.NIC
```

BUTTONS AND CONTROLS

The three buttons are arranged left to right on the PCB:

```
[ DOWN ]   [ ENTER ]   [ UP ]
```

BUTTON	NORMAL OPERATION	IN MENUS
DOWN (left)	Open Setup menu	Navigate down / decrease
ENTER (middle)	Open file browser	Confirm selection
UP (right)	Swap to previous disk	Navigate up / increase

The **bottom bar** of the OLED always shows the three icons corresponding to the current button actions.

STARTUP SEQUENCE

1. Power on → SDisk2 logo displayed briefly
2. SD card detection
   - If no SD card: `No SD inserted` message
   - If SD card present: filesystem initialised
3. Two cases:
   - If `SDISKII.CFG` exists (created beforehand via **Save config**, see Setup Menu): the last mounted NIC image is reloaded automatically and the main screen is shown directly.
   - Otherwise (first use, or the configuration file was deleted): SDisk2 starts directly in the **file browser**, ready to select an image — no need to press ENTER to reach it.
4. Once an image is mounted, the main screen shows:
   - Top bar: `NIC` label + filesystem type (`FAT16` / `FAT32`)
   - Disk icon + filename of the mounted image, wrapped over two rows for long names
   - A **lock icon** to the right of the name if the image is write-protected (see Writing and Write Protection)
   - Current track number (updated in real time when the Apple II reads)
   - Bottom bar: button icons

SELECTING A DISK IMAGE

1. Make sure the Apple II is **not reading** (LED off, drive motor stopped)
2. If the file browser is not already showing (a disk is mounted and the main screen is visible), press **ENTER** to open it. On first power-on, or as long as no image has been mounted, the browser is already active.
3. In the file browser:
   - Folders are shown as `[FOLDERNAME]`
   - NIC files are shown with their names
   - Use **DOWN** / **UP** to scroll
   - Press **ENTER** on a folder to enter it
   - Press **ENTER** on a NIC file to mount it
   - Press **DOWN** on `[..]` to go up one directory level
4. The selected image is mounted immediately and the Apple II can boot from it

> **Tip:** The list is sorted alphabetically. Up to 200 files are shown per directory.

SWAPPING BETWEEN TWO DISKS

If `SDISKII.CFG` is present, SDisk2 remembers the **previous** NIC image.

Press **UP** (right button) to instantly swap between the current and the previous disk. This is useful for:

- Multi-disk games (insert "Disk 2 side B" etc.)
- Copying files between two images

WRITING AND WRITE PROTECTION

Firmware v6.x supports **writing** from the Apple II (`SAVE`, `INIT`, file copies, and so on). Sectors written by the Apple II are pushed straight back into the `.nic` file on the SD card: there is no manual save step, and the file is modified in place.

> **Safety net:** every captured sector is validated (GCR check) before it is written to the card. If the capture looks wrong the write is **cancelled** and the message `Write checksum error !` appears on screen. A bad capture can never corrupt the `.nic` image.

Write-protecting an image

An image is automatically write-protected as soon as its `.nic` file carries the **read-only** attribute on the SD card:

- **Windows:** right-click the file → Properties → tick "Read-only"
- **macOS / Linux:** `chmod -w MYDISK.NIC`

When mounted, a **lock icon** appears next to the filename and the Apple II sees the disk as protected: DOS then refuses to write, exactly as with a real floppy whose write-enable notch is covered.

Manual toggle

The **Enable W-Protect** / **Disable W-Protect** entry in the Setup Menu forces the opposite of the current state; the label always names the action that will be performed, and it takes effect immediately.

> **Note:** this toggle is **temporary**. The protection state is re-read from the file attribute on every mount — including when you leave the menu, which remounts the current image. For permanent protection, use the read-only attribute above.

SETUP MENU

Press **DOWN** (left button) to open the Setup menu. Available options:

OPTION	DESCRIPTION
Set SD delay	Timing compensation for SD card speed (1 = fast, 10 = slow). Default: 5. Try higher values if you get read errors, especially with ProDOS images. The value wraps around: past 10 it goes back to 1.
Set contrast	Brightness of the OLED display (0–255). Default: 175. The effect is applied live while you adjust it.
Save config	Writes the current SD delay and contrast to `SDISKII.CFG`, creating the file if it does not exist. Shows `Done!` on success, or `SD full / no space` if the file could not be allocated.
Delete config	Marks `SDISKII.CFG` as deleted. Settings revert to defaults on next power-on.
Enable W-Protect / **Disable W-Protect**	Forces or lifts write protection on the mounted image. The label names the action that will be performed. Immediate but temporary — see Writing and Write Protection.
Exit	Returns to the main disk display (the current image is remounted).

Navigate with **DOWN** / **UP**, confirm with **ENTER**. The menu closes only via **Exit**.

Values set in *Set SD delay* and *Set contrast* take effect immediately, and are written to `SDISKII.CFG` when confirmed **if that file already exists**. On first use — as long as no `SDISKII.CFG` has been created — the settings do not survive a power cycle: use **Save config** to create the file and store both values in it.

CONFIGURATION FILE (SDISKII.CFG)

The file `SDISKII.CFG` is stored in the **root directory** of the SD card. It is **not** created automatically at startup: use **Save config** in the Setup Menu to generate it.

Saved settings:

- Last mounted NIC image (directory + FAT entry ID)
- Previous NIC image (for swap feature)
- SD card speed delay
- OLED contrast value

As long as the file exists, the first two entries are updated **automatically on every mount** — this is what makes reload-at-startup and disk swapping work.

Write protection is **not** stored there: it is re-read on every mount from the read-only attribute of the `.nic` file.

If the file is deleted, settings revert to defaults (SD delay 5, contrast 175) and the unit starts up in the file browser again.

ERROR MESSAGES

MESSAGE	MEANING	SOLUTION
`No SD inserted`	No SD card detected	Insert a card
`SD can't idle`	SD card initialisation failed (step 1)	Try a different card or lower the speed delay
`SD can't init`	SD card initialisation failed (step 2)	Same as above
`Boot sector`	Cannot read the boot sector / MBR	Reformat the card
`Not FAT16/32`	Unsupported filesystem	Format as FAT16 or FAT32
`Cluster < 32kB`	Cluster size too small (< 4 KB)	Reformat with larger cluster size

MESSAGE	MEANING	SOLUTION
`Missing MBR`	No valid partition table found	Reformat with MBR partition table
`SD Block ERROR`	Read/write error during operation	Check card contacts, try another card
`FAT overflow`	NIC file is fragmented into too many clusters	Defragment or reformat + copy files cleanly
`Write checksum error !`	A write from the Apple II was rejected by the GCR check; the `.nic` image was **not** modified	Check the card contacts and wiring; the message stays on screen until the next mount
`SD full / no space`	*Save config* could not create `SDISKII.CFG` (no free cluster)	Free up space on the SD card
`Nothing found here!`	The open directory contains no `.nic` image and no sub-folder	Go back up one level with `[..]`

After most errors, the device waits for the SD card to be ejected before resuming.

WARNINGS

- **Never hot-plug** -- never connect or disconnect the SDisk2 while the Apple II is powered on. Always power off the computer before inserting or removing the SDisk2, to avoid any risk of hardware damage or data corruption.
- **Disk II controller card required** (not included) -- the SDisk2 connects onto an existing Disk II controller card, it does not replace it.
- Intended to be installed inside an Apple II -- designed for hobbyist and retrocomputing use.
- **Write functions** (SAVE, INIT, etc.) -- require an SD card correctly formatted as FAT16 or FAT32, MBR partition table, with 32 KB clusters (see SD Card Preparation).
- Performance may vary depending on the SD card used -- which is why a tested SD card is included with the SDisk2.

Questions? **contact@sed-p.net** -- **www.sed-p.net**