

Guide de l'utilisateur – EasySave

Version 1.0

GROUPE 2:

- FABO GAETAN
- ATONLE EDSON
- BOGNING JICELIN
- KINDA ALEX
- FOGOU JORDAN

Table des matières

| | |
|--|---|
| 1. Présentation..... | 3 |
| 2. Configuration et installation..... | 3 |
| 3. Mode interactif..... | 5 |
| 3.1 Sélection de la langue..... | 5 |
| 3.2 Menu principal..... | 5 |
| 4. Mode CLI | 6 |
| 5. Chemins et fichiers générés | 6 |
| 6. Différence : Complete vs Differential | 6 |
| 7. Bonnes pratiques | 7 |
| 8. Support & dépannage..... | 7 |

1. Présentation

EasySave est un utilitaire léger de sauvegarde de dossiers Windows, écrit en .NET 8.0.

Il propose :

- Sauvegarde complète (copie récursive de tous les fichiers et sous-dossiers)
- Sauvegarde différentielle (copie uniquement des fichiers nouveaux ou modifiés depuis la dernière complète)
- Interface multilingue (Anglais/Français)
- Mode interactif et exécution en ligne de commande (CLI)
- Fichiers de configuration JSON et journaux d'exécution

2. Configuration et installation

2.1 Prérequis

- Windows 10+
- .NET 8.0 Runtime installé
- Permissions en lecture sur les dossiers source et écriture sur les dossiers cibles

2.2 Déploiement

Copier le répertoire `bin\Debug\net8.0\EasySoft*` sur votre machine cible.

Conservez la structure :

EasySoft\

└─ EasySoft.exe

└─ EasySoft.dll

└─ Works\backupList.json

└─ State\state.json

└─ DailyLogs\

└─ EasySoft/

└── .vs/

└── bin/

```
|   └── Debug/
|       └── net8.0/
|           ├── EasySoft.exe
|           ├── EasySoft.dll
|           ├── Newtonsoft.Json.dll
|           ├── EasySoft.deps.json
|           ├── EasySoft.runtimeconfig.json
|           ├── EasySoft.pdb
|           ├── State/
|               └── state.json
|           ├── Works/
|               └── backupList.json
|           └── DailyLogs/
|               └── DailyLogs_YYYY-MM-DD.json
└── Controller/
    └── Controller.cs
└── Model/
    ├── Backup.cs
    ├── DataLogs.cs
    ├── DataState.cs
    ├── FileCompare.cs
    └── Model.cs
└── State/
    └── state.json
└── View/
```

```

|   └── View.cs
|
|── Works/
|
|   └── backupList.json
|
|── EasySoft.csproj
|
└── Program.cs

```

2.3 Premier lancement

Lancer EasySoft.exe en double-clic ou via un terminal :

```

cd C:\chemin\vers\EasySoft
EasySoft.exe

```

3. Mode interactif

3.1 Sélection de la langue

In which language would you like to continue? / Dans quelle langue souhaitez-vous continuer ?

1. English

2. Français

Choice (1/2) :

3.2 Menu principal

```

-----
|||           Menu           |||
-----
||| 0. Exit / Quitter           |||
||| 1. Open a backup job / Ouvrir une sauvegarde   |||
||| 2. Create a backup job / Créer une sauvegarde  |||
||| 3. List backups / Lister les sauvegardes      |||
-----

```

Enter menu number:

0. Quitter : Ferme l'application.

1. Ouvrir une sauvegarde existante :

- Affiche la liste des SaveName enregistrés dans Works\backupList.json.
- Saisir le nom exact ou 0 pour revenir.
- Exécute la sauvegarde (full ou differential).

2. Créer une nouvelle sauvegarde :

- Sub-menu pour choisir full ou differential.
- Saisie du nom, chemin source, miroir (diff), chemin cible.
- Ajoute le job à Works\backupList.json.

3. Lister et filtrer :

- Affiche et filtre par type (full/differential).

4. Mode CLI

Vous pouvez lancer directement plusieurs jobs sans l'interface :

dotnet run -- 1-3 ou dotnet run -- 1;3;5

- 1-3 exécute les jobs n°1,2,3
- 1;3;5 exécute les jobs n°1,3,5

Placez-vous dans le dossier du .csproj, par exemple :

```
cd E:\...\EasySoft\V1\EasySoft
dotnet run -- 1-3
```

5. Chemins et fichiers générés

Works\backupList.json : Liste des jobs enregistrés.

State\state.json : État en temps réel de la dernière exécution.

DailyLogs\<YYYY-MM-DD>.json : Journal détaillé par fichier copié.

Exemple journal :

```
[
  {
    "Name": "Sauvegarde1",
    "FileSource": "\\serveur\share\f.txt",
    "FileTarget": "D:\Save\f.txt",
    "FileSize": 1024,
    "FileTransferTime": 12.345,
    "Time": "16/05/2025 14:12:07"
  }, ...
]
```

6. Différence : Complete vs Differential

Complete : copie tous les fichiers et sous-dossiers.

Differential : copie uniquement les fichiers nouveaux ou modifiés depuis la dernière full.

7. Bonnes pratiques

- Sources ≠ Cible : éviter écrasement et boucles.
- Ne pas lancer deux sauvegardes vers le même dossier.
- Nettoyer régulièrement les JSON si changement de tests.

8. Support & dépannage

- "Backup not found" : vérifiez le nom ou listez de nouveau.
- "Incorrect Path" : chemin invalide ou guillemets non retirés.
- Indices hors plage en CLI : vérifiez backupList.json.
- "Aucun projet à exécuter" si dans bin\Debug... : retournez au dossier racine contenant le .csproj.