**DOCUMENTATION TECHNIQUE / DSI ProSoft**

**1. Présentation d’EasySave**

EasySave est une application console développée en C# sur la plateforme .NET Core, permettant d'effectuer des sauvegardes de fichiers sur différents types de supports (disques locaux, externes et réseaux).

L’application permet de gérer jusqu'à 5 travaux de sauvegarde qui peuvent être exécutés individuellement ou de manière séquentielle.

Grâce à son interface en ligne de commande, EasySave offre une solution rapide et efficace pour sécuriser vos fichiers.

EasySave s’adresse aux utilisateurs ayant besoin d’une solution simple et efficace pour effectuer des sauvegardes régulières de leurs fichiers.

Ce guide est destiné aux :

* Utilisateurs professionnels souhaitant sauvegarder des fichiers critiques sur des serveurs ou disques externes.
* Particuliers ayant besoin d’une méthode automatisée pour protéger leurs données personnelles.
* Administrateurs système cherchant une solution légère et performante pour la gestion des sauvegardes.

Avant d’installer et d’utiliser EasySave, assurez-vous que votre système répond aux exigences suivantes :

* **Système d’exploitation** : Windows avec **.NET Core** installé.
* **Droits d’accès** : Permissions suffisantes pour lire et écrire sur les répertoires source et cible.
* **Outils complémentaires** : Un éditeur de texte (ex. Notepad, VS Code) pour lire les fichiers logs au format **JSON ou XML.**

**2. Installation et Configuration**

1**.** Utiliser l’exécutable fourni depuis le zip “bin/Release/net8.0/Publisch.zip”  
2. Décompresser les fichiers dans un répertoire de votre choix.  
3. Cliquer sur l’application EasySave.exe pour commencer à l’utiliser

L’application fonctionne immédiatement après son installation !

**3. Utilisation d’EasySave**

Création d’un travail de sauvegarde

Un travail de sauvegarde est une tâche enregistrée dans le logiciel permettant de copier des fichiers d’un emplacement à un autre.

Chaque sauvegarde est définie par les éléments suivants :

* Nom de la sauvegarde : Identifiant unique pour la reconnaître.
* Répertoire source : Dossier contenant les fichiers à sauvegarder.
* Répertoire cible : Destination où seront copiés les fichiers.
* Type de sauvegarde :
* **Complète** : Copie l’ensemble des fichiers à chaque exécution.
* **Différentielle** : Ne copie que les fichiers modifiés depuis la dernière sauvegarde complète.

Exécution d’une sauvegarde

EasySave s’exécute via la **ligne de commande** et prend en charge plusieurs modes d'exécution :

* Pour Exécuter un seul travail :

Entrez l’indice de la sauvegarde à exécuter, par ex., ‘1-3’ pour exécuter automatiquement les sauvegardes 1 à 3 :

1

à Lance uniquement la **sauvegarde n°1**.

* Pour Exécuter plusieurs sauvegardes en séquence :

Entrez l’indice de la sauvegarde à exécuter, par ex., ‘1-3’ pour exécuter automatiquement les sauvegardes 1 à 3 :

1-3

à Exécute **séquentiellement** les sauvegardes **1, 2 et 3**.

* Pour Exécuter plusieurs sauvegardes en parallèle :

Entrez l’indice de la sauvegarde à exécuter, par ex., ‘1-3’ pour exécuter automatiquement les sauvegardes 1 à 3 :

1;3

à Lance simultanément **les sauvegardes 1 et 3**.

**4. Types de Sauvegarde**

* Sauvegarde Complète

🔹 **Description** : Copie **tous les fichiers** du répertoire source vers le répertoire cible.  
🔹 **Avantage** : Facile à restaurer, assure une copie intégrale.  
🔹 **Inconvénient** : Plus long et plus gourmand en espace disque.

* Sauvegarde Différentielle

🔹 **Description** : Copie **uniquement** les fichiers **modifiés** depuis la dernière sauvegarde complète.  
🔹 **Avantage** : Gagne du temps et économise de l’espace disque.  
🔹 **Inconvénient** : Nécessite une sauvegarde complète pour être restaurée correctement.

**5. Gestion des fichiers Logs et États**

Les logs sont gérés en fonction du type de format choisis des logs juste avant l’exécution !

Fichier Log journalier

EasySave enregistre toutes ses actions dans un fichier log JSON/XML mis à jour en temps réel.

Informations contenues :

* Horodatage de chaque action.
* Nom de la sauvegarde concernée.
* Adresse source et destination des fichiers.
* Taille et durée du transfert (en millisecondes).
* Erreurs éventuelles.

🔹 Format du fichier : JSON/XML  
🔹 Exemple de nom de fichier : backup\_log\_2024-02-10.json/xml

Fichier État en Temps Réel

En complément des logs, EasySave génère un fichier d’état en temps réel contenant la progression des sauvegardes.

Données enregistrées :

* Nom de la sauvegarde
* Horodatage de la dernière action
* État (Actif, Non Actif)
* Progression : % de complétion, fichiers restants, taille des fichiers restants
* Adresse du fichier en cours de transfert

🔹 Format du fichier : JSON/XML  
🔹 Nom du fichier : backup\_state.json/xml

**6. Bonnes Pratiques et Recommandations**

🔹 Évitez les espaces dans les chemins de fichiers pour éviter les erreurs.  
🔹 Les fichiers de logs journaliers sont stockés dans : *C:\ProgramData\CESI\EasySave\Logs*

🔹 Les fichiers de logs temps réel sont stockés dans le répertoire du projet : *bin/Relase/net8.0/Publish*

🔹 Le fichier dans lequel sont enregistrées les différentes sauvegardes créées est stocké dans le même répertoire que la solution du projet (fichier .sln). Qui est appelé « RegisteredBackups.txt »  
🔹 Vérifiez l’espace disque disponible sur le répertoire cible.  
🔹 Contrôlez les fichiers logs régulièrement pour anticiper d’éventuelles erreurs.

**7. Dépannage et FAQ**

Problèmes fréquents et solutions

| **Problème** | **Solution** |
| --- | --- |
| **Les fichiers ne sont pas sauvegardés** | Vérifiez que les répertoires source et cible existent et sont accessibles. |
| **Le fichier log est vide ou absent** | Assurez-vous qu’EasySave a bien les droits d’écriture. |