

|  |
| --- |
| Livrable 3 : EasySave V3.0 |
|  |
| 26FÉVRIER  Service : DSI ProSoft  Créé par : Hatice Yildiz, Lisa Jung, Aya Salem, Ayse Deniz |

Une image contenant capture d’écran, Police, Graphique, texte

Description générée automatiquement

Table des matières

[1. CONTEXTE : 3](#_Toc191498826)

[1.1 DSI ProSoft : 4](#_Toc191498827)

[2. INTRODUCTION : 5](#_Toc191498828)

[Version 3.0: 6](#_Toc191498829)

[3. DIAGRAMMES : 6](#_Toc191498830)

[3.1 Diagramme Use Case : 6](#_Toc191498831)

[3.2 Diagramme d’activité : 8](#_Toc191498832)

[3.3 Diagramme de classes : 10](#_Toc191498833)

[3.4 Diagramme de séquence : 13](#_Toc191498834)

[4. Documentation CryptoSoft V3.0: 40](#_Toc191498835)

[5. Documentation de la Dynamic Link Library V3.0: 40](#_Toc191498836)

[6. Documentation Utilisateur V3.0: 40](#_Toc191498837)

[7. Documentation Technique V3.0 : 40](#_Toc191498838)

[8. Release note V3.0 : 40](#_Toc191498839)

[9. Conclusion 41](#_Toc191498840)

1. CONTEXTE :

Le projet EasySave, développé dans le cadre de l’éditeur de logiciels ProSoft, vise à concevoir une solution performante et évolutive pour la gestion et l’automatisation des sauvegardes de fichiers. Ce logiciel, structuré en plusieurs versions successives, repose sur une architecture modulaire favorisant la maintenance, l’évolutivité et la compatibilité avec les futures améliorations. En intégrant des fonctionnalités telles que la sauvegarde complète et différentielle, un suivi détaillé via des logs journaliers et une interface utilisateur optimisée, EasySave répond aux besoins des entreprises en matière de protection des données. Le développement suit les bonnes pratiques du génie logiciel, avec une gestion rigoureuse des versions, l’utilisation de C# et .NET 8.0, ainsi qu’une documentation garantissant une prise en main rapide et efficace par les utilisateurs et le support technique.

## DSI ProSoft :

Nous sommes une équipe de 4 membres, représentant la DSI ProSoft, qui vise à concevoir une solution performante et évolutive pour la gestion et l’automatisation des sauvegardes de fichiers autour d’un nouveau logiciel qui sera implémenté « EasySave ».

Lisa JUNG

Une image contenant personne, Visage humain, sourire, mur

Description générée automatiquement

Ayse DENIZ

Une image contenant personne, Visage humain, sourire, habits

Description générée automatiquement

Hatice YILDIZ

Une image contenant Visage humain, personne, habits, sourire

Description générée automatiquement

Aya SALEM

Une image contenant personne, Visage humain, habits, verres

Le contenu généré par l’IA peut être incorrect.

1. INTRODUCTION :

EasySave 3.0 marque une évolution majeure en introduisant des fonctionnalités avancées de gestion des sauvegardes, optimisant la performance, la flexibilité et l’interactivité du logiciel. Cette version s’appuie sur les fondations posées par EasySave 2.0, qui a intégré une interface graphique (WPF) et supprimé la limite du nombre de sauvegardes tout en renforçant la gestion des logiciels métiers et l’intégration de CryptoSoft pour le cryptage des fichiers.

Avec la version 3.0, EasySave franchit un cap décisif en introduisant l’exécution parallèle des sauvegardes, garantissant ainsi une gestion plus efficace des ressources. Une nouvelle fonctionnalité permet de prioriser certaines extensions de fichiers, empêchant la sauvegarde de fichiers non prioritaires tant que des fichiers critiques sont en attente. La bande passante est également optimisée grâce à la restriction du transfert simultané de fichiers volumineux, assurant ainsi une meilleure répartition des ressources réseau.

En parallèle, EasySave 3.0 propose une console déportée permettant aux utilisateurs de surveiller et d’interagir avec les sauvegardes en temps réel depuis un poste distant via un système de communication basé sur des sockets. Enfin, pour améliorer la fiabilité et la sécurité du processus, CryptoSoft devient une application mono-instance, empêchant ainsi plusieurs exécutions simultanées.

Ces évolutions font d’EasySave 3.0 une solution de sauvegarde robuste, moderne et adaptée aux exigences des professionnels, alliant performance, sécurité et interactivité.

# Version 3.0:

1. DIAGRAMMES :

## Diagramme Use Case :

Un diagramme de cas d'utilisation (Use Case Diagram) est une représentation graphique des interactions entre les utilisateurs (acteurs) et un système. Il illustre les fonctionnalités principales offertes par le système sous forme de cas d'utilisation, permettant ainsi de comprendre qui utilise le système et comment.

Ce type de diagramme est essentiel en phase de conception, car il permet de clarifier les besoins et de définir les interactions entre le logiciel et ses utilisateurs sans entrer dans les détails techniques.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

Explication :

Ce diagramme de cas d'utilisation illustre les principales fonctionnalités du logiciel EasySave et les interactions possibles entre l'utilisateur et le système. Elle est composée de :

Acteur :

Utilisateur : L’acteur représenté par un bonhomme est celui qui interagit avec le logiciel pour effectuer différentes opérations de sauvegarde.

Cas d’utilisation principaux :

* Réaliser une sauvegarde

Fonctionnalité centrale du logiciel, permettant à l’utilisateur de gérer ses sauvegardes.

* Créer un travail de sauvegarde

Avant d’exécuter une sauvegarde, l’utilisateur doit créer un travail de sauvegarde, définissant :

* Un nom
* Un répertoire source
* Un répertoire cible
* Un type de sauvegarde (Complète ou Différentielle)
* Exécuter une ou plusieurs sauvegardes

Une fois le travail de sauvegarde configuré, l’utilisateur peut lancer manuellement une ou plusieurs sauvegardes.

Il peut choisir de lancer toutes les sauvegardes en séquence ou de sélectionner des sauvegardes spécifiques.

* Consulter l’état temps réel des sauvegardes

Permet à l’utilisateur de suivre en direct la progression des sauvegardes.

Les informations incluent :

* Nom de la sauvegarde
* Source
* Destination
* Le nombre total de fichiers
* La taille totale des données
* Le nombre de fichiers restants
* Le progrès de la sauvegarde
* Consulter le fichier log journalier

L’utilisateur peut consulter les logs journaliers générés par le logiciel.

Ces fichiers enregistrent toutes les actions de sauvegarde et leurs détails (date, heure, état de la sauvegarde, erreurs éventuelles, temps, source, destination).

## Diagramme d’activité :

Un diagramme d'activité est une représentation graphique des flux de contrôle et des processus d'un système. Il illustre la succession des actions et des décisions effectuées au sein du logiciel, souvent sous la forme d'un workflow.

Ce type de diagramme est utilisé pour modéliser le comportement dynamique d’un système en mettant en évidence les différentes étapes d'un processus, les choix possibles, ainsi que les conditions de transition entre les actions. Il est particulièrement utile pour visualiser les algorithmes, les flux de travail métier, ou encore les enchaînements logiques d'exécution.

Une image contenant texte, diagramme, dessin, croquis

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme activité :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqtWOtu2zYUfhVCQOEWSJrGuTjxjw1Bk_1Ym2Zr1mGYHQiMRNvcJFLhpU3m5l36c95r-MV2SEoUZUt2MqxAGvvwnO8cnjszjxKekmgYTTL-JZlhodAv52OG4N_5cnGr1Uv36xXa3f0OvccsITlhKsZFkdEEK8rZyFIFynoB8cZhtApYqLczTu9jnFgE80VSC5EE0iFPIJRhNtWkFsLIUbqkfsIC58uFEngaSBGJCnvwtxJEdmoUy4U1OpZYfyZTLFLAACJcWDOCamoXwsX9cpHoVYiS6lD2UjCmPuw2hjOpM-XsyfhUjkoKcfcxpC7ZnzVVwDcqf7dGy_3_4kXF7L5XEgblB8pews-rMfO8VlMZgVC3o6zFOpC85CmdVDkRxMIxhFELUCZc5FiVt_exRI6MJNczTNVygQquzSVnVCou6J3Pjg5YwKMJJZmMzSEFR10wsMX71R2i8hBRlvC8AMtvzTn-TBLLuB7DDnWJeCiUScdAC7lXhEnwhUQpXIgmM9Ak0fIbstxEbIZkPFFESY-IkcI0A9fk-J7mGD6sZBkiDHRWydnIm8DJVsOvOKOpS7sgTiNLJptrac2xGwBDucpDT2Qvb7-Vu_2sJUsb9eC7AEp7zaovbWjpEhbzGpsEBevysPgdFSofwYEJ9noX2QIIeS4S0opJkDv8T7iQFoqytYYXgAccz9NQFmusHoqG4VURgzfMUStoqxtXY51UmrvychP0mkM70Rti7f7aItvhiS1SbSdOZLkQvo-GWT1_v3J_yZnapdDGEs5MiQPp-8cw1TuhjKKvV5p-RZdESijL2obgBvHVu9HZxHQucH5tZMPt6OrdzVOUfeCsVlYfxjlmdxozVSvSzM7PEACismcua_u0ECRRsjHitlyiox1ssKZzVwjGXb0IdHSR1k3BIp_pOOeUyXg_NJQwQaYw3wQh8wupdqHm73QPa2R50X7odfAI8vywvfjAb4VejwbWCZgf1vA48tEYAmLSvF1D8Ti6aXpzDa17a3qOxTZZr-tpF6dUFpzZgV3njtkdQahtNtYmS98qWtGCbAkQ4uW3mFRLXrCr7q3sCWa-e77G5OwCcztthh9GZ0WhH4Agtd2BbrlWkFw9c9YrF6ByNW_fTYCvpYNUEzvOiZnX80_ML0B-_4F0gkZiNogE1EBn7dWbxMaesgJeh6oKCRy4vFjhrGMWZMptxu-0yWZJQ7JmZuGzlzeHXCMMGyxTZquy7QjC3kjP8hYGZbkAUQYazepUaCrBrUkGBpVO22Zn58K73Q-2yC7uSZn3Ju5hDsAyR3Oz20ICOy6QW83bgMl3vKq7PAHYvfL4NFYkL2QMxZQFxRJs08ZplgeZCsmq515Dsnx8aUkamdqzpJ7fZuGLa3IuRiP7OxU2SoIUgkrv-zKKhvsjkTpfAf5IikJY2QrdcVWPJxjV2hRj9QluFKRt-yOs5Upvy9U0rnb0-TmI-oU95fQzKQft8h9oSG5_Xy6kXaAm0AnC7ON_-oKpDWtV02nQteJF0xOGUvnAfHYOhkkIc1MIsGq9KfgyqJxtuDsmYVXbaybW9fxJUeisUCeWh1_ziRpVNFTTXJsqHzjWJA-V2J7pXxOtqmzJdC3uLSY4P_zBTe2UXqweGqYkR-VRmdppGTrzDEkxjNSW92Q32Kanb_iSt4lnnlr_w6t_ldXdt2pZpevsSfzj9dWHcHBn_s1pO6QRhowwXH5kPwf-t8v329GByYOvtNc1WztSsVP7Gn-0E-UE3rU0jYbR3EiPIzUjOawkQ_iYkgmGy42jMXsEVgyz9PqBJdFQCU12IsH1dBYNJziT8E0XsIqTc4qnsApWLAVmv3Oeeyb4Hg3n0X003N0_ORi8Pu0P-kenh4OT48PDw53oAegn-wevD_aPBoPjwZuDk_7R4070l4Xovz7qv-kfHx0NTvv9g-ODPgiQlELuXbq_19k_2z3-C4hIRA8)

Explication :

Notre diagramme d’activité illustre le déroulement des évènements au sein de notre application EasySave.

Le choix de l’action représente le clic sur un des boutons de la part de l’utilisateur.

Dans le cas où l’utilisateur modifie la langue du logiciel, le flux de contrôle revient directement au choix de l’action. Effectivement, le choix de la langue se fait en 1 clic dans une liste déroulante affichée par défaut sur l’Interface Homme Machine.

Dans le cas où l’utilisateur modifie les paramètres, il est invité à remplir plusieurs champs, puis à cliquer sur un bouton de validation. Ensuite, l’utilisateur peut choisir une autre action.

Dans le cas où l’utilisateur choisit de créer une nouvelle sauvegarde, il est invité à remplir plusieurs champs, puis à cliquer sur un bouton de validation. Ensuite, une vérification permet de s’assurer que les saisies de l’utilisateur sont correctes. Enfin, il peut choisir une autre action.

Dans le cas où l’utilisateur choisit d’exécuter une ou plusieurs sauvegardes, il est invité à remplir un champ en donnant les numéros des sauvegardes. Ensuite, l’utilisateur peut cliquer sur un bouton play pour lancer les sauvegardes. Pendant l’exécution des sauvegardes, cet utilisateur peut appuyer sur des boutons pause, resume, et stop pour agir sur les sauvegardes en cours. Enfin, il peut choisir une autre action.

Dans le cas où l’utilisateur choisit de consulter les logs, le fichier d’historique du jour est affiché. Enfin, il peut choisir une autre action.

Dans le cas où l’utilisateur clic sur le bouton quitter, l’application EasySave se ferme.

## Diagramme de classes :

Un diagramme de classe est une représentation graphique des structures statiques d’un système logiciel. Il décrit les classes, leurs attributs, leurs méthodes, ainsi que les relations qu’elles entretiennent entre elles (association, héritage, agrégation, composition, etc.).

Ce type de diagramme est essentiel en programmation orientée objet car il permet de visualiser l’architecture du code avant son implémentation. Il aide à mieux structurer le projet, favorise la réutilisabilité et assure une bonne compréhension du modèle de données du logiciel.

Une image contenant diagramme, texte, Plan, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de classes :



Différents types de relations entre deux classes ont été utilisées dans le diagramme de classes :

*  : une flèche vide avec un trait en pointillés représente une dépendance. C’est-à-dire que la classe d’où part la flèche utilise la classe vers laquelle la flèche est dirigée.
*  : une flèche vide avec un trait plein représente un héritage. Cela signifie que la classe vers laquelle est dirigée la flèche est une fille de la classe d’où la flèche part. Dans ce type de relation la classe fille est un type plus spécifique de sa classe mère et implémente les mêmes attributs et méthodes que celle-ci.
*  : un losange vide avec un trait plein représente une agrégation. Cela veut dire que la classe vers laquelle le losange est dirigé contient un ou plusieurs objets de la classe d’où part le losange. Mais les instances de la classe d’où part le losange peuvent vivre une fois l’autre classe détruite.
*  : un losange plein avec un trait plein représente une composition. Cela veut dire que la classe vers laquelle le losange est dirigé contient un ou plusieurs objets de la classe d’où part le losange. Mais les instances de la classe d’où part le losange ne peuvent pas vivre une fois l’autre classe détruite.

Explication :

Dans le diagramme de classes de l’application EasySave nous trouvons plusieurs patrons de conception.

En premier lieu, nous avons un patron de conception nommé usine. Il permet de centraliser la création d’objet dans le but de découpler l’instanciation de classes spécifiques du reste du code.

Dans notre cas, le patron de conception usine est formé par l’interface BackupStrategy et ses deux classes héritières CompleteBackupStrategy et DifferentialBackupStrategy qui permettent d’exécuter une sauvegarde en fonction de son type (complète ou différentielle). De plus, il y a l’interface BackupStrategyFactory et ses deux classes filles CompleteBackupFactory et DifferentialBackupFactory dont l’objectif est de créer des objets de type CompleteBackupStrategy ou DifferentialBackupStrategy.

L’implémentation du patron de conception usine respecte le principe SOLID ouvert/fermé. En effet, si nous devons ajouter un type de sauvegarde il nous faudra uniquement ajouter une classe héritière de BackupStrategy et une classe fille de BackupStrategyFactory. Il n’y aura alors pas nécessité de modifier tout le code.

En second lieu, nous avons aussi un patron de conception appelé stratégie. Il permet de changer dynamiquement le comportement d’un objet en déléguant une partie de son comportement à des objets appelés stratégies.

Dans notre cas, le patron de conception Stratégie est formé par l’interface ITranslateStrategy ainsi que ses deux classes héritières French et English. Ces dernières permettent de déterminer la langue dans laquelle l’application est traduite (Français ou Anglais). Il y a encore la classe Translation qui implémente une instance de type ITranslateStrategy. Elle permet de traduire du texte dans la langue déterminée par ITranslateStrategy.

L’implémentation du patron de conception stratégie respecte, lui aussi, le principe SOLID ouvert/fermé. Effectivement, si nous devons ajouter une langue pour l’application il nous faudra uniquement ajouter une classe héritière de ITranslateStrategy. Il n’y aura alors pas nécessité de modifier tout le code.

En troisième lieu, nous avons un patron de conception singleton. Il garantit qu’une classe ne peut avoir qu’une seule instance et fournit un point d’accès global à l’instance en question.

Dans notre cas, le patron de conception singleton est mis en place dans la classe Translation. Pour ce faire, le constructeur de la classe est redéfini. Plutôt que de créer une nouvelle instance de la classe Translation son constructeur vérifie tout d’abord s’il existe déjà un objet de cette classe. Si tel est le cas l’objet existant est retourné, sinon une nouvelle instance est créée. De cette manière, il y aura toujours une seule et unique instance de la classe Translation dans le code.

L’implémentation du patron de conception singleton limite le nombre d’instance d’une classe à 1. Cela peut être utile dans certains cas, notamment lorsque l’instanciation d’une classe est couteuse en mémoire, il est alors préférable de garantir l’unicité de son instance.

En dernier lieu, nous avons encore un patron de conception nommé observateur. Il permet à un objet (le diffuseur) de notifier automatiquement plusieurs autres objets (les souscripteurs) lorsqu’un événement se produit.

Dans notre cas, le patron de conception observateur est formé par la classe BackupStateJournal qui a le rôle de diffuseur. Ce dernier est également relié à l’interface IObserver ainsi que sa classe héritière ConsoleView. Cette classe implémente une méthode qui retourne les informations comprises dans les notifications de la classe BackupStateJournal. En outre, le diffuseur implémente plusieurs méthodes qui ont pour objectif, entre autres, d’ajouter un objet à la liste des souscripteurs ou encore de notifier tous les souscripteurs.

L’implémentation du patron de conception observateur permet d’ajouter facilement des souscripteurs sans modifier tout le code. Effectivement, il nous suffit d’appeler la méthode adéquate du diffuseur. Un nouveau souscripteur sera alors ajouté à la liste de diffusion et notifié comme tous les autres souscripteurs. Le patron de conception observateur permet également d’automatiser les réactions dans le code sans devoir le modifier.

Enfin, l’utilisation de ces différents patrons de conception dans le code a encore pour objectif la réutilisabilité des différentes parties du code sans devoir le modifier.

* 1. Diagramme de séquence :

Le diagramme de séquence UML est un outil graphique permettant de représenter l’interaction entre plusieurs composants d’un système au fil du temps. Il est utilisé principalement en modélisation logicielle pour visualiser le flux d’exécution d’un programme.

Dans le projet EasySave, le diagramme de séquence est essentiel pour représenter le fonctionnement du processus de sauvegarde.

Un diagramme de séquence fait partie des diagrammes UML. Il permet de :

* Décrire l’ordre chronologique des interactions entre les composants.
* Visualiser la communication entre objets et acteurs d’un système.
* Aider à comprendre le flux d'exécution d'une fonctionnalité.
* Servir de référence pour le développement et la documentation technique.

Composé essentiellement de :

* Acteurs :

Représentés par un bonhomme   
Ils symbolisent un utilisateur ou un système externe qui interagit avec l’application.   
*Exemple concret :* Un utilisateur qui déclenche une sauvegarde.

* Classes/Objets et participants :

 Représentés par des rectangles en haut du diagramme   
Ils correspondent aux différents composants logiciels qui participent à l’exécution du scénario.

*Exemple concret :* Une chaîne de restaurants :

* L’acteur serait le client qui passe commande.
* Les classes/objets seraient les différents services du restaurant :
* Serveur (BackupController) qui prend la commande et la transmet à la cuisine.
* Chef cuisinier (LogController) qui prépare la commande et note les plats réalisés.
* Responsable de salle (MenuController) qui organise le bon déroulement du service.

* Messages et interactions :

Flèches dirigées entre les objets   
Elles indiquent les appels de méthodes et les retours de valeurs.

Il existe différents types de flèches :

*  : ce type de flèches représente des messages synchrones dont l’expéditeur attend une réponse de la part du destinataire avant de pouvoir continuer son exécution.
*  : ce type de flèches représente des messages asynchrones dont l’expéditeur n’attend pas de réponse de la part du destinataire avant de pouvoir continuer son exécution.
*  : ce type de flèches représente des messages asynchrones qui sont utilisés lorsque le destinataire d’un message synchrone répond à son expéditeur.
* Une image contenant blanc, Police, clé, ligne

  Le contenu généré par l’IA peut être incorrect. : ce type de flèches représente des messages asynchrones qui sont utilisés lorsqu’un nouvel objet d’une classe est créé.
* Une image contenant symbole, ligne, Symétrie, conception

  Le contenu généré par l’IA peut être incorrect. : un message avec une croix au bout de la flèche est utilisé lorsqu’un objet d’une classe est détruit.
* Barres d'activation :

Représentées par des rectangles verticaux sur les objets   
Elles indiquent qu’un objet est actif et en train d’exécuter une tâche.

* Boucles et conditions :

 Représentées par des blocs alt, loop, opt en UML   
Elles permettent d’ajouter des conditions et des répétitions.

Dans notre projet, nous nous sommes appuyés sur le diagramme de cas d'utilisation (Use Case Diagram). En effet, chaque bulle dans ce type de diagramme représente une action, ce qui nous a permis de structurer notre approche.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

Ainsi, nous avons conçu cinq diagrammes de séquence, chacun illustrant un processus spécifique. Voici ces diagrammes accompagnés de leurs explications détaillées :

* 1. Diagramme séquence pour choisir les paramétrages, format des logs, cryptage, taille des fichiers, fichiers prioritaires, logiciels métiers :

Une image contenant texte, diagramme, Parallèle, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme séquence paramétrage :



Explication :

Le diagramme de séquence du paramétrage d’EasySave 3.0 illustre le processus de configuration des paramètres généraux de l’application par l’utilisateur via l’interface graphique (WPF). Ce processus permet de définir plusieurs aspects essentiels du logiciel, tels que le format des logs, les extensions de fichiers à crypter, la gestion des logiciels métiers, les extensions prioritaires et la taille maximale des fichiers pouvant être sauvegardés en parallèle. Voici une explication détaillée des différentes interactions présentes dans le diagramme.

1. Sélection du format des logs

L’utilisateur démarre la configuration en accédant au menu des paramètres via l’interface graphique de MainWindow. Lorsqu’il clique sur le bouton "Settings\_b", l’interface déclenche l’affichage du menu des paramètres.

* MenuController récupère la liste des formats de logs disponibles (JSON/XML) via MenuModel.
* L’utilisateur sélectionne ensuite un format, qui est transmis à MenuController via HandleLogFormat().
* MenuController met à jour GlobalVariables, où la modification du format des logs est prise en compte et appliquée grâce à la méthode GetLogFileName() de la DLL Logger.
* MainWindow affiche un message de confirmation une fois la modification validée.

2. Ajout des extensions de fichiers à crypter

L’utilisateur peut ajouter des extensions de fichiers spécifiques à crypter après la sauvegarde. Lorsqu’il valide son choix :

* L’interface appelle la méthode ExtensionsChange() de ExtensionController.
* Cette mise à jour est enregistrée dans GlobalVariables, qui conserve la liste des extensions chiffrables.
* Cette information sera utilisée ultérieurement lors de l’exécution d’une sauvegarde.

3. Gestion des logiciels métiers

L’utilisateur peut également indiquer une liste de logiciels métiers, qui sont des applications critiques pouvant interférer avec les sauvegardes. L’ajout d’un logiciel métier suit le processus suivant :

* L’interface graphique transmet la liste des logiciels métiers via LoadBusinessSoftware() à SoftwareWatcher.
* SoftwareWatcher enregistre ces logiciels dans GlobalVariables.
* Cette configuration permettra au logiciel d’arrêter temporairement les sauvegardes en cas de détection d’un logiciel métier actif.

4. Gestion des extensions prioritaires

Une nouvelle fonctionnalité de la version 3.0 permet aux utilisateurs de définir des extensions de fichiers prioritaires, qui doivent être sauvegardées en premier avant tous les autres fichiers.

* L’utilisateur ajoute ces extensions via l’interface.
* Elles sont enregistrées dans extensions.txt et synchronisées avec GlobalVariables.

Lorsqu’une sauvegarde est exécutée, les fichiers contenant ces extensions seront copiés en priorité, conformément aux règles de gestion de la bande passante et des fichiers volumineux.

5. Définition de la taille maximale des fichiers à sauvegarder

Pour éviter la surcharge du réseau, EasySave 3.0 introduit un paramètre permettant de limiter le transfert simultané de fichiers volumineux. L’utilisateur définit une taille maximale de fichiers (en Ko) dans les paramètres :

* Cette valeur est enregistrée dans GlobalVariables sous maximumSize.
* Elle sera utilisée par BaseBackupStrategy pour différencier les petits fichiers des gros fichiers, et ainsi appliquer une gestion adaptée (exécution parallèle limitée pour les fichiers volumineux).

6. Validation et application des paramètres

Une fois tous les paramètres définis, l’utilisateur confirme les changements. MainWindow appelle alors MenuController pour appliquer les modifications. Le système valide et stocke ces nouveaux paramètres, garantissant qu’ils seront pris en compte lors des prochaines sauvegardes.

7. Impact sur l’exécution des sauvegardes

L’application des paramètres a des répercussions directes sur la gestion des sauvegardes :

1. Format des logs : Les fichiers logs seront générés en JSON ou XML, en fonction du choix utilisateur.
2. Fichiers à crypter : Seuls les fichiers avec les extensions définies seront chiffrés après la sauvegarde.
3. Logiciels métiers : L’exécution des sauvegardes sera bloquée en cas de détection d’un logiciel critique actif.
4. Fichiers prioritaires : Ces fichiers seront sauvegardés en premier.
5. Taille des fichiers : Les fichiers volumineux ne pourront pas être transférés en parallèle.

Une fois les paramètres appliqués, l’utilisateur peut lancer une sauvegarde via BackupController, qui appliquera les règles définies.

* 1. Diagramme de séquence pour le choix de la langue :

Une image contenant texte, ligne, diagramme, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Lien vers le [diagramme de séquence pour le choix de la langue](https://mermaid.live/edit#pako:eNq1Vt1O2zAUfhXLN1BROmihpRFCQjAQ2tAmOpg0VZpMctpacu3MdqAd4l12t_EcfbEdO0mbNikdF8tF659zPh9_5zsneaKhioAG1MCPBGQI55wNNRv3JcGHhVZpcmtAE2bIreWCG2Yh0el2zLTlIY-ZtOTy9srZXGoWj3jIROp1JS3oAQuh7HDNuPzKZaQey3t33PB7UeH0kclhwoZwpqTVSgjQ622u8WJi_fYdh0cX8dI8IFfFedn7i2bSCOTgtQhyI65kefP0syfK_V0qNRSwwCwbX2jMycjZZyMMcG7esxp_h9Oy23s5xEx5v3xY7ThPM3_AFZ-zdMlnb7dGdjCvATkTHNWhiUk0wYDvVWKVJH3quQLyjoiMtD5dQVzN8mLu0BezgPRG6jHn3mzXyuYnJ2RnKbcBuZm9hEk8e9Hg40IK2Hj2x2rIY3NYeUxLvmQXj98tnu_ZGYN13hux1sXlRBOQc25iwaaL28xHddIDAaGFKF-qLQfnVViK7QbGseB4R0YwmUhrBCblHEOqDCirIJLyur10hezI3KR0GhLEB9xRusWQDn6fWOTAm2MDsNM-xfNJCNqikyGzFzF7GYO0pirFXj83IKOU1nCk-MS5u6uk6nkoFnuuu5Ls-vTTBxIrHD0wwaM04RVoEWZxztMmzc3byQgHsE50hXIOFsWaWxd2U3OsbG8WJdyLxw9Dvw-ShHgF02g0ciW5PuAzUD4lE-N6BGL9KbYikNfU_a-AVYksFtwW7sEK_ZgRwyt5Xynd08HA1cGrMGT2C4NdrZiqGqxVvBnwjDgW8ypMYfDKeeKYsOQMBWSwsC6QntlvxjMJFytyAehZTftwQI6PQw0og5OT1AWEgTlc1nQ3gmV2FWgyWu4LRcfaQpJL1-2B_W6y1r5d7vb_Q91WJVht2IwsTCz-uSRuMaQdPwEWr7_NGt-EkwvTvFHqb8StbF3YevEDRm_GwlcGdsVksgIawdq3YWELDyutuWZI63QMesx4hB9pT86kT-0IxlgGAQ4jGLBEWFcTz2jK8MXcm8qQBlYnUKdaJcMRDQYM1VmnSRwhbPaFN1_FL4ZvSi3NafBEJzTY3W8c7LfanfbhXqfdPjrstg7rdEqDVrfR3Gs3293WfruDz9Fznf70EM1GZ69z1Do4OGrud5vdbuv5Ly3ajOc).

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqlVeFq2zAQfhWhP21ZGtKkSRpRAqVbQ2CFsawdjMDQ7IstUCRPktekIe-yf6PPkRfbyY7jNHbbwUyIJd9330n33UkrGugQKKMWfqagAngveGT4fKoIPjxw2pA7C4ZwS-6ckMJyB6nJzQk3TgQi4cqR0d3YY0aGJ7EIuMy9xsqBmfEAqg63XKivQoX6oWq7F1b8kDVOH7mKUh7BtVbOaCnBvIy5xY3Jl833Ah78ip_NGRnvz6veXwxXVmIOXltBARJaVY1Xn7JE-ddI60hCyVkF3xjUJPb47QgXuINPnMH_aFl1-6AiVCrzK4b1jjuZxS_8kml28OlQpkzW0xPyDgVn5DrWKJUhkuMP8wYkxBjZVNhiQyWFdyxnrJQzxgEcn9ThszifN09BmmyeDAaXR2iDBQlhL2wQaytq4z0rB0auZjMI3Os0ZPObTOkEpEeGBcGUvkZf1gPGSBK5LAw5DZZCsT0u3S5vNyjJ5g8XNjf5p0pITjFOrj8jl5eBARRmOMxdQFrY0W3FfpNsi6thU2E-qHNE0WvKn5EJuO92W1LH1SqrUXU4LKkwM6xsgSJJe9Ycjv2SwcJUGCBOp1hlYImDhcOXF_GIY9rx6CnbzveY3-5pNdYcnPsHHuKyiM7WrMoT7xfzf_DWVTzME4kHp3mbCzufBJAuDkhDeLGJ90wYrPLN9zht0DmYORchXg4rD5lSF8Mc24DhMIQZT6XzPbFGKE-dnixVQJkzKTSo0WkUUzbjWJ0NmiYh0m5vlgKCB9U3rfenlK3ogrKzfrPV6bd75-12p92_6J21G3RJGc6aOO51eq3BRb-LhnWDPmYMreYgf85bnUG30-2iB4QC767b_HLL7rj1X8ClXlE)

Explication :

Ce diagramme de séquence illustre le processus par lequel l’utilisateur peut choisir la langue d’affichage de l’application.

Sur l’interface graphique de l’application, plusieurs boutons permettent à l’utilisateurs de choisir une action à effectuer. Par ailleurs, une liste déroulante concerne le choix de la langue utilisée pour les affichages dans le logiciel.

Au clic sur le champ de la liste, cette dernière se déroule pour faire apparaître les différentes langues disponibles. L’utilisateur peut alors choisir la langue dans laquelle il souhaite continuer sa navigation en cliquant sur l’option adéquat dans la liste déroulante.

Une fois que l’utilisateur a cliqué sur un élément de la liste, un objet de la classe French ou de la classe English est créé selon la langue choisie. Celui-ci servira pour les traductions des différents affichages au sein de l’application.

Enfin, l’ensemble des textes affichés sur l’Interface Homme Machine est instantanément traduit.

* 1. Diagramme de séquence pour la création d’une sauvegarde :

Une image contenant texte, diagramme, ligne, Tracé

Le contenu généré par l’IA peut être incorrect.

Lien vers le [diagramme de séquence pour la création d’une sauvegarde](https://mermaid.live/edit#pako:eNqtVttOGzEQ_RXLL4AIUQgRgX1ASqmEUItalUKlKi_GO0lcvHbwJUAR_9JH8h37Yx17c9lsNiDU5iW2d-bMzPGZ2X2iXKdAE2rhzoPi8FGwoWFZXxH8Me60IVcWDGGWXDkhhWUOvCkej5lxgosxU46cXZ0HmzPDxiPBmSy8zpUDM2Ac1h0umFA_hEr1_fqza2HFjaxx-sD4rR-HQLNVQs6L1SbbU62c0VKC2WRxgQTImoKkvmHymhnBMBW7bvDdMGUl0vFaiN7XSEv4O9N6KGHp1lcLksUE95GxytHGGJHdvR2yi7wn5FSKOx_CWW9In56afMqc0IqkQCzzExgyk0KfVtCrV7DcB-jlLiGXI31_aiCCbu-sW5-ckN3ZpZHCejukUTj06cxjbrGH6HtleLwCMRBYkdxizhlx4x2WEc1Rcu6xT0MlHIxDJ0vyqcynGShn6_KOjHwDlRogkhG-JGPLqzIfZFKW2ZzRJaG4L_j88omMNa4mTIo0ZAk2XDLL8hdncG0ZAtnC5tWIb9E8k-wbRFeFjekGDyjO515Vq7JvEqvt2dtV47JJ7IqkEGAGeClAlM4axGKRHBp4H9YJFfNsEPc4hjJQdA7FrWAFWWIedZSsuAV1zPMsx0dqlUY_rGYDQNm3zM_7UMqc7ZB1tN4v7R3Eqy6JKf8TDnBAYmOlURTzR3Yz-G5lyqBw8yn343xqos4IH0EmUEieDAQfhRYJLe00v2XDtTgEHjA8zoF5yBQWvV47zt5dbw1zhC1y2wRabXd8NhAmg2q3wCql9eqvGYnJck7i2Pm8ci0Mp4XLpzFMeI1NgGNXc56_2OZiMNVgFsFwbkfw1AuUzvuwC-gw-SMBm_OeCfN96MTFrNwrBVRp_1_BVufXXhyYvUHQQNDkvyCX9Xp1vnYWhhZtUBROxkSKHy1PwaRPHTYJvt0SXKYwYF66wP4zmjLv9OWj4jRxxkODGu2HI5oMmLS48-MUYWdfPHMTfG3_1DpbGOGeJk_0gSbt_U7zoHN8fNTpdlutdrfTbtBHmuw3u-1O66B9dHzY7bS6h4cHzw36O0LsN1uzX_tw_7jVOXj-CyvhT-E)

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqlVttOGzEQ_RXLL4BIEEsgkH2oRKmEUItatYVKVV6Md5K4eO3gS4Ai_qWPzXfsj3XszWWz2dCi5iW-zJyZOT6e9RPlOgOaUgt3HhSHd4INDcv7iuCPcacNubJgCLPkygkpLHPgTbk9ZsYJLsZMOXJ-dRFszg0bjwRnsvS6UA7MgHFYd7hkQn0TKtP363vXwoob2eD0lvFbPw6BZqOUXJSjTbZnWjmjpQSzyeISCZANBUl9w-Q1M4JhKnbd4Kthykqk46UQp58iLeHvXOuhhKVbXy1IFhOcR8ZqSxtjRHbbO2QXeU_JmRR3PoSz3pA-PTPFlDmhFcmAWOYnMGQmgz6todePYDkP0MtZSr6M9P2ZgQi6vbNu_eYN2Z0dGimtt0MapUOfzjzmFm1Eb1fh8QjEQGBFcos5Z8SNd1hGNEfJucc-DZVwMA6dLCmmspjmoJxtyjsy8hlUZoBIRviSjC2vqnyQSVVmc0aXhOK85PPjezLWOJowKbKQJdhwyCwvfjuDY8sQyJY2L0b8G80zyf6F6LqwMd3gAeX63KtuVfVNY7Wn9nbVuGoSb0VaCjAHPBQgSuctYrFIDi08D-uEinm2iHscQxUoOofiVrCCLDGPJkpW3II65nlW4yO1SqMfVrMBoOpb5ed1KFXOdsg62ukP7R3Eo66IqfgVFrBB4sXKoijmW3Yz-G6ty6Bwiyn342Jqos4IH0EuUEieDAQfhSsSrrTT_JYN1-IQeMDw2AfmITNY3PXGdvbqehuYI2yR2ybQ-nXHvYEwOdRvC6xS2qz-hpaYLvsktp0PK8fCsFu4YhrDoPwImwDHa8158dvuLTpTA2gZDRt3RM-8QO28ErzEDr0_UrA585k0XwlPXMzLvVBCnfl_jRY--C8FW21h7dgzTwdBBkGW_1VHVbNXF2troXHRFkXx5Exk-HB5CiZ96vCi4BcuxWEGA-alC_w_oynzTn95VJymznhoUaP9cETTAZMWZ36cIezs1TM3wU_3d63zhRHOafpEH2jaTjq9ZC85PjnuJN2jg-Pk8KRFH3H95DDZO0iSw16vd9Q72O90n1v0Z8To7HVOksNubz9JOvvd4-5Bp0UhE_iyuiyfXvEF9vwHuGla3A)

Explication :

Ce diagramme de séquence illustre le processus par lequel une sauvegarde peut être créée et enregistrée dans l’application.

Sur l’interface graphique de l’application, plusieurs boutons permettent à l’utilisateurs de choisir une action à effectuer. Un de ces boutons concerne la création d’une sauvegarde qui sera enregistrée dans le logiciel.

Au clic sur le bouton, plusieurs éléments qui n’étaient pas visibles jusqu’alors sont affichés. Cela comprend trois champs pour la saisi du nom de la sauvegarde, sa source, sa destination, ainsi qu’une liste déroulante pour le choix du type de la sauvegarde. Un bouton « OK » permettant de valider les paramètres entrés est également affiché.

Au clic sur le bouton de validation, les différents paramètres saisis par l’utilisateur sont récupérés. Un objet de la classe BackupModel est alors créé avec ces paramètres. L’instance de BackupModel, correspondant à la sauvegarde, est transmise au contrôleur.

Ensuite, la nouvelle sauvegarde sera ajoutée à la liste des sauvegardes, ainsi que dans le fichier contenant toutes les sauvegardes enregistrées dans le logiciel via le contrôleur.

Pour finir, un message informera l’utilisateur de la création de la nouvelle sauvegarde.

* 1. Diagramme de séquence pour le blocage logiciels métiers, de l’exécution de sauvegardes :

Une image contenant texte, ligne, diagramme, reçu

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme logiciels métiers : [Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqdVdtOGzEQ_RXLLyQipSUhhOwDUrgUoRZREaWVqn0xu7OJhWNvfQmkiP9p-hv5sY7XgSa7GyqaB8SOx8fHc854HmmiUqARNfDDgUzgjLOxZtNYEvyxxCpNRgY0YYaMLBfcMAtOh-WcacsTnjNpycXo0udcaJZPeMJE2HUpLeiMJVDdcMW4_MZlqu6raycsuXP5qZJWKyGg5rhTNc0FWAiZQ6uR1njuGTyvvD_jWQYapOWAGJuJdUeaGrBqNJYvpeEzjFTu8febvGuufUVkaPG8KyU5FpXL8YkzXIIxQ5XZe6ah0fwXwGfF0sGMccFuBZS3vRDyhQ-hQgLE2EVxInIqOEqsiXGaCCC3ylklSUyvP8WUfFEYhYflInGoGHESiGFuBmOmU6jjVUDe-Px8udDgIY0vKJsuf1mN_6c7KziOp6RVuLLGHrRe1QhJDvLcq0hAkkzJJIA6Yuc5FOA-dbkYc8ywGNikkkwUN9zEtFSp-uNC0haDIcmqJ6LVZa4lfOQCFfEkZlgWnmErrLhq77KYXpqy7jdOSrQD6ljmV-c-n5DCW1Jeu-W6psfHZLcsSkTOH8BbIiwMzFwmjVUIbxuijY2cRrPZ3K5w9YCYft2oFJv5QkEBid_ozddKFgomLBlJItQYuxkEmS4X2PToZ2O9YRL09qYfw7ZtHAtrDzJkNMFATN8GTV8DLzX0QOvlb99vYrNZzFq3mIAHwgAZuMTVkJE7_3XTEpnzl-4XmwRIY6Z48WakYThMQ88tF2Fk4LH4V0uvm9ja9diS_o3JhTN-IUXu-JKZlYQg0xpv102BtYTy07u2hBpWYv45pC06BT1lPMWZ9-hTYmonMIWYeiumTN95BZ8wj-H7OES_08hqBy3q8hRRVvPxOQipf82vwgwtRmmL4jyh0SN9oNF-e6_b6Xc67f5-p3fUbnfaLTr34d5er9s9PMS1fvfo4LD71KI_lULUD3v98DvaP-j1-wdd3KGVG09olDG0QIH-vUj1BJ7-AGiTwTc)

Explication :

Ce diagramme de séquence illustre la gestion des logiciels métiers dans EasySave 3.0, un aspect essentiel du système permettant de détecter les applications critiques en cours d’exécution et de bloquer temporairement les sauvegardes si nécessaire. Cette fonctionnalité assure l’intégrité des fichiers et évite les conflits avec des processus métiers en cours.

1. Initialisation de la Surveillance des Logiciels Métiers

Dès le lancement du programme, MainWindow initialise le suivi des logiciels métiers avec les appels suivants :

* StartMonitoringBusinessSoftware() : Cette méthode active un thread de surveillance qui détecte en continu si un logiciel métier est actif.
* LoadAvailableSoftware() : Charge la liste des logiciels métiers définis par l’utilisateur dans les paramètres.

Cette surveillance est permanente et fonctionne en arrière-plan tout au long de l’exécution d’EasySave.

2. Déclenchement de la Sauvegarde

L’utilisateur interagit avec l’interface graphique (GUI) en cliquant sur OK pour lancer une sauvegarde.

* MainWindow récupère les paramètres d’exécution, notamment le type de sauvegarde, les fichiers concernés et les stratégies définies.
* BackupController est sollicité pour exécuter la sauvegarde via ExecuteBackupAsync().

3. Vérification de l’Activité des Logiciels Métiers

Avant d’exécuter la sauvegarde, BackupController effectue un contrôle critique à l’aide de IsBusinessSoftwareRunning() :

1. Si un logiciel métier est détecté :
   * Un message d’alerte est affiché à l’utilisateur.
   * L’exécution de la sauvegarde est bloquée immédiatement.
   * Les sauvegardes en cours sont mises en pause jusqu’à ce que l’application métier soit fermée.
2. Si aucun logiciel métier n’est actif :
   * BackupController autorise l’exécution des sauvegardes.
   * L’exécution passe à CompleteBackupStrategy, qui choisit la méthode de sauvegarde (complète ou différentielle).
   * BaseBackupStrategy gère le processus de sauvegarde fichier par fichier, en continuant à vérifier si un logiciel métier se lance en cours d’exécution.

4. Intégration avec les Stratégies de Sauvegarde

* CompleteBackupStrategy initie la sauvegarde en appliquant l’algorithme approprié.
* BaseBackupStrategy gère la copie de chaque fichier individuellement via BackupOneFile().
* Avant chaque fichier copié, BaseBackupStrategy appelle IsBusinessSoftwareRunning() pour vérifier si une pause est nécessaire en cas d’activité d’un logiciel métier.

5. Impact sur le Système

L’intégration de cette fonctionnalité dans EasySave 3.0 permet :

* Une gestion proactive des conflits : Les fichiers critiques ne sont pas sauvegardés lorsque des logiciels métiers sont actifs.
* Un arrêt et une reprise automatique des sauvegardes dès qu’un logiciel métier est détecté ou fermé.
* Un contrôle en temps réel grâce à une surveillance active en arrière-plan.
  1. Diagramme de séquence pour l’exécution des sauvegardes :

Une image contenant texte, diagramme, Parallèle, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour l’exécution des sauvegardes :



Explication :

Ce diagramme illustre le processus d’exécution des sauvegardes dans EasySave 3.0, mettant en évidence la prise en charge des stratégies de sauvegarde complète et différentielle, l'intégration avec la gestion des logiciels métiers, l'exécution parallèle des fichiers, et l'affichage en temps réel de l'état d'avancement.

1. Déclenchement de l’Exécution

L’utilisateur interagit avec l’interface graphique (IHM) en sélectionnant une ou plusieurs tâches de sauvegarde à exécuter.

1. L’IHM affiche la page d’exécution en appelant ShowExecution().
2. L’affichage des sauvegardes disponibles est transmis par ControllerBackup.
3. L’utilisateur entre un numéro de sauvegarde (ex: 1-3 pour exécuter les sauvegardes 1 à 3).
4. BackupController reçoit cette entrée et parse la sélection avant de déclencher ExecuteBackupAsync().

2. Vérification des Logiciels Métiers

Avant l’exécution des sauvegardes, BackupController effectue une vérification critique avec IsBusinessSoftwareRunning() :

* Si un logiciel métier est actif :
  + L’IHM affiche un message d’alerte et met en pause toutes les sauvegardes en cours.
  + L’exécution ne reprend que lorsque le logiciel métier est fermé.
* Si aucun logiciel métier n’est actif, la sauvegarde se poursuit.

3. Sélection et Exécution de la Stratégie de Sauvegarde

1. BackupController appelle BackupStrategyFactory pour déterminer le type de sauvegarde à appliquer :
   * Complète → CompleteBackupStrategy
   * Différentielle → DifferentialBackupStrategy
2. Une fois la stratégie sélectionnée, BackupController récupère les informations nécessaires (chemins des fichiers, priorités, historique des logs).

4. Gestion des Fichiers et Exécution en Parallèle

Une fois la sauvegarde démarrée :

1. Tri des fichiers :
   * Les fichiers sont classés par priorité en fonction des extensions spécifiées par l'utilisateur (extensions.txt).
   * La taille des fichiers est évaluée pour appliquer des restrictions sur les fichiers volumineux.
2. Traitement des fichiers :
   * Fichiers volumineux → ProcessLargeFileAsync() (traités un par un via sémaphore).
   * Fichiers plus petits → ProcessSmallFileAsync() (traités immédiatement en parallèle).
3. Copie et cryptage :
   * BackupFile() est appelé pour chaque fichier à sauvegarder.
   * Les fichiers sont copiés en respectant l’ordre des priorités et des contraintes de taille.
   * Une fois la copie terminée, CryptoSoft est utilisé pour crypter les fichiers si l’option est activée.

5. Mise à Jour en Temps Réel et Affichage de l’État

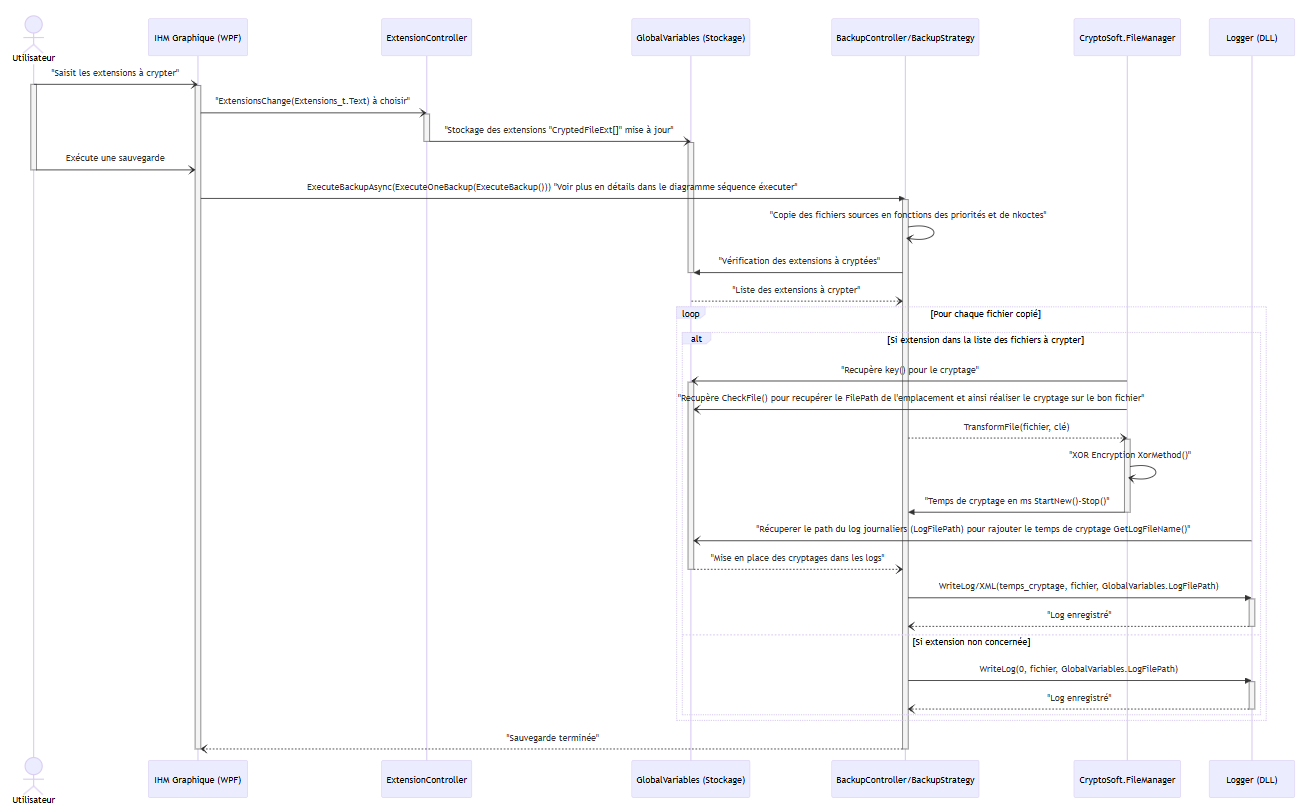
Pour suivre la progression des sauvegardes :

1. BackupController met à jour BackupStateJournal via ComputeState().
2. BackupStateJournal stocke l’état des fichiers et met à jour RealTimeState.
3. L’interface utilisateur (IHM) récupère l’état en temps réel et affiche la progression (JSON/XML).

6. Finalisation et Gestion des Logs

1. Arrêt du chronomètre une fois la sauvegarde terminée.
2. Création des logs :
   * Les informations sont stockées via Logger.WriteLog() ou Logger.WriteLogXML().
   * Les logs contiennent le nom des fichiers sauvegardés, leur taille, la durée de la sauvegarde et le temps de cryptage.
3. L’IHM affiche un message de confirmation indiquant que la sauvegarde a été effectuée avec succès.

* 1. Diagramme séquence pour le cryptage des fichiers choisis :



Lien vers le diagramme séquence paramètre de cryptage :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNrVV21v2zYQ_isEv0wGHDcvbWr7Q4EtTbMBdhvUaRJsHgpGuticZVIlqTZekP-yj9Pv0B_bkZRsWS_Z0G8TYMA-8u4ePs_diX6koYyAjqmGLymIEN5ytlBsPRcEHxYaqcgnDYowTT4ZHnPNDKTKLydMGR7yhAlDpoyLGy4i-c1u_eXnKblQLFlyjEqCm8t3vabL-YMBobkUZ1IYJePYp2kxN30vYnnH4mumOLuLQVu_uimYGRmu2AJaUv_EwlWa7Oet2154w8woPPJi0wzyjscwZQIzOP8ztUmMnMl7M6isNN0mclF4FN-Ct5MJYvQ7HdkHvQqfYzKnM8Y1N8QeC0p6NMn_IqFNilnoVjD-FeG6MDXTLqJfqCiG-VpYt4m3Zn22ZGIBwc7w2QyuEEzPwVhKBNiE0allm_aIoqahO3qhIon2Dz-njnCILNkY7rff55SsuQaL5w-ZNsHUgvvVCJ4H263K-UOehSn6pQKIZulXWDAVQSPsTooq42_eNArOhgQb0S_8qDciDArTB1FYg709Qa_XQyauJVckiVMkSJAozwzjsSYRQ5piJM639BpR5pnvcpJnPkyTpTqqkoJGyyAdzROgKjLhXqx7Hi45KE00qhGCA3cvBSay-tkdieJScZNnuGbQQsRKhgaloV05kbaWGrnOM8UxHbOh64VSdkme7QJX5GmtivosOeg47IRrA10JPbk-YCxlQi6RB2wVZmdiQQ4Jka4885ucDrEhM74LV6jISLxNteW1kmjrb5_qYGpvqo8ofpL_rYCsYBP0SGKRYam4cOhX8rRF9Sxb35P3bAnhyjqU2ZVbyhQ4IHblkpmlLYr4B1gnMQthDTg-sVCwjTQnKs8YkgJ7wIn2B7lD5gqe6mdpFhVirUAfkyuFnN9LtXb4ijB9EsZ51uvgpTHxOzjZyzOntx8-knPhwFutb6WaglnKKOjVUe_FaR0ec3qFPNnG2rGBHbfWZGbw9fMevgW9A5ymSTN4pR06D1K8rDp0taMwgUK8xAmXYtEv3CQWqJOt1wBjlMKWqjPcYLybacC_AFO4vGdraOL-r206ta8F5MJVkeuhMsV2SmqLVj9DzLOV_28VhoJ5_sbkBgce4I8Xt9NJ4I78uQTTJ9taq2UbVJnrKEGfoF20rvGFAoFQsMDZgu30zOnrwSFGRvfmlMBPiNMdlMBJ-x2EBIf_p_OLyP9wX7reVi33uPKagNWu1txR1Xwn1a9qlaXm65n26RpjMR7hNf7ROsypWeKwnFObMmJqZVM84T6W4gUVbxV0bFQKfZomEcYsrvylESKOt_6p_1vg_h30KV5c6fiRPtDx0evB8cujo9Phy9fD49PD4dGwTzd0fDI4OXn1auRtp6OT0VOf_iklBj0cjIrneDQ6PBqeHPepkuliScf3DKvIBf_VbbX5n_4BYdJE0A)

Explication :

Ce diagramme illustre la manière dont EasySave 3.0 permet aux utilisateurs de paramétrer les extensions de fichiers à crypter et de gérer le cryptage automatiquement après chaque sauvegarde. Il met en avant l’intégration entre l’IHM, les modules de gestion des extensions, le mécanisme de sauvegarde, et CryptoSoft, l’outil externe de chiffrement.

1. Paramétrage des Extensions à Crypter

L’utilisateur sélectionne les extensions des fichiers qu’il souhaite chiffrer.

1. Saisie des extensions dans l’interface (via MainWindow).
2. Transmission des extensions choisies à ExtensionController.
3. Mise à jour des variables globales (GlobalVariables.CryptedFileExt[]) pour stocker la liste des extensions à crypter.
4. Confirmation de la mise à jour, les extensions sont désormais enregistrées pour toutes les sauvegardes futures.

2. Déclenchement de la Sauvegarde

Après la configuration des extensions à crypter, l’utilisateur lance une exécution de sauvegarde.

1. L’IHM appelle BackupController via ExecuteBackupAsync().
2. BackupController commence l’exécution des copies en respectant les priorités et contraintes de taille (NKoctets).
3. Vérification automatique des extensions des fichiers copiés :
   * Si une extension fait partie des fichiers à crypter, elle est transmise à CryptoSoft.
   * Sinon, la copie se poursuit sans cryptage.

3. Cryptage des Fichiers Après Copie

Une fois un fichier copié, BackupController vérifie si son extension est dans la liste des extensions à crypter.

1. Si l’extension est à crypter :
   * CryptoSoft.FileManager récupère la clé de cryptage (GlobalVariables.Key()).
   * Il identifie le chemin du fichier copié pour garantir que le bon fichier est traité.
   * L’algorithme XOR est appliqué via XorMethod().
   * Un chronomètre est lancé et stoppé pour mesurer le temps de cryptage.
2. Mise à jour des logs après cryptage :
   * BackupController enregistre le temps de cryptage en millisecondes.
   * Logger (DLL) est utilisé pour mettre à jour le fichier log journalier (JSON/XML).
   * Le fichier log intègre les informations suivantes :
     + Nom du fichier crypté
     + Durée du cryptage
     + Chemin du fichier source et destination
3. Si l’extension n’est pas concernée :
   * Aucune action de cryptage n’est effectuée.
   * Un log standard est enregistré pour le fichier concerné.

4. Finalisation du Processus

1. Tous les fichiers sont traités en boucle jusqu’à la fin de la sauvegarde.
2. BackupController informe l’IHM que la sauvegarde et le cryptage sont terminés.
3. L’interface affiche un message de confirmation indiquant que l’ensemble des fichiers a été copié et chiffré si nécessaire.
   1. Diagramme de séquence pour la mise en pause des sauvegardes :

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour la mise en pause des sauvegardes :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqVVN1u2jAUfhXLV0MDBAQCyUWltpMQ0zpV69CkKTdufALWEjvzT1eGeJe9y15sx0n5T5nqm8TH3_n5vmOfNU0VBxpTAz8dyBQ-CLbQrEgkwcVSqzSZG9CEGTK3IheGWXC6Pi6ZtiIVJZOWTOczj5lqVi5FyvLaayYt6IylcO5wx4T8JiRXv87Pblj6w5W3Slqt8hwa0n3VTJoca7kEur6vavKfqVKLHPZuidwxFE-4r8qtTVXhnRZ5j5RicpsLFEYT4zTBAI_KWSVJQu-ZM5DQ2sWT9x57TjGpAA-QQ2qB14TMu1aN3-NI5-rqjK531gY-qseZ5PC89TqFkQ7mPExZsSvAWg0kZwSbhcQ4GGKYe4IF0_7_7x8CWYZVuS3hw2pajcUgk9r8eilNjlOwtfEzK-BmdUTmOGkl9bXvBRzL_AVKDZJrlLrbqF3DRYj3xm22BlTljVejQnMnULOEfjoRyyCcgMRbhRJ0t-32F8pr_3rulyb8PyCxVXJ7ocwLTX5r_CbRs0ykS8zz1lgcdm_n9CkfHGGOM5t_YLRNC9AFExxnz9pDEmqXUGCfY_zlkDGXWy_4BqEMb8PDSqY0ttpBm2rlFksaZyw3uHMlx7Avg2sLwQHwXaliB8I9jdf0mcadfnfYD8JxOOqNw3AyioJRm65oHETdQS8chFHQD8e4Jps2_V2FGHTHvfEkGA4ng340iKKgTYELHI139eysRujmH74-yk4)

Explication :

Pendant l’exécution d’une sauvegarde, plusieurs boutons permettent à l’utilisateur d’agir sur la sauvegarde en cours. L’un de ces boutons concerne la mise en pause de la sauvegarde.

Pour pouvoir mettre en pause une ou plusieurs sauvegardes, l’utilisateur doit obligatoirement avoir lancé ces sauvegardes préalablement, en cliquant sur le bouton adéquat.

Au clic de l’utilisateur sur le bouton « Pause », la ou les sauvegardes en cours sont arrêter. Toutefois, l’état des sauvegardes au moment où elles ont été suspendues est gardé en mémoire. Cela est nécessaire dans le cas où l’utilisateur souhaite reprendre les sauvegardes là où elles ont été laissées.

Ensuite, le bouton permettant de reprendre une sauvegarde là où elle s’est arrêtée est activé. Effectivement, le bouton de reprise devient actif uniquement après le clic sur « Pause » car il n’est pas possible de reprendre une sauvegarde qui est en cours (et n’a pas été mise en pause).

Enfin, un message, préalablement traduit par l’API de Google Translate, informe l’utilisateur sur l’état de ses sauvegardes.

* 1. Diagramme de séquence pour la reprise des sauvegardes :

Une image contenant texte, diagramme, ligne, Police

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour la reprise des sauvegardes :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqVVNtuGjEQ_RXLT0UFBCwsrB8ipamEqJoqSooqVfvirGfBqtfe-pImRfxL_6U_1tndcAkQqvhl1-NzZubM2LOimRFAGXXwM4DO4KPkC8uLVBNcPPPGkrkDS7gjcy-VdNxDsM1xya2XmSy59mQ6n1WYqeXlUmZcNayZ9mBznsEx4ZpL_U1qYX4dn33g2Y9QXhntrVEKToT7arl2CnM5B7q8qXOqPlNjFgp2tFRvFcoH3NfpNqY68U6LvEdJjFwpiYWxxAVL0MG9Cd5oktJbKC1oYSGlDa0qQMXa6WLkFlwo4A4UZB5Eo8q9azWEHZB0Li6ONDNyw62DT-Z-pgU8bliHMNLBoPsxa4kFeG-BKE6wY6hOgCOOhwdYcFv9__1DIM8xq7BRvZ9N60QyjZTG_noup5hT8I3xC0cHTy_UHNTgRFfZzrghnUDVbOxzjRZBovaUfj4QjWhisW3SdTdNq65GVcDXAz9X8j_eiK_D-jMJnmnTm5y_bFR9Ry_zXGZLDPImRwK2l__wLe4dYYAjW_VCaJsWYAsuBQ6PVQVJqV9Cge-B4a-AnAflqzqvEcrx1dw96YwybwO0qTVhsaQs58rhLpQC3T5Png0EX_B3Y4otCPeUregjZZ1-d9iP4nE86o3jeDJKolGbPlEWJd1BLx7ESdSPx7gm6zb9XbsYdMe98SQaDieDfjJIkqhNQUicbdfN8Ktn4Pofkpm2ng)

Explication :

Pendant l’exécution d’une sauvegarde, plusieurs boutons permettent à l’utilisateur d’agir sur la sauvegarde en cours. L’un de ces boutons concerne la reprise de la sauvegarde.

Pour pouvoir reprendre une ou plusieurs sauvegardes, l’utilisateur doit obligatoirement avoir mis ces sauvegardes en pause préalablement, en cliquant sur le bouton adéquat.

Au clic de l’utilisateur sur le bouton de reprise, la ou les sauvegardes en pause continues leur exécution. Cela est possible car l’état des sauvegardes a été gardé en mémoire au moment du clic sur le bouton permettant la mise en pause.

Enfin, un message, préalablement traduit par l’API de Google Translate, informe l’utilisateur sur l’état de ses sauvegardes.

* 1. Diagramme de séquence pour l’arrêt des sauvegardes :

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour l’arrêt des sauvegardes :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNqdVMtu2zAQ_BWCpxq1DT9lW4cAeQBGiqYIkgYFCl020somKpEqH2lcw__Sa_ob_rEupdhOYjloootEcoa7szvaJY9VgjzkBn86lDGeCZhpyCPJ6IHYKs1uDGoGht1YkQkDFp2ujgvQVsSiAGnZ9ObcY6YairmIIatY59KiTiHGfcIFCPlNyET92j87gfiHK06VtFplGdaE-6pBmoxyeQ10fFnm5F9TpWYZ7miR3CoUd7Qu0622ysRbDfaRJIXsNBNUGM2M04wuuFXOKskifm1VEfGK4bV7wk5SyPz5NWYYW0wqOeZDo4LvYKx1dLQnNmSXoA1-UrfnMsH7DesljLUo5NOIpbYcrdXIMmDUKpKVoGEG3B3OQPvv9R-GaUpZuY3cp9k0apLxQqrdw5nU8aZoq80vkOPJ4pmW5zHLOh_7RuDzGl9msIh4-yDnbP1goI4HzuB7iFdYaJSJfhe5ckS7tsc1dg13m5uy1KBKNhm4RCdOUG8j_vlFUw3BGWi9_mvXD2jaG19643uXHI7-aJf_uZLZMgH7SqqvGPLtEeo8kqYinlOkt9-W4PZPfzl4nhxRlL09Pw54k-eocxAJTcqlh0TczjEnn4T0mWAKLrO-7CuCAhnieiFjHlrtsMm1crM5D1PIDK1ckdC1j2N2A6Fx9V2pfAuiNQ-X_J6HrW570O0Ho2DYGQXBeDjpD5t8wcP-pN3rBL1g0u8GI3rGqyb_XV7Ra486o3F_MBj3upPeZNJvckwEDfKLatKXA3_1D8RdC1A)

Explication :

Pendant l’exécution d’une sauvegarde, plusieurs boutons permettent à l’utilisateur d’agir sur la sauvegarde en cours. L’un de ces boutons concerne l’arrêt total de la sauvegarde.

Pour pouvoir arrêter une ou plusieurs sauvegardes, l’utilisateur doit obligatoirement avoir lancé, mis en pause, ou repris ces sauvegardes préalablement, en cliquant sur l’un des boutons adéquats.

Au clic de l’utilisateur sur le bouton « Stop », la ou les sauvegardes en cours sont arrêtées et annulées. L’état des sauvegardes n’est pas gardé en mémoire dans le logiciel. Ces sauvegardes ne pourront donc plus être reprises.

Ensuite, le bouton permettant de lancer une sauvegarde est activé. Les trois autres boutons, permettant la mise en pause, la reprise, ainsi que l’arrêt total des sauvegardes, sont désactivés. Effectivement, une fois les sauvegardes en cours stoppées, il n’est plus possible de les mettre en pause, de les reprendre, ou même de les arrêter une deuxième fois. En revanche, lorsque l’utilisateur arrête des sauvegardes, il lui est à nouveau possible de lancer d’autres sauvegardes.

Enfin, un message, préalablement traduit par l’API de Google Translate, informe l’utilisateur sur l’état de ses sauvegardes.

* 1. Diagramme de séquence pour la consultation de l’historique :

Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour la consultation de l’historique :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNp9VO1O2zAUfRXLf2iltJCOljY_kCbY2CbKJjHYNFVCt4nbeHPszLFLO8T7wHP0xXZt92sNLH9iX1-fe-45th9oqjJGE1qx35bJlJ1zmGooRpLgB6lRmtxUTBOoyI3hgldgmNVhuQRteMpLkIbc8oqPcd0sXOp2Vs_8-GHoUobA5TcuM3XfnkMhDtOqnirUtHK57l9fvRBqDOIWNIexYD5xL1Tfc6mm09DNatQ4v7xsjmTI9J22mp5iQs4ER00wudTLJ4SfKa5JmivsDSlhQQSGYvlkNI5H9EzJygqDG9yap0w3MvIZ6ubx90JYKkRczRbW3kqXkOtc3TugRhPxNwvLZ5IxIg4chpK1IvvaZ-zVJV_z9NSTTcg5r0oBi1BwD3TrgPekFURKkNZXDVXBDIpAYDLhaQ5T5vkB6hOGaVAGdunusNpiez7NfRcT59V7LtgXMHmN2YuW7x-NlusyOJ6QC2ZWgFdQMNS2AWXJhFjTnijplcVpab3VJBNiv25AqzXzIp3VWcPW1hxG9N2cV8Y5e7t81nzCcZ25kLuEQeTMkp8K79raYOGPL8bRJyX5Fn63hGt05YxPNlrZGZ6Yxqfrz1eH34d42OkLu5r_bFpVJmNIf9nyDtncLfBrDYetLGuvkUhDqiIiDI2NSIX5KYtQispw6a2OiBxrYpQB4ZR1ZwO7DNHtnFRgZ2wKOls-R8QARyMikisNXOPAsKJEQgdszlLrUDf8majY_zp5a1MrnZJrDXYbX9-2Wu5WXALWiZAd5JZvasqs5vgrB2FzsXdi4f7TiBZMF8AzfHcfXNKImpwVbEQdlwz0L1fvEfPAGnW9kClNjLYsorbMEGf1Rq-DLOP4TA_DO-6f84jiY0eTBzqnSSdux3Gn2--c9OPjbm_Q6UR0QZNu3O4PTnpxtz84PnozODnuPkb0j1KIetQehK_Xi-Nutx_jDlRwmtNkAii6R__hUx2Bx7_8EhOs)

Explication :

Ce diagramme décrit le processus de consultation des logs dans EasySave 3.0, permettant aux utilisateurs d’accéder aux fichiers journaliers contenant les informations de sauvegarde. Cette fonctionnalité assure une traçabilité des opérations effectuées et permet à l’utilisateur de vérifier l’état des sauvegardes via des fichiers au format JSON ou XML.

1. Demande de Consultation des Logs

L’utilisateur souhaite consulter l’historique des sauvegardes en ouvrant l’interface dédiée aux logs.

1. L’utilisateur clique sur "Consulter les logs" dans l’interface graphique (IHM - MainWindow.xaml/cs).
2. L’IHM déclenche la fonction ShowLogs(), permettant d’afficher la fenêtre de consultation.
3. L’interface de consultation des logs est chargée et affichée à l’utilisateur.

2. Vérification et Chargement des Logs

1. L’IHM interroge les variables globales pour obtenir le chemin du fichier log journalier via GlobalVariables.LogFilePath().
2. La DLL Logger est sollicitée via Logger.GetLogFileName() pour récupérer l’emplacement exact du fichier log.
3. Logger vérifie l’existence du fichier log du jour en utilisant la méthode Exists().

3. Affichage des Logs

Deux scénarios sont possibles :

Cas 1 : Logs disponibles

1. Logger confirme la présence du fichier log (JSON/XML).
2. Les données du fichier sont récupérées et envoyées à l’interface.
3. L’IHM affiche les informations suivantes :
   * Nom de la sauvegarde
   * État de la sauvegarde (terminée, en cours, annulée)
   * Chemin source et destination
   * Nombre total de fichiers
   * Nombre de fichiers sauvegardés
   * Taille totale des fichiers
   * Horodatage de l’opération
   * Temps d’exécution

Cas 2 : Aucun log trouvé

1. Logger indique que le fichier log du jour est introuvable.
2. L’IHM affiche un message d’information indiquant qu’aucune sauvegarde n’a été enregistrée aujourd’hui.

4. Fin du Processus

1. L’affichage est mis à jour en fonction de la présence ou non des logs.
2. L’utilisateur peut alors naviguer dans l’interface et revenir à l’écran principal.
   1. Diagramme de séquence de la communication client/serveur via socket :

Une image contenant ligne, reçu, texte

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence de la communication client/serveur via socket :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNrtWttu2zgQ_RUuX3aLOkZiJ3ast8TxFgGabWDFKLAIUHAlOhYqk1pe3GSDfM8i-xv-sQ4lX1jdItneh7jRkyzOHJIzZ4bk0I_Y4z7FDpb0b02ZRy8CcifI9JYheIinuEAjFYSBJIpqkXyOiFCBF0SEKfRhdPlFUjGjAhGJPggSTQKPhGgEH9ElU1SMiUeRa0SK9L0woPBapt-PRbLqVyRgnwPm829f4i5EqUgRSqLa50wJHoZ5IImmLbGyUDAD02SNtGopGGJeuz2-VfvaQjkN0kLMdIQO3mU_OshVMLHkx5l8YN5v70r036dtU6yfljTqWe2-mD9Dm2ZIcu8rVSjiWqCQImlThIRgc8KACD5FHp9ONQNaqIAzNObMMy-Mhols9b7PYXbL0VbX-hhIRVl9vVHkg5tGl_U1Xcr8G56wob62pC_P8n2a0A4asBl_gNZbvAhWRBkiCqYOXIu9ACa_B8M3m81bjIhGCSub5T2tWeqgs_E48CaV-1gDW9G1xEw6B8wo0mbYUiewCYanTELSK1JZWFbG-TE--gvEfgIAo7CtmBE02lkrLpRveE6AmCetUTCGtTP6S5NY00lZJ3doP9hobfcKcHlDzJkoxHESkL5Oh7KdrmogJqarT_gzz6OR2q8ALYvPhZMXRJ8_o1-2j8c8zG1ZG-NYNNuKs2VgJekBwGFjk6QHWHP-4lpxttkKsHR4JqargQypR4MZ_YloCss2YZDVBZ3_q-kOOJoGrJthln6skZuzIDl-NLuV85hZ6JpoSdctRTPM7rWcRNWloVm8_HPifdWRtHvZjjHbs2Zz5mzCHpfoGb0jAtwNe4TI2KZ441F381GA_Xqt1f_0x--Xwyt0fTZyByVG2kmcbBUrGy0nlsOmgVx7zZzRrLNDxoV1VplN-qAhtC8if0gjAZ4UG0b_kEo9fQv_pceNNY0XwBshQXLlm53lgNIOXn8iGA7c0dUeZoIit-0yE2zSh50JXMWjzZKA0XxLAdmkT4SY_wcHALr7DYAN_frD3r35dL3Xy__KXf_T0l8JPw52N655DO5N7SGwz5Q_V5R-XJZ7kPyVSoX8-fOqiLCjaH2hiy3JdWGhxbWsHRCqCibYvu6JvB9ySeufWC8C6WVra9V0DckW5-76ylUqCG_Vln2ttrgTrSBaWH3eVOJ5XuXe1ouTdE563m8aDIQwVdDVvUdlFhwU0CANWO6Ug9JUu8ACTi1TuLmJ2LQsXI62Sq8-fekqNl_Cvoy1JNLXsakmmY9rVahxA0-pmJLAxw5-NJK3WE3oFCLMgVefjokOlZnGE4gSOFG4kAGxo4SmDSy4vptgZ0yA3Q2sY5Yu7uyXIhFhf3I-XQnBb-w84nvsHBw1j4_anW7n5LDb6Zye9NonDfyAnXav2TrstDq99lGnC8_pUwP_E0O0mt3D7mn7-Pi0ddRr9XrtBqZ-oLi4Sv42EP974Ok7Wr7FCg)

Explication :

Pendant l’exécution d’une sauvegarde, il est possible de lancer une autre interface graphique de l’application. Cette dernière permet à l’utilisateur d’agir sur les sauvegardes en cours d’exécution.

L’interface graphique déportée est mise en place grâce à des socket entre celle-ci et l’interface principal du logiciel.

La deuxième interface comprend un bouton permettant d’établir une connexion (via socket) avec la partie concernant l’exécution des sauvegardes de l’interface principal.

Une fois la connexion établie entre les deux interfaces graphiques, l’utilisateur peut alors agir sur ses sauvegardes en cours grâce à d’autres boutons. Parmi ces derniers, il y a un bouton « Pause » pour arrêter les sauvegardes en cours en gardant leur état en mémoire. Il y a aussi un bouton « Reprendre » qui permet de continuer les sauvegardes là où elles ont été mises en pause. Le dernier bouton « Stop » permet d’arrêter totalement les sauvegardes en cours d’exécution.

Pour connaître le déroulement des évènements dans l’interface principale du logiciel une fois que l’utilisateur a cliqué sur un bouton de l’interface déportée, vous pouvez vous référer aux diagrammes de séquence adéquats (Diagramme de séquence pour la mise en pause des sauvegardes, Diagramme de séquence pour la reprise des sauvegardes, Diagramme de séquence pour l’arrêt des sauvegardes).

Enfin, des messages sont également affichées sur l’interface graphique déportée de l’application pour informer l’utilisateur que les actions effectuées ont bien été prises en compte.

* 1. Diagramme de séquence pour quitter l’application :

Une image contenant texte, capture d’écran, Police, diagramme

Le contenu généré par l’IA peut être incorrect.

Lien vers le diagramme de séquence pour quitter l’application :

[Online FlowChart & Diagrams Editor - Mermaid Live Editor](https://mermaid.live/edit#pako:eNptU-1q2zAUfZWLoCwBN8RO2iT-URjdug1W2CilMAxDlW9rUUfy9NG1C3mAPYqfwy-2K7ldvGT6pY9zzj33QxsmdIksZxZ_eFQC30l-b_i6UECLC6cNXFs0wC1cO1lLyx160z833DgpZMOVg0su1Y1Upf4ZoJ8-XsIHw5tKkiqMbr5cjA8pb5umloI7qVXgDI9EKFTPODqC7rfjDUIKOXx-43cuQNRR39K2RrjV3hG3YF-9dA5NwXqF6P94PLCYw_mOOcDD6P2TdN9vx3_Tl48UKSrsXe3EDo1m0WjXPnatwjVSqmgdCN44b7oW0EHNYd21rqLaQwg5GkdI2bWipi5UXYu96KCs-xlEOYT6n0CUl3g4pJ6dDcubDw-Tc28MUSdXlXcEVqPxYUYzyugCzRpjyDJE5TuJHj3s33GIOHQbyeC0t9QpC43RAq2lEyoQ2ht76JnyDXUPpRzEAotwF8ReTZb4357sPfUdZAlbE5fLkiZ-E0AFcxVVrmA5bUtuHsLQbAnHaZaunpVguTMeE-abknRefsfrJZaSPshl_4PiR0oYjTbLN-yJ5fPlZJ5N5-kszbJFli5O5gl7Znm2mKTZbDU7XaanJ9l8lS63CfulNalOJ6uXla1W03Q5yxJmtL-vWH7Ha4tR_VuEBgPbPwoiRE4)

Explication :

Ce diagramme décrit le processus de fermeture de l’interface graphique (IHM) lorsque l’utilisateur clique sur le bouton "Quitter" dans EasySave 3.0. Cette action permet de fermer proprement l’application, en s’assurant que tous les processus en cours sont arrêtés.

1. Déclenchement de l'Événement de Fermeture

1. L’utilisateur clique sur le bouton "Quitter" *(Exit\_b)* dans l’interface WPF.
2. L’IHM capture l’événement Click, déclenchant ainsi l’exécution de la méthode Exit().

2. Fermeture de l’Application

1. L’IHM envoie une commande de fermeture à l’application via Application.Current.Shutdown().
2. L’application WPF reçoit l’ordre de fermeture et commence à arrêter tous les processus en cours.

3. Finalisation de la Fermeture

1. L’application termine les opérations en arrière-plan, comme :
   * Libération de la mémoire
   * Fermeture des connexions ouvertes
   * Arrêt des processus en cours
2. L’interface est fermée et l’utilisateur quitte l’application.
3. Documentation CryptoSoft V3.0:

1. Documentation de la Dynamic Link Library V3.0:



1. Documentation Utilisateur V3.0:

1. Documentation Technique V3.0 :

1. Release note V3.0 :

1. Conclusion

Le livrable 3 marque une avancée majeure dans l’évolution d’EasySave en introduisant une gestion optimisée des sauvegardes parallèles et une meilleure répartition des ressources pour éviter la saturation du réseau. Cette version repose sur les fondations posées par EasySave 2.0, qui avait déjà apporté une interface graphique (WPF), supprimé les limitations sur le nombre de sauvegardes et intégré CryptoSoft pour le chiffrement des fichiers. Avec EasySave 3.0, plusieurs améliorations clés ont été mises en place pour répondre aux nouvelles exigences des professionnels.

L’exécution des sauvegardes est désormais entièrement parallèle, permettant une réduction significative des temps de traitement. Une gestion avancée des fichiers prioritaires assure que les fichiers critiques sont sauvegardés en premier, évitant ainsi tout retard dans leur protection. De plus, pour préserver la bande passante et garantir des performances optimales, les fichiers volumineux ne peuvent pas être transférés simultanément, une règle strictement appliquée grâce à un système de gestion de taille configurable par l’utilisateur.

Un autre point crucial de cette version est la mise en pause automatique des sauvegardes en cas de détection d’un logiciel métier en cours d’exécution. Cette fonctionnalité garantit que les tâches professionnelles sensibles ne sont pas perturbées par les opérations de sauvegarde. Une interface de contrôle distant via une console déportée a également été ajoutée, permettant aux utilisateurs de surveiller et gérer leurs sauvegardes à distance grâce à une communication réseau basée sur des Sockets.

Enfin, CryptoSoft devient mono-instance, empêchant plusieurs exécutions simultanées afin d’éviter les conflits de chiffrement et d’assurer l’intégrité des fichiers protégés.

Grâce à ces évolutions, EasySave 3.0 devient un outil de sauvegarde plus performant, intelligent et sécurisé, répondant parfaitement aux besoins des entreprises et des utilisateurs exigeants. En anticipant les futures évolutions, cette version prépare déjà le terrain pour EasySave 4.0, qui pourrait intégrer une gestion dynamique de la charge réseau, ainsi qu’un système avancé de gestion des utilisateurs et des permissions, renforçant ainsi sa position comme solution incontournable de gestion des sauvegardes.