Gestion des utilisateurs dans l’AD de l’établissement

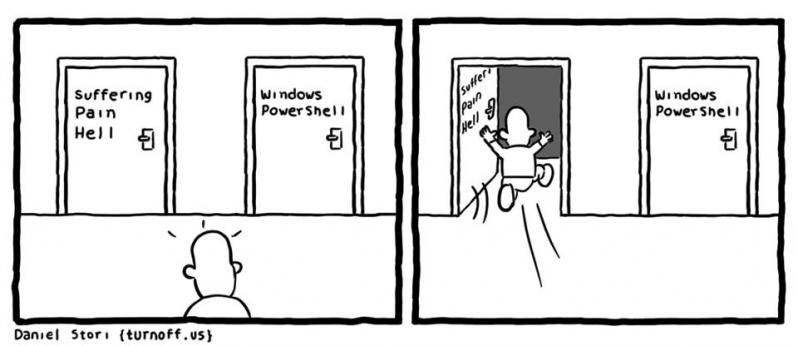


Figure 1 Image de titre

Sylvain Philipona – CIN4B

Ecole des Métiers Lausanne

Avenue de Valmont 30, 1010 Lausanne

89 heures

Roberto Ferrari

08.05.2023 – 07.06.2023

Table des matières

[1 Analyse préliminaire 3](#_Toc136868103)

[1.1 Introduction ✔ 3](#_Toc136868104)

[1.2 Objectifs ✔ 4](#_Toc136868105)

[1.3 Planification initiale 5](#_Toc136868106)

[2 Analyse / Conception 7](#_Toc136868107)

[2.1 Concept 7](#_Toc136868108)

[2.2 Outil de versioning ✔ 7](#_Toc136868109)

[2.3 Stratégie de test ✔ 8](#_Toc136868110)

[2.3.1 Création des comptes 8](#_Toc136868111)

[2.3.2 Modification des comptes 9](#_Toc136868112)

[2.3.3 Suppression des comptes 9](#_Toc136868113)

[2.4 Risques techniques 9](#_Toc136868114)

[2.5 Planification 9](#_Toc136868115)

[2.6 Dossier de conception 11](#_Toc136868116)

[2.6.1 Création des comptes ✔ 11](#_Toc136868117)

[2.6.2 Modification des comptes ✔ 12](#_Toc136868118)

[2.6.3 Suppression des comptes 13](#_Toc136868119)

[3 Réalisation 15](#_Toc136868120)

[3.1 Dossier de réalisation 15](#_Toc136868121)

[3.1.1 Environnement ✔ 15](#_Toc136868122)

[3.1.2 Version du projet ✔ 16](#_Toc136868123)

[3.1.3 Normes de codage et de nommage ✔ 16](#_Toc136868124)

[3.1.4 Création des comptes ✔ 19](#_Toc136868125)

[3.1.5 Modification des comptes ✔ 38](#_Toc136868126)

[3.1.6 Suppression des comptes ✔ 47](#_Toc136868127)

[3.2 Description des tests effectués 56](#_Toc136868128)

[3.3 Liste des documents fournis 58](#_Toc136868129)

[3.4 Glossaire 58](#_Toc136868130)

[4 Conclusions 59](#_Toc136868131)

[5 Annexes 60](#_Toc136868132)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 60](#_Toc136868133)

[5.2 Sources – Bibliographie 60](#_Toc136868134)

[5.3 Journal de travail 60](#_Toc136868135)

[5.4 Manuel d'Installation 60](#_Toc136868136)

[5.5 Manuel d'Utilisation 60](#_Toc136868137)

[5.6 Archives du projet 60](#_Toc136868138)

# Analyse préliminaire

## Introduction ✔

L’ETML, compte plus de 1200 étudiants et apprentis.

Chaque année on compte près de 500 nouveaux arrivants. En début d’année scolaire, il est nécessaire de gérer les comptes utilisateurs des élèves et apprentis. Pour cela il faut :

* Créer les comptes des nouveaux élèves et apprentis.
* Mettre à jour les comptes des élèves et des apprentis en cours de formation.
* Supprimer les comptes des élèves et des apprentis qui ont terminé leur cursus

Ces actions prennent beaucoup de temps, surtout si cela est fait à la main pour chaque compte utilisateur. C’est pourquoi il est nécessaire d’automatiser ces tâches au maximum afin de limiter l’action humaine et donc aussi limiter les pertes de temps et les erreurs humaines.

Le but de ce projet, est de proposer une solution de création / modification / suppression automatique des comptes utilisateur sur l’Active Directory (AD).

La technologie qui sera utilisée dans ce projet est PowerShell. La raison de ce choix réside dans le fait que PowerShell est un langage de Scripting natif à Windows et développé par Microsoft. De plus, y’a de nombreux composants installés par défaut permettant notamment d’interagir avec l’AD.

Le projet se déroule dans le cadre d’un Travail Pratique Individuel (TPI). L’environnement logiciel et matériel est le suivant :

* 1 PC standard de l’ETML
* Suite Office365
* Visual Studio Code
* Un serveur virtuel Windows 2022 avec un AD installé
* Un ordinateur virtuel Windows 10 dans le domaine de l’AD
* Un dépôt GitHub.

Les différents modules qui seront utiles au projet sont les suivants :

* Compétences en Scripting ➔ Module 122 (PowerShell).
* Compétences en OS et Serveurs ➔ Modules 305, 123 et 127 (MS Windows 10 et Server 2k19).
* Compétences en outils bureautiques ➔ Module 302.
* Compétences en gestion de projets ➔ Modules 306 et 431

## Objectifs ✔

Les objectifs techniques du projet définis dans le cahier des charges, sont les suivants :

1. Les scripts sont commentés.
2. Les comptes sont créés correctement, il est possible de se loguer, il est obligatoire de devoir changer son mot de passe.
3. Les dossiers personnels (home directory) sont créés avec les bons droits et sont montés au login du compte.
4. Les comptes sont modifiés correctement.
5. Les comptes sont supprimés / désactivés correctement.
6. Lors de la suppression du compte, le dossier personnel est supprimé correctement.
7. Les fichiers de sortie CSV contiennent toutes les informations nécessaires pour savoir ce qui a été créé, modifié, désactivé, supprimé.

## Planification initiale

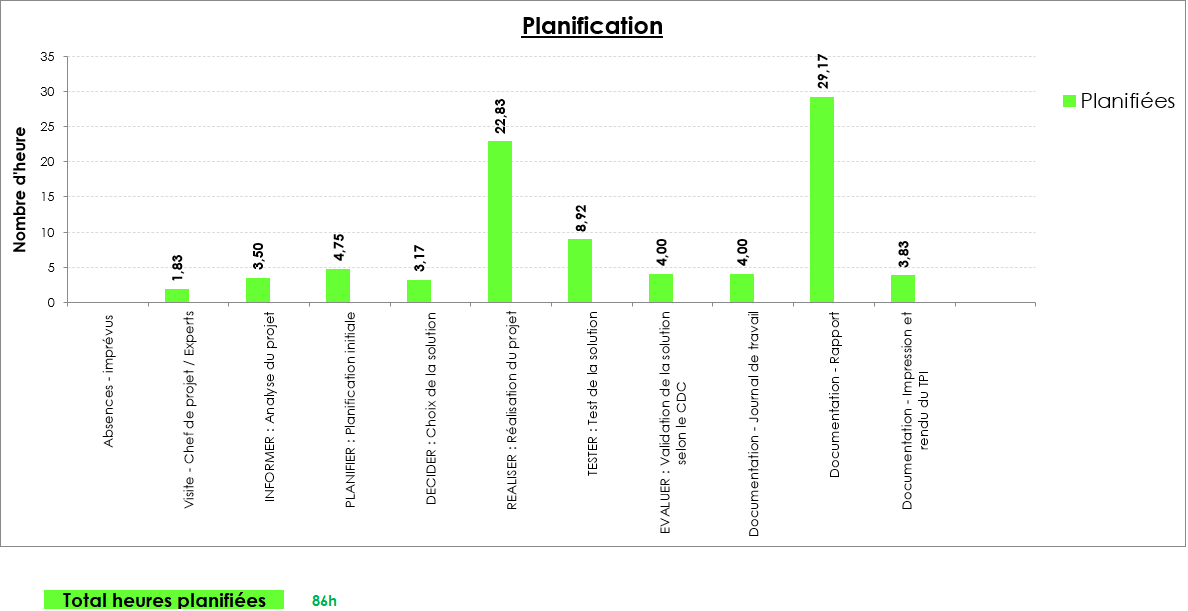


Figure 2 Planification initiale

La planification initiale du projet a été élaborée en suivant méthode des 6 pas.

86H => Car pense finir en avance

*Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.*

*Ces éléments peuvent être repris des spécifications de départ.*

# Analyse / Conception

## Concept

Le concept de ce projet se résume en l’automatisation de la création, la modification et la suppression des comptes utilisateurs. Ces comptes sont utilisés par les étudiants et apprentis pour se connecter aux ordinateurs du domaine.

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

* *Multimédia: carte de site, maquettes papier, story board préliminaire, …*
* *Bases de données: interfaces graphiques, modèle conceptuel.*
* *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle…*
* *…*

## Outil de versioning ✔

L’outil de versioning que j’ai utilisé, est Git avec la plateforme GitHub.

Vous pouvez trouver le répertoire GitHub de ce projet en suivant le lien ci-dessous : <https://github.com/SylvainPhilipona/TPI-AD-Users-Management>

Le répertoire a été organisé en deux branches. "Main" et "Dev". Toutes les fonctionnalités sont développées dans la branche "Dev", puis une fois fonctionnelles, la branche est "merged" dans la branche "Main".

## Stratégie de test ✔

La liste des tests à effectuer durant le projet pour valider son bon fonctionnement est décrite dans les trois tableaux ci-dessous. Chaque tableau correspond aux tests à effectuer dans un script. Tous les tests pouvant être automatisés facilement avec des tests unitaires le seront. Le reste est testé à la main. Les données de tests sont un mélange de données réels d’utilisateurs ainsi que des données fictives et erronées afin de tester les limitations des scripts.

### Tests du script de création des comptes

|  |  |
| --- | --- |
| Test | Impact en cas d’échec |
| Une recherche est effectuée pour vérifier si l'utilisateur existe déjà dans l'AD. | • Cela peut entrainer une réinitialisation non voulue d’un compte utilisateur. |
| Le nom d’ouverture de session est généré avec les 3 premiers caractères du prénom + les 8 premiers caractères du nom de famille. De plus les diacritiques, espèces et apostrophes sont retirés. | • Compromission du compte.  • Impossibilité d’utilisation du compte. |
| Le nom d’ouverture de session proposé est unique, dans le cas contraire un autre est généré. | • Risques de doublons d’identifiants  • Impossibilité de connexion |
| Un mot de passe aléatoire devant être changé à l’ouverture de session est généré. | • L’utilisateur ne peut pas changer de mot de passe.  • Risques de sécurité. |
| Un script d’ouverture de session est généré selon la profession de l’utilisateur. | • Accès inappropriés.  • Difficultés de gestion |
| Une description est générée selon la classe de l’utilisateur. | • Gestion des comptes plus compliquée. |
| L’utilisateur est créé dans la bonne OU. | • Gestion des utilisateurs plus compliquée.  • Problèmes de sécurité liés aux GPO. |
| Le dossier personnel est créé avec les bons droits d’accès. | • Impossibilité d’avoir un dossier personnel.  • Risques de compromission des données personnelles. |
| L’utilisateur est membre des groupes spécifiés dans le CSV. | • Impossibilité d’utilisation du compte. |
| Un fichier CSV contenant les actions effectuées est généré. | • Impossibilité d’obtenir un retour des actions effectuées. |

### Tests du script de modification des comptes

|  |  |
| --- | --- |
| Test | Impact en cas d’échec |
| Une recherche par le nom d’ouverture de session est effectuée pour vérifier si l'utilisateur déjà dans l'AD. | • Impossible de vérifier si un compte existe.  • Peux générer des erreurs dans le script. |
| Le compte est modifié selon les informations du CSV. | • Impossible de modifier les comptes. |
| Un fichier CSV contenant les modifications effectuées est généré. | • Impossibilité d’obtenir un retour des actions effectuées. |

### Tests du script de suppression des comptes

|  |  |
| --- | --- |
| Test | Impact en cas d’échec |
| Une recherche par le nom d’ouverture de session est effectuée pour vérifier si l'utilisateur existe dans l'AD. | • Impossible de vérifier si un compte existe.  • Peux générer des erreurs dans le script. |
| Il est possible de choisir de désactiver ou supprimer un compte. | • Impossible d’utiliser le script. |
| Un compte est supprimé uniquement s’il est désactivé. | • Perte de données.  • Difficulté de suivi. |
| Le dossier personnel est supprimé lors de la suppression du compte. | • Utilisation inutile d’espace disque.  • Risques de fuite de données personnelles. |
| Un fichier CSV contenant les modifications et suppressions est généré. | • Impossibilité d’obtenir un retour des actions effectuées. |

## Risques techniques

Le risque principal de ce projet est lié au remplissage des fichiers CSV par les utilisateurs de ce projet. Les données fournies peuvent êtres mauvaises voir incohérentes. Afin de limiter au maximum ce risque, de multiples tests sur les données fournies sont effectués.

## Planification

*Révision de la planification initiale du projet :*

* *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
* *partage des tâches en cas de travail à plusieurs.*

*Il s’agit en principe de la planification* ***définitive du projet****. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.*

## Dossier de conception

Les différents points (Création, modification et suppression) sont détaillés ci-dessous.

### Création des comptes ✔

Voici comment la création des comptes va se dérouler.

* La première étape consiste au remplissage des données des apprentis dans un fichier CSV. Ces données sont par exemple : Prénom, Nom, Classe…

Une fois ce fichier remplis, l’utilisateur exécute le script de création des comptes.

* Une fois le script exécuté, toutes les données du fichier CSV vont être parcourues apprenti par apprenti. Le script va tester si le compte existe déjà. Si c’est le cas, le script passe à l’apprenti suivant. Dans le cas contraire, la phase de création du compte commence.
* Le script va passer par plusieurs étapes afin de créer le compte :

1. Génération de l’identifiant de connexion. Suppression des diacritiques dans cet identifiant.
2. Vérifie si l’identifiant existe déjà. Dans ce cas l’étape 1 est réexécutée afin de générer un nouvel identifiant. Dans le cas contraire, l’identifiant est gardé.
3. Génération d’un mot de passe aléatoire. Génération d’une description renseignant la classe de l’apprenti.
4. Création d’un dossier personnel (Home Directory) sur le serveur. Création d’un script d’ouverture de session qui va connecter au PC le dossier personnel de l’apprenti.
5. Le compte est créé avec les paramètres générés dans les étapes précédentes et avec les données fournis dans les CSV. Les groupes l’apprenti est ensuite ajouté dans les différents groupes AD.

* Une fois ces étapes terminées, un fichier CSV est généré avec les informations concernant les comptes nouvellement créés.

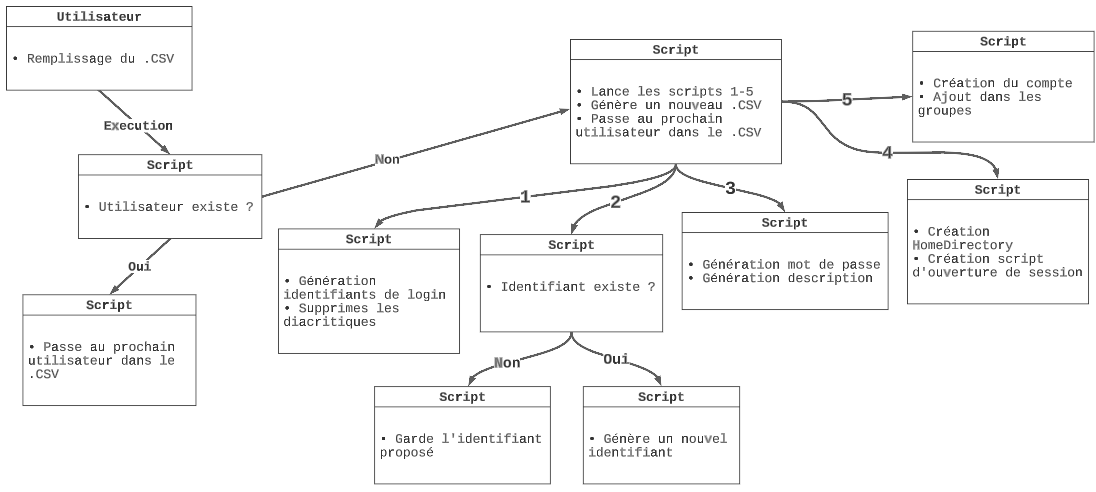


Figure 3 Schéma du processus de création des comptes

### Modification des comptes ✔

Voici comment la modification des comptes va se dérouler.

* La première étape consiste au remplissage des données des apprentis dans un fichier CSV. Ces données sont par exemple : Prénom, Nom, Classe…

Une fois ce fichier remplis, l’utilisateur exécute le script de modification des comptes.

* Une fois le script exécuté, toutes les données du fichier CSV vont être parcourues apprenti par apprenti. Le script va tester si le compte existe. Si c’est le cas, la phase de modification du compte commence. Dans le cas contraire, le script passe à l’apprenti suivant.
* Pour la modification, le script va commencer par comparer les nouvelles informations fournies dans le CSV, et les comparer avec les informations actuelles des comptes. Toutes les informations qui sont différentes sont modifiés dans le compte. Ensuite un nouveau fichier CSV est généré avec les modifications effectuées.

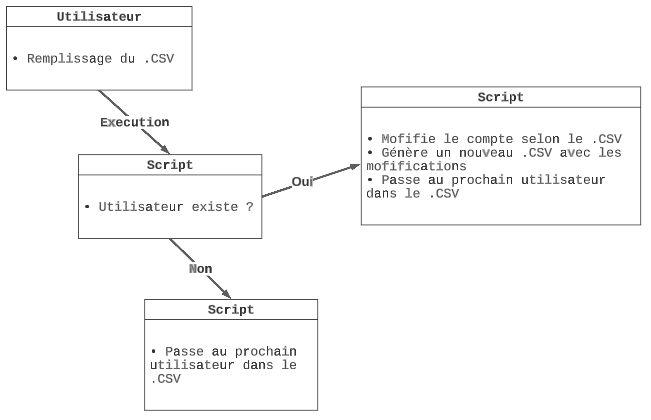


Figure 4 Schéma du processus de modification des comptes

### Suppression des comptes

Voici comment la suppression des comptes va se dérouler.

* La première étape consiste au remplissage des données des apprentis dans un fichier CSV. Ces données sont par exemple : Prénom, Nom, Classe…

Une fois ce fichier remplis, l’utilisateur exécute le script de suppression des comptes.

* Une fois le script exécuté, toutes les données du fichier CSV vont être parcourues apprenti par apprenti. Le script va tester si le compte existe. Si c’est le cas, la phase de suppression du compte commence. Dans le cas contraire, le script passe à l’apprenti suivant.
* Pour la suppression, le script va commencer par vérifier si le compte doit être désactivé ou bien supprimé. S’il doit être désactivé, le compte est simplement désactivé. Dans le cas d’une suppression, le script vérifie si le compte à été désactivé au préalable, et ensuite supprime le répertoire personnel puis enfin supprime le compte. Si le compte n’est pas désactivé, le script passe à l’utilisateur suivant. Ensuite un nouveau fichier CSV est généré avec les actions effectuées.

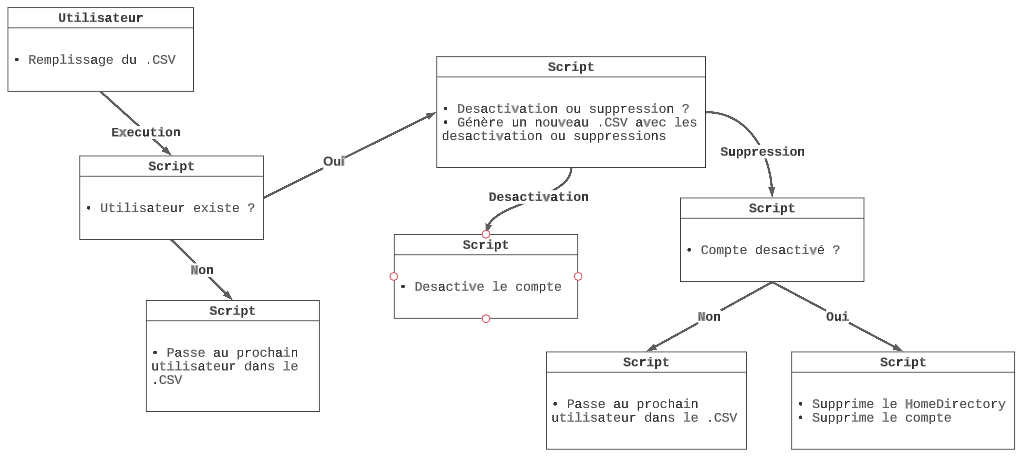


Figure 5 Schéma du processus de suppression des comptes

# Réalisation

## Dossier de réalisation

### Environnement ✔

Pour ce projet, j’avais à disposition un ordinateur standard de l’ETML. Les spécifications sont les suivantes :

* Windows 10 Education 21H2 64 Bits
* Intel i7-11700 2.50Ghz
* RAM 32 Gb 3200 MHz

L’environnement logiciel est le suivant :

* Visual Studio Code, Version 1.78.2
* GitHub Desktop, Version 3.2.3
* Oracle VirtualBox, Version 6.1
* Machine virtuelle, Version Windows Server 2022
* Machine virtuelle, Version Windows 10
* PowerShell, Version 5.1

Voici un schéma de l’environnement réseau utilisé durant ce projet.

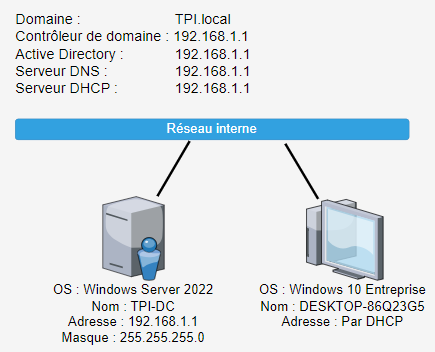


Figure 6 Schéma réseau de l'environnement

### Version du projet ✔

Ce projet est en version 1.0. Car je suis parti de 0 pour le réaliser.

### Normes de codage et de nommage ✔

Voici les normes de codage et de nommage utilisées tout au long de ce projet.

#### Variables et paramètres

Les noms de variables dans les scripts sont définis avec la norme "Lower Camel Case". C’est-à-dire que le nom de la variable débute par une lettre minuscule et les mots suivants commencent par une majuscule. Les noms de variables sont en anglais.



Figure 7 Format de nommage de variables

Les noms de paramètres dans les scripts sont définis avec la norme "Upper Camel Case". C’est-à-dire que le nom de la variable débute par une lettre majuscule et les mots suivants commencent aussi par une majuscule. De plus, le type du paramètre est forcé. Les noms de paramètres sont en anglais.



Figure 8 Format de nommage de paramètres

#### Constantes

Les noms de constantes dans les scripts sont définis sous le format suivant. Tous les mots de la constante sont en majuscule et séparés par un tiret bas. Les noms de constantes sont en anglais.



Figure 9 Format de nommage de constantes

#### Nom des scripts et tests unitaires

Les noms de scripts sont définis par les recommandations Microsoft de nommage des fonctions et scripts PowerShell. Cela consiste à choisir un "verbe" suivi d’un tiret et d’un nom. Le verbe doit faire partie de la liste approuvée par Microsoft et le nom doit spécifier ce qui va être affecté par le script. Les noms de scripts sont en anglais. Ces recommandations sont accessibles en suivant le lien ci-dessous :

<https://learn.microsoft.com/en-us/powershell/scripting/developer/cmdlet/approved-verbs-for-windows-powershell-commands?view=powershell-7.3>

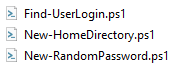


Figure 10 Format de nommage des scripts

Les noms des tests unitaires sont définis par les recommandations du Framework "Pester". Cela consiste à ajouter le mot ".test" entre le nom du script et l’extension du fichier. Ces recommandations sont accessibles en suivant le lien ci-dessous : <https://pester.dev/docs/usage/file-placement-and-naming>

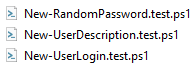


Figure 11 Format de nommage des tests unitaires

#### Entêtes et commentaires

Les entêtes des scripts doivent être en anglais et contenir les informations suivantes :

1. Nom du script, auteur et date de création.
2. Date des modifications, auteur et raison de la modification.
3. Le titre du script.
4. Une description plus détaillée que le titre.
5. Une description de chaque paramètre.
6. Ce que le script retourne.
7. Un exemple de l’utilisation du script.
8. Les liens utilisés pour créer le script.



Figure 12 Format d'entête de scripts

Les commentaires dans les scripts sont pertinents et écrits en anglais.

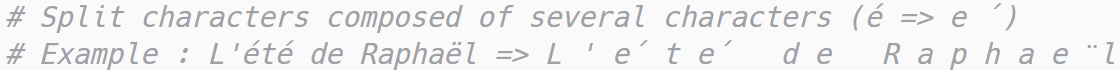


Figure 13 Format de commentaires

### Création des comptes ✔

Le processus de création des comptes utilisateur commence tout d’abord par le remplissage d’un fichier CSV avec les données nécessaires. Ce fichier contient des informations telles que le prénom, le nom, l'adresse e-mail, la classe, la profession et les groupes dans lesquelles mettre l’utilisateur.

Ensuite, le script de création des comptes est exécuté. Pour chaque compte, le script vérifie les données du fichier CSV et effectue plusieurs opérations pour créer les comptes utilisateurs. Génération des identifiants de login et des mots de passe, génération de description, scripts de connexion, création des répertoire personnels et ajout dans les bons groupes. De plus, toutes les actions effectuées sont enregistrées dans un fichier CSV.

Dans cette section, nous détaillerons les différentes étapes du processus de création des comptes utilisateur, depuis le remplissage du fichier CSV jusqu'à la génération du fichier de journalisation.

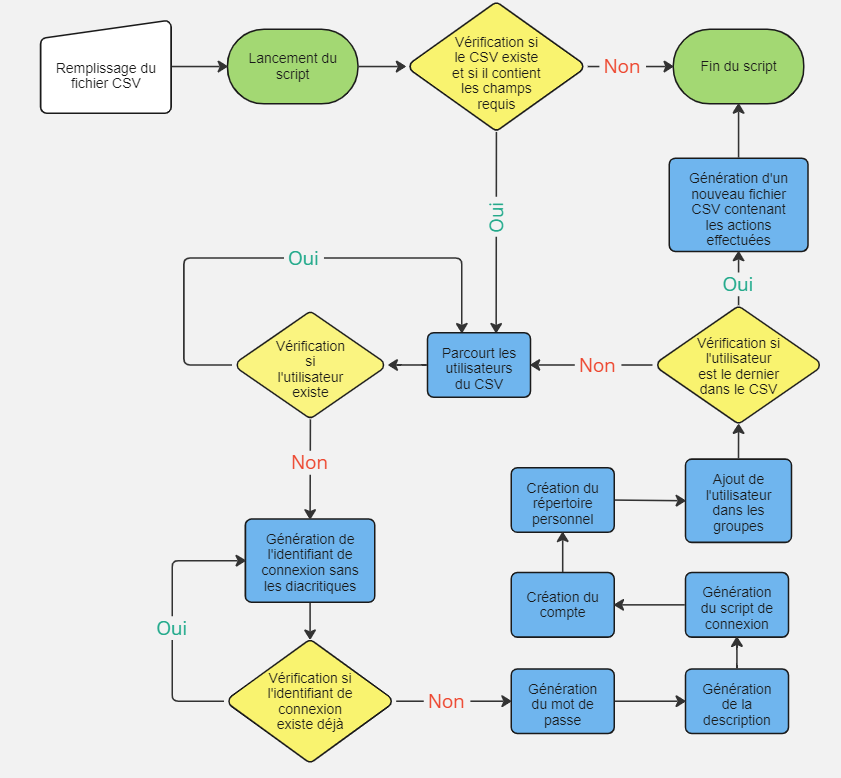


Figure 14 Diagramme de flux de la création des comptes

#### Remplissage du fichier CSV

Afin de pouvoir créer les comptes utilisateur, la première étape consiste à remplir un fichier CSV avec les données des futur comptes. Les données nécessaires à la création des comptes sont les suivantes :

* **Prénom**
* **Nom**
* **E-mail**
* **Classe**
* **Profession**
* **OptionsAD**

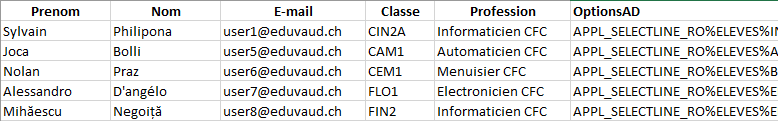


Figure 15 Exemple de données dans le CSV

Le champ "OptionsAD" doit contenir la liste des groupes AD à ajouter à l’utilisateur. Chaque groupe dans ce champ est séparé par un "%". Lors de la création des comptes la liste des groupes est obtenue en divisant la chaine de caractères contenant les groupes aux endroits où il y’a le caractère "%". De plus, le préfix "GUS\_ETML\_" est ajouté au nom du groupe.



Figure 16 Division des groupes avec le caractère "%"

#### Exécution du script

Une fois le fichier CSV remplis avec les informations des comptes à créer, l’utilisateur peut exécuter le script. Lors de l’exécution ce celui-ci, il faut renseigner au script les paramètres suivants :

* **UsersCSV :**

Ce paramètre renseigne le chemin du fichier CSV contenant les données des comptes à créer. Ce paramètre doit obligatoirement être renseigné.

* **ActionsCSV :**

Ce paramètre renseigne le chemin du fichier CSV qui sera crée avec toutes les actions effectuées et les informations des nouveaux comptes (identifiant, mot de passe…). Ce paramètre doit obligatoirement être renseigné.

* **OUPath :**

Ce paramètre renseigne l’unité d’organisation (OU) où les comptes utilisateur vont être créés. Ce paramètre doit obligatoirement être renseigné.

* **CSVDelimiter**

Ce paramètre renseigne le caractère délimiteur des données du fichier CSV contenant les données des comptes à créer. Par défaut celui-ci est une virgule. Cependant certains fichiers ont en un autre comme un point-virgule par exemple. Ce paramètre est facultatif.

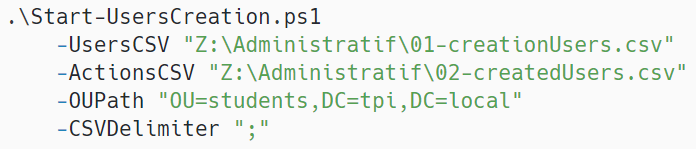


Figure 17 Exemple d'exécution du script de création avec des paramètres

#### Vérification des données du fichier CSV

Une fois le script de création exécuté, le fichier CSV et ses données sont vérifiés pour garantir le bon fonctionnement du programme. Les tests effectués sont les suivants :

1. **Vérification si le fichier existe et qu’il est dans le bon format**

Le chemin du fichier est testé afin de voir s’il existe. De plus le script vérifie si l’extension de fichier est bien ".csv". Si l’un des deux tests ne passe pas, un message d’erreur est affiché à l’utilisateur et l’exécution du script s’arrête.

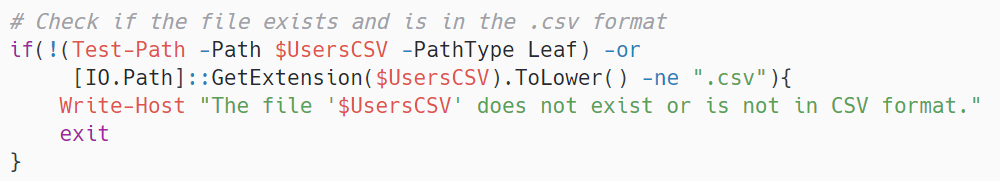


Figure 18 Vérification existence et format du CSV

1. **Vérification des en-têtes du fichier**

Le script teste si tous les champs d’entêtes de données nécessaires sont présents dans le fichier CSV. (Prénom, nom, E-mail, Classe…).

Pour ce faire, le script parcourt tous les champs nécessaires un par un, et vérifie si ce champ est présent dans l’entête. Si un des champs n’est pas présent dans l’entête, un message d’erreur est affiché à l’utilisateur et l’exécution du script s’arrête.

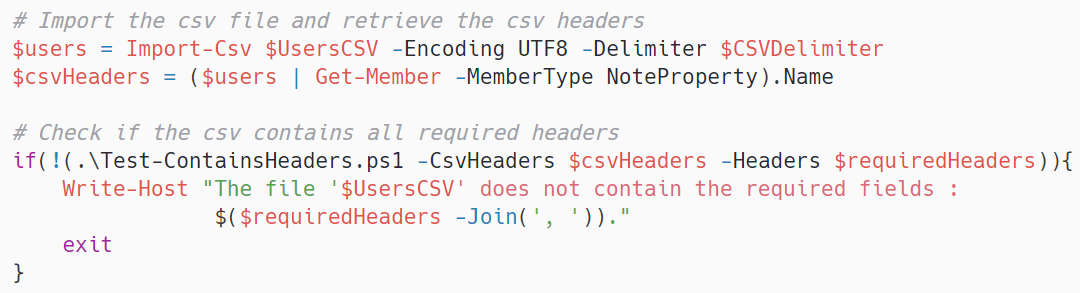


Figure 19 Vérification des en-têtes du fichier

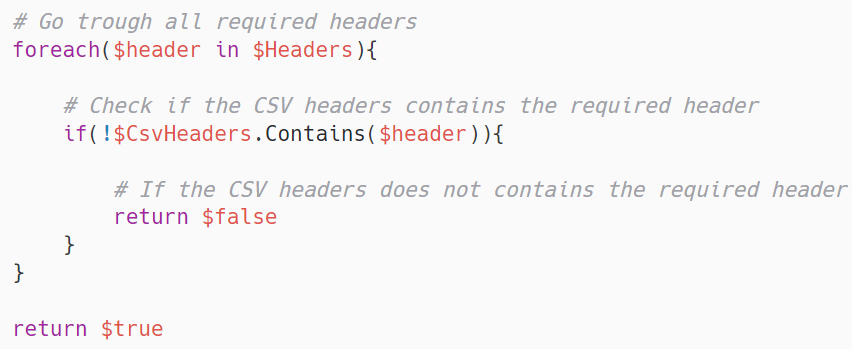


Figure 20 Fonction de vérification des en-têtes du fichier

#### Création des utilisateurs

Une fois ces vérifications effectuées, le script va parcourir tous les utilisateurs du fichier CSV un par un, puis les créer. Pour ce faire le script va effectuer les étapes décrites ci-dessous.

##### Vérification si l’utilisateur existe

Le script vérifie si l’utilisateur à créer existe déjà dans l’annuaire. Pour ce faire, une recherche sur le nom et le prénom de l’utilisateur est effectuée. Si aucune donnée n’est retournée, cela veut dire que le compte n’existe pas. Dans le cas contraire, l’utilisateur est ignoré et le script passe au prochain.



Figure 21 Recherche d'un utilisateur par son nom et prénom

##### Génération de l’identifiant de connexion

Le nom de l’identifiant de connexion doit est généré sous le format suivant :

Les trois premiers caractères du prénom ainsi que les (maximum) 8 premiers caractères du nom de famille. De plus, cet identifiant ne doit pas contenir de caractères spéciaux dit "Diacritiques". Pour ce faire, le script procède en plusieurs étapes.

1. **Suppression des caractères diacritiques**

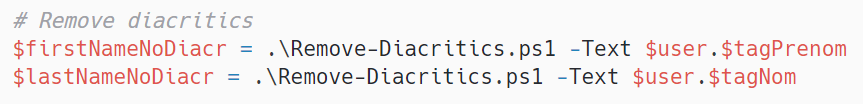


Figure 22 Suppression des caractères diacritiques

Les caractères diacritiques sont supprimés sur les champs nom et prénom de l’utilisateur. Pour ce faire, la forme de normalisation "FormD" (Canonical Decomposition) est utilisée. Cette forme de normalisation Unicode spécifie que les caractères doivent êtres décomposés en utilisant la suite de caractères la plus basique possible. Par exemple le caractère "é" sous la forme "FormD" devient "e ´".

Ensuite, toute la chaine de caractère normalisée est parcourue caractère par caractère. Chacun est testé s’il est de type "nonspacing". S’il ne l’est pas le caractère est ajouté à la chaine finale. Le type de caractère "nonspacing" indique qu’un caractère de ce type ne prend pas d’espace visuel. Il s’agit notamment de caractères accentuant qui modifient un caractère de base. Par exemple le caractère " ´ " composé avec un "e" va l’accentuer et donner "é" sans prendre d’espace en plus.

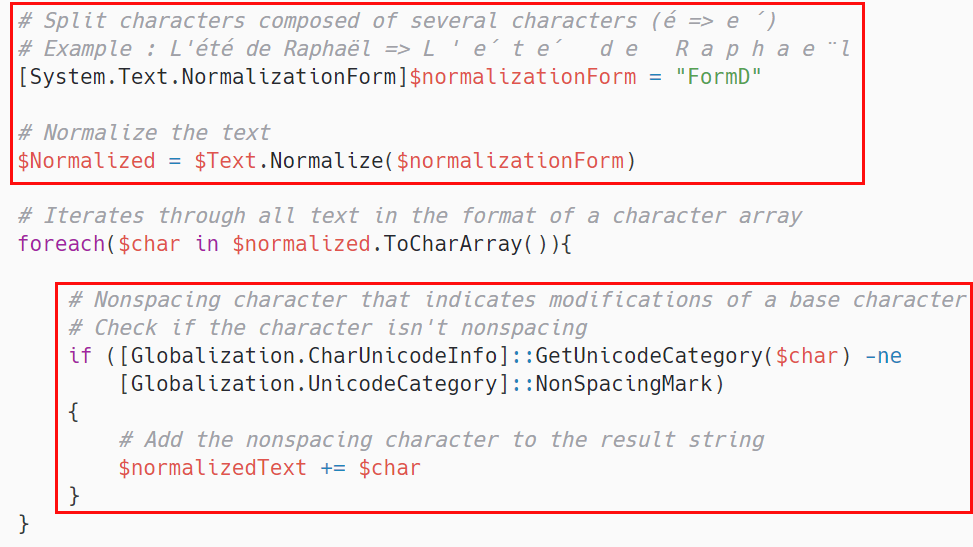


Figure 23 Fonction de suppression des caractères diacritiques

1. **Génération de l’identifiant de connexion**

L’identifiant de connexion est généré en prenant les trois premiers caractères du prénom ainsi que les (maximum) 8 premiers caractères du nom de famille.

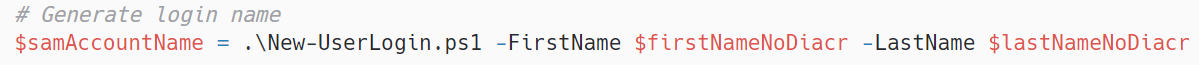


Figure 24 Génération de l’identifiant de connexion

1. **Régénération si l’identifiant de connexion existe déjà**

Un identifiant de connexion peut se trouver être en doublon avec un autre notamment dans le cas où un deux utilisateurs aurait un prénom proche et le même nom de famille. Par exemple :

Max Rochat => maxrochat

Maximilien Rochat => maxrochat

Maxence Rochat => maxrochat

Si l’identifiant de connexion généré existe déjà, un autre est généré en rajoutant un chiffre à la fin. Ce chiffre est incrémenté tant que le nouvel identifiant existe déjà. Si nous reprenons l’exemple précédent, avec cette régénération cela donnerait :

Max Rochat => maxrochat

Maximilien Rochat => maxrochat1

Maxence Rochat => maxrochat2

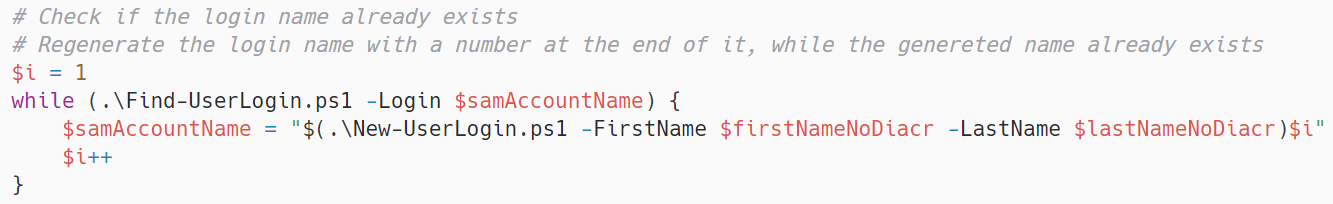


Figure 25 Régénération si l’identifiant de connexion existe déjà

##### Génération du mot de passe

Une fois la génération de l’identifiant terminée, il faut créer un mot de passe aléatoire pour l’utilisateur. Pour ce faire, le script procède en trois étapes.

1. **Liste des caractères utilisables**

Lister les caractères de cette manière permet d’effectuer plus simplement des actions plus spécifiques lors de la génération comme par exemple l’exclusion de caractères similaires ou bien forcer les caractères spéciaux.

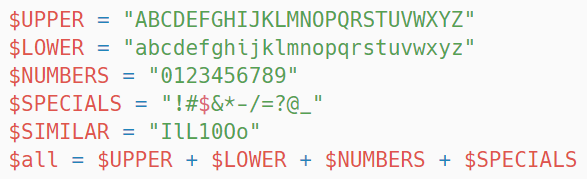


Figure 26 Liste des caractères utilisables

1. **Génération aléatoire d’un caractère**

Un nombre aléatoire entre 0 et la longueur de la liste contenant tous les caractères utilisables mois 1 est généré. Ce nombre représente un index dans cette liste. En récupérant le caractère a cet index le script obtient un caractère aléatoire. Cette étape ainsi que la suivant sont effectuées autant de fois que la longueur de mot de passe requise.

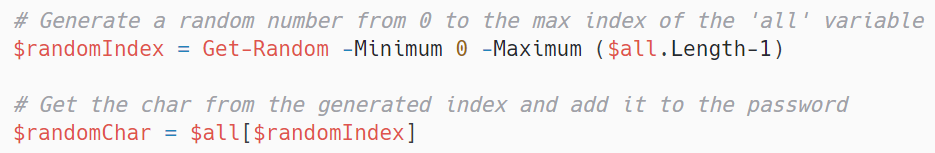


Figure 27 Génération aléatoire d’un caractère

1. **Exclusion des caractères similaires**

Si lors de l’appel de la fonction le paramètre "ExcludeSimilar" a été spécifié, le mot de passe ne doit pas contenir de caractères similaires qui pourraient êtres confondus par l’utilisateur. Ces caractères sont par exemple les "i" majuscules et les "L" minuscules.

Si le caractère généré se trouve être dans la liste des similaires, le script va simplement passer à la prochaine itération sans ajouter le caractère généré dans le mot de passe final.

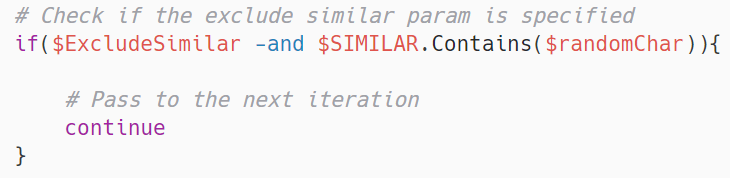


Figure 28 Exclusion des caractères similaires

##### Génération de la description

Chaque utilisateur doit avoir une description qui mentionne la classe dans laquelle il est. Cette description est générée en remplaçant le mot "[classe]" dans la description par défaut par l’actuelle classe de l’utilisateur.

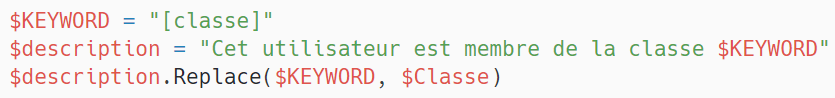


Figure 29 Génération de la description

##### Génération du script de connexion

A l’ETML, Il existe des scripts de connexion propres à chaque sections / professions. Selon la profession de l’utilisateur, le bon script de connexion doit lui être attribué. Ceci est fait en deux étapes.

1. **Dictionnaire des professions**

Un dictionnaire contenant les différentes professions ainsi que leurs scripts respectifs est déclaré. De cette manière il est simple d’attribuer un script selon la profession de l’utilisateur.

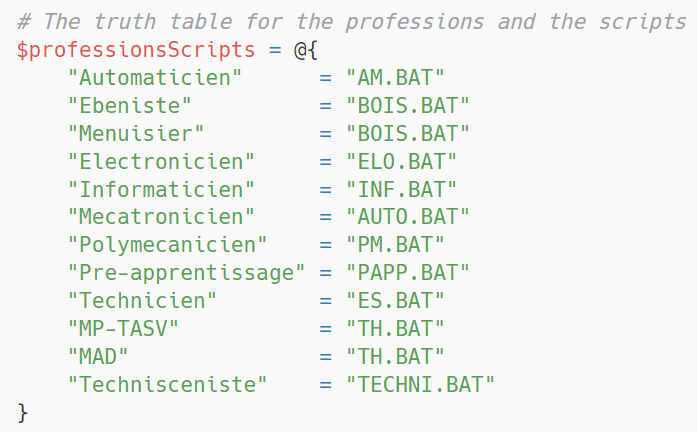


Figure 30 Dictionnaire des professions

1. **Attribution du script**

Tous les champs du dictionnaire sont parcourus afin de trouver la profession de l’utilisateur dans le dictionnaire. Dès que celle-ci est trouvée, le script correspondant est retourné. Dans le cas où la profession ne serait pas retrouvée, la valeur null est retournée.

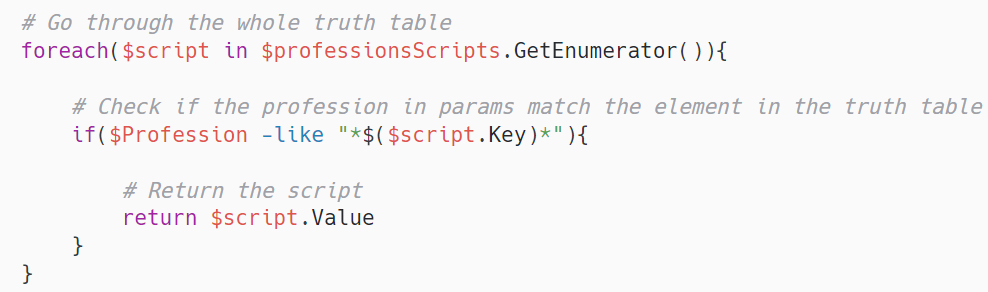


Figure 31 Attribution du script

##### Création du compte

Une fois toutes les étapes précédentes effectuées, le compte utilisateur est prêt à être crée et activé. Pour ce faire le script procède en deux étapes.

1. **Définition des données utilisateur**

Un dictionnaire est créé contenant toutes les données nécessaires à la création de l’utilisateur.

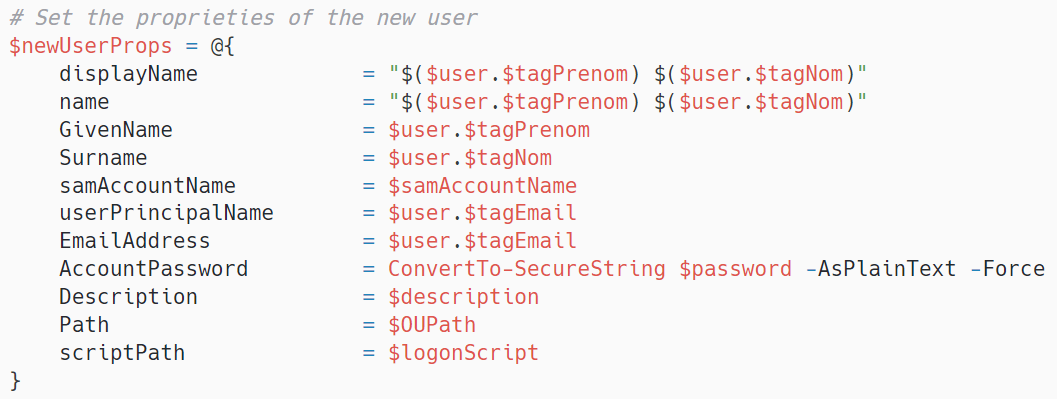


Figure 32 Définition des données utilisateur

1. **Création et activation du compte**

Le compte est créé avec les données définies dans le dictionnaire. Ces données sont fournies avec l’opérateur "@" qui permet de passer les propriétés du dictionnaire en tant qu’arguments. Ensuite, la propriété

"ChangePasswordAtLogon" est définie sur le compte utilisateur forçant le changement de mot de passe lors de la première connexion. Cette propriété doit être définie après la création du compte en raison d’un bug dans la fonction Microsoft de création de comptes AD. Pour finir le compte est activé.

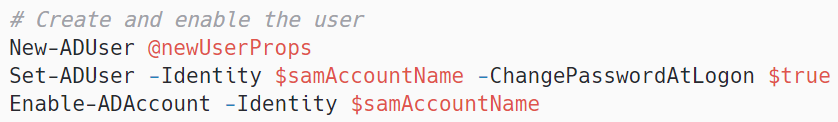


Figure 33 Création et activation du compte

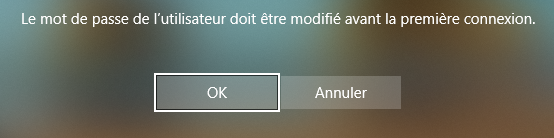


Figure 34 L'utilisateur doit changer de mot de passe

##### Création du répertoire personnel

Afin de créer le répertoire personnel de l’utilisateur, le script va passer par trois étapes.

1. **Création du dossier principal**

La première étape est de vérifier si le dossier principal contenant tous les dossiers personnels des utilisateurs existe. Si celui-ci n’existe pas, il est créé.

L’héritage des droits est ensuite désactivé et les droits d’accès NTFS sont accordés en contrôle total au système et au propriétaire du dossier.

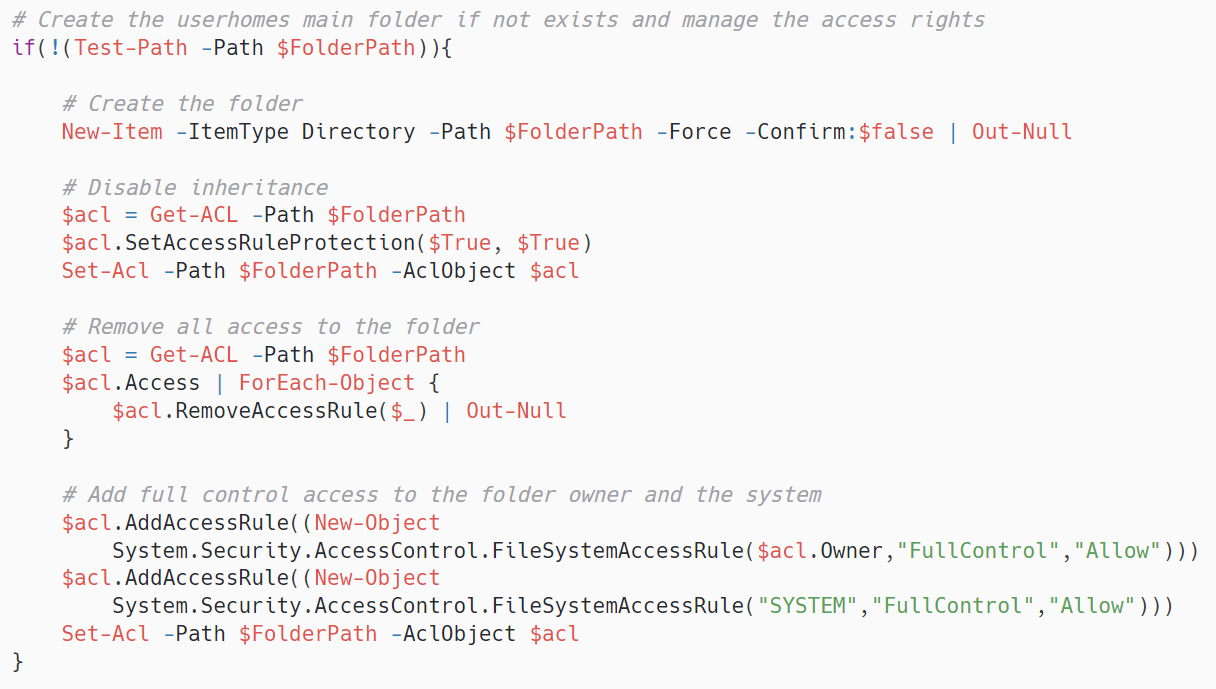


Figure 35 Création du dossier principal

1. **Création du répertoire personnel**

Le script vérifie si le dossier personnel de l’utilisateur existe déjà. Si c’est le cas l’exécution continue sans recréer le dossier. Dans le cas contraire, le dossier personnel est créé. Ensuite, les droits d’accès NTFS sont accordés en uniquement en modification. De cette manière, l’utilisateur ne pourra pas modifier les droits d’accès à son dossier. Pour finir, le chemin d’accès réseau au dossier est retourné sous le format suivant :

"\\SRV-NAME\SHARE-NAME\username".

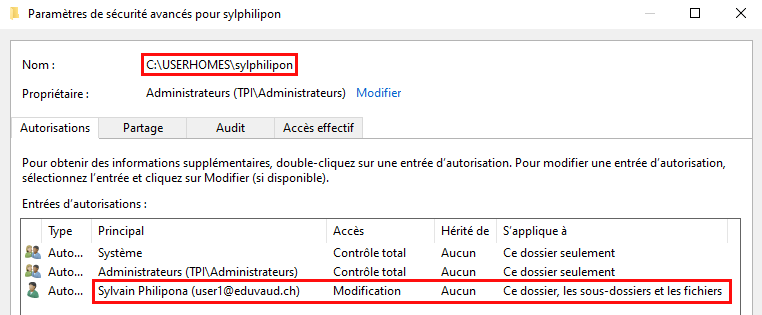


Figure 36 Droits d'accès au dossier

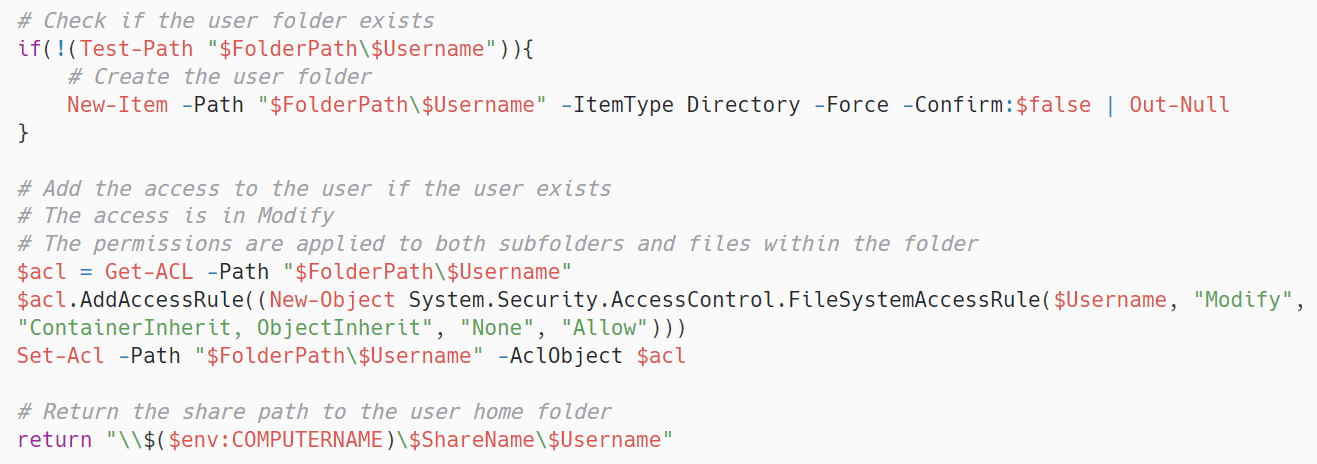


Figure 37 Création du répertoire personnel

1. **Ajout du dossier dans le compte utilisateur**

Le dossier personnel est ensuite ajouté dans les paramètres du compte utilisateur. Le dossier est associé à une lettre.

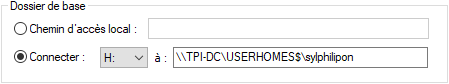


Figure 38 Chemin du dossier personnel

Cette lettre sera associée au lecteur réseau lors de l’accès au dossier personnel.



Figure 39 Lecteur réseau

##### Ajout dans les groupes

Une fois répertoire personnel de l’utilisateur créé, il faut maintenant l’ajouter dans les groupes. Le script reçoit en entrée la liste des groupes dans une chaine de caractères séparés par un caractère "%".



Figure 40 Exemples de chaine ce caractères de groupes

La liste des groupes est divisée aux endroits ou il y’a un "%", puis l’utilisateur est ajouté dans chaque groupe. De plus, le préfix "GUS\_ETML\_" est ajouté au nom du groupe.

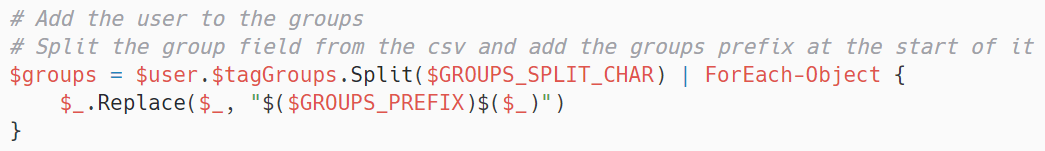


Figure 41 Obtention de la liste des groupes



Figure 42 Ajout dans les groupes

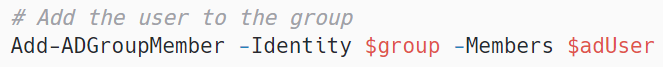


Figure 43 Fonction d'ajout dans les groupes

#### Génération du CSV

Afin de journaliser ce qui se passe lors de la création des comptes, un fichier CSV contenant ce qui se passe est généré. Tout d’abord un dictionnaire contenant les titres des informations à journaliser est déclaré. Toutes les valeurs sont vides sauf l’utilisateur.

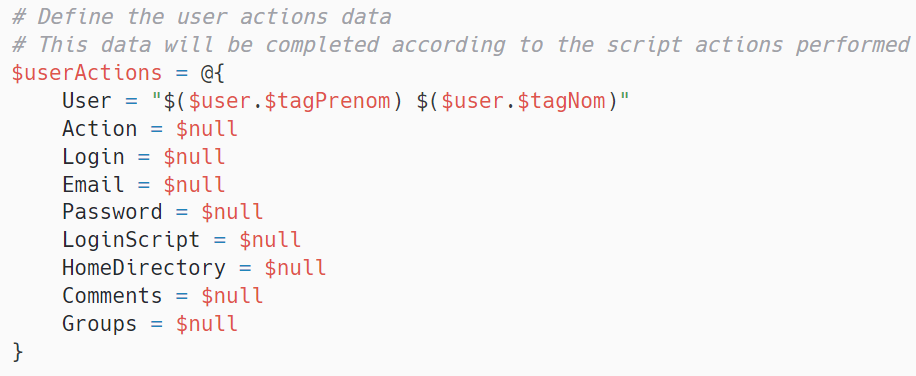


Figure 44 Déclaration du dictionnaire

Selon ce qui se passe durant la création du compte, les informations de journalisation sont remplies.

Ici, l’utilisateur existe déjà. L’action est définie sur "Failed" et le commentaire ajouté.



Figure 45 Erreur utilisateur existe déjà

Ici, il y’a eu une erreur lors de la création de l’utilisateur. L’action est définie sur "Failed", le commentaire ajouté ainsi que l’identifiant de connexion et le mot de passe.

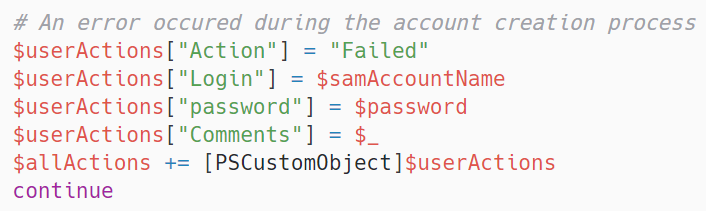


Figure 46 Erreur lors de la création de l'utilisateur

Ici, le compte a été créé avec succès. Tous les champs sont donc renseignés. De plus, le dictionnaire est ajouté à la liste contenant toutes les actions des utilisateurs créés.

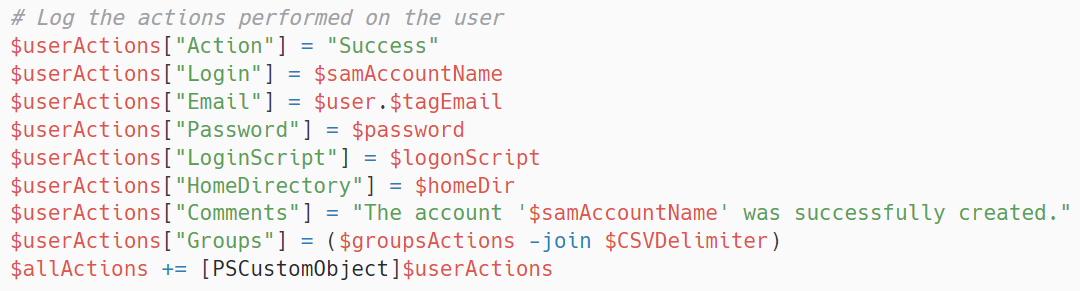


Figure 47 Compte créé avec succès

Une fois tous les comptes créés, le fichier csv est exporté avec les données des utilisateurs.

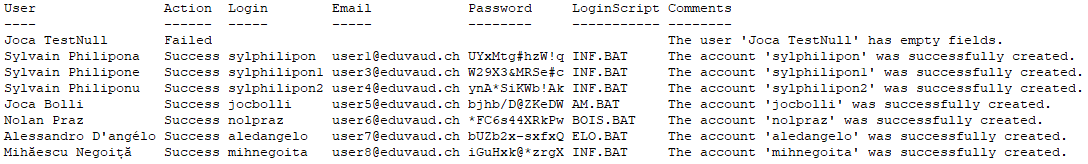


Figure 48 Données CSV exportés

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Evitez d’inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez Fue cette partie…*

### Modification des comptes ✔

Le processus de modification des comptes utilisateur commence tout d’abord par le remplissage d’un fichier CSV avec les données nécessaires. Ce fichier contient des informations telles que le nom d’ouverture de session, l’e-mail, la classe, la profession et les groupes dans lesquelles mettre l’utilisateur.

Ensuite, le script de modification des comptes est exécuté. Pour chaque compte, le script vérifie les données du fichier CSV et effectue plusieurs opérations afin de modifier les comptes utilisateurs. Génération des descriptions, scripts de connexion, modification de l’utilisateur et ajout dans les bons groupes. De plus, toutes les actions effectuées sont enregistrées dans un fichier CSV.

Dans cette section, nous détaillerons les différentes étapes du processus de modification des comptes utilisateur, depuis le remplissage du fichier CSV jusqu'à la génération du fichier de journalisation.

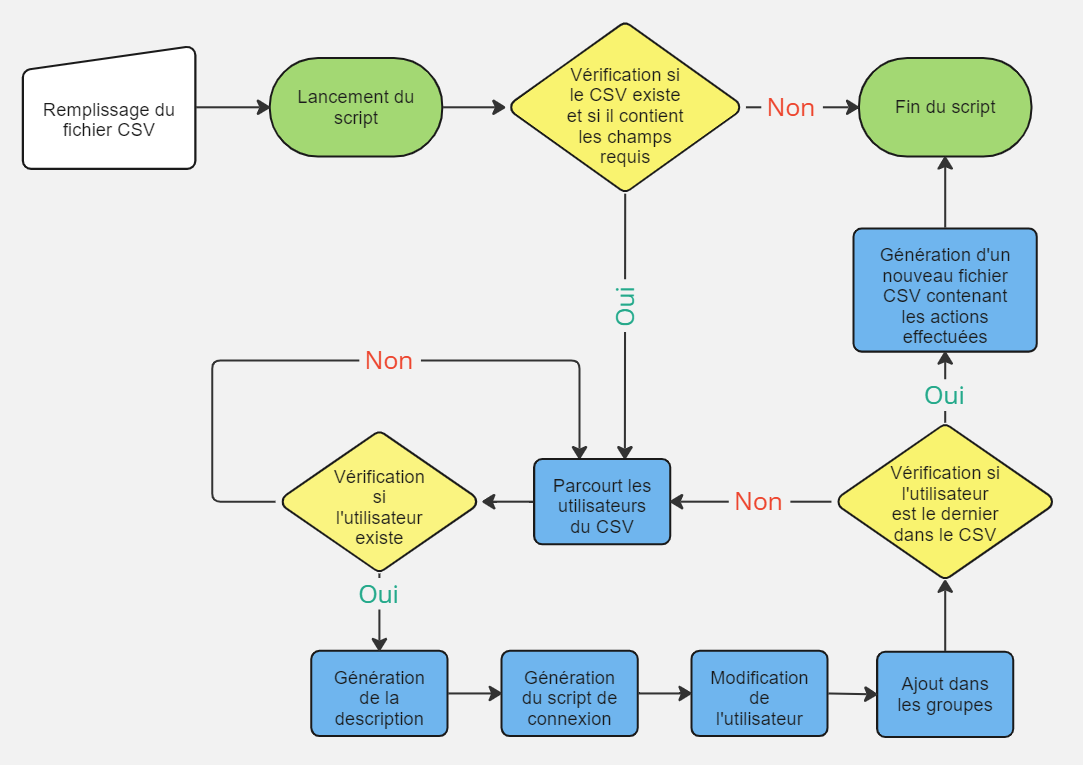


Figure 49 Diagramme de flux de la modification des comptes

#### Remplissage du fichier CSV

Afin de pouvoir modifier les comptes utilisateur, la première étape consiste à remplir un fichier CSV avec les nouvelles données des comptes. Les données nécessaires à la modification des comptes sont les suivantes :

* **Identifiant de connexion**
* **Nouvel E-mail**
* **Nouvelle Classe**
* **Nouvelle Profession**
* **Nouvelles OptionsAD**

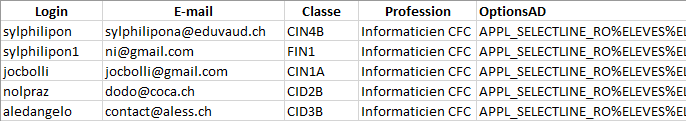


Figure 50 Exemple de données dans le CSV

Le champ "OptionsAD" doit contenir la liste des groupes AD à ajouter à l’utilisateur. Chaque groupe dans ce champ est séparé par un "%". Lors de la création des comptes la liste des groupes est obtenue en divisant la chaine de caractères contenant les groupes aux endroits où il y’a le caractère "%". De plus, le préfix "GUS\_ETML\_" est ajouté au nom du groupe.



Figure 51 Division des groupes avec le caractère "%"

#### Exécution du script

Une fois le fichier CSV remplis avec les informations des comptes à modifier, l’utilisateur peut exécuter le script. Lors de l’exécution ce celui-ci, il faut renseigner au script les paramètres suivants :

* **UsersCSV :**

Ce paramètre renseigne le chemin du fichier CSV contenant les données des comptes à modifier. Ce paramètre doit obligatoirement être renseigné.

* **ActionsCSV :**

Ce paramètre renseigne le chemin du fichier CSV qui sera créé avec toutes les actions effectuées. Ce paramètre doit obligatoirement être renseigné.

* **CSVDelimiter**

Ce paramètre renseigne le caractère délimiteur des données du fichier CSV contenant les données des comptes à modifier. Par défaut celui-ci est une virgule. Cependant certains fichiers ont en un autre comme un point-virgule par exemple. Ce paramètre est facultatif.

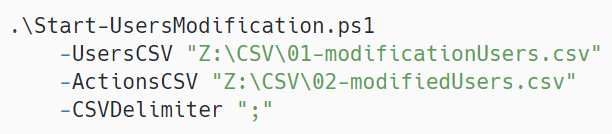


Figure 52 Exemple d'exécution du script de modification avec des paramètres

#### Vérification des données du fichier CSV

Une fois le script de modification exécuté, le fichier CSV et ses données sont vérifiés pour garantir le bon fonctionnement du programme. Les tests effectués sont les suivants :

1. **Vérification si le fichier existe et qu’il est dans le bon format**

Le chemin du fichier est testé afin de voir s’il existe. De plus le script vérifie si l’extension de fichier est bien ".csv". Si l’un des deux tests ne passe pas, un message d’erreur est affiché à l’utilisateur et l’exécution du script s’arrête.

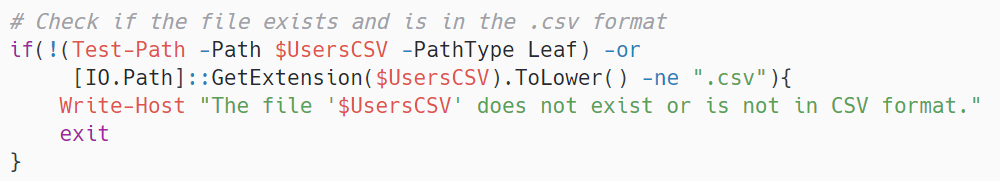


Figure 53 Vérification existence et format du CSV

1. **Vérification des en-têtes du fichier**

Le script teste si tous les champs d’entêtes de données nécessaires sont présents dans le fichier CSV. (Identifiant de connexion, E-mail, Classe…).

Pour ce faire, le script parcourt tous les champs nécessaires un par un, et vérifie si ce champ est présent dans l’entête. Si un des champs n’est pas présent dans l’entête, un message d’erreur est affiché à l’utilisateur et l’exécution du script s’arrête.

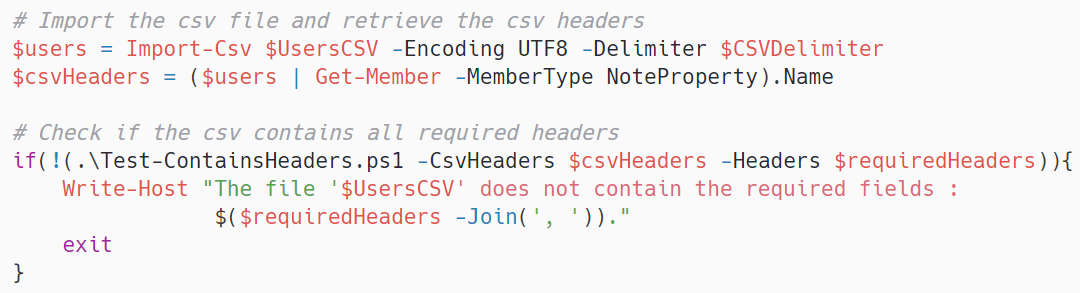


Figure 54 Vérification des en-têtes du fichier

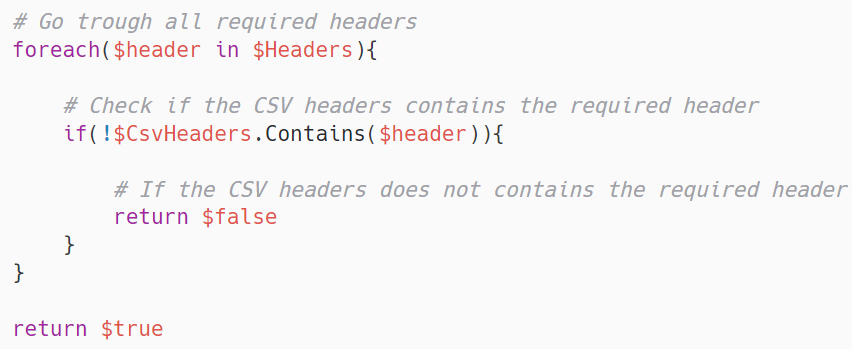


Figure 55 Fonction de vérification des en-têtes du fichier

#### Modification des utilisateurs

Une fois ces vérifications effectuées, le script va parcourir tous les utilisateurs du fichier CSV un par un, puis les modifier. Pour ce faire le script va effectuer les étapes décrites ci-dessous.

##### Vérification si l’utilisateur existe

Le script vérifie si l’utilisateur à modifier existe déjà dans l’annuaire. Pour ce faire, une recherche sur le nom d’ouverture de session de l’utilisateur est effectuée. Si une donnée est retournée, cela veut dire que le compte existe. Dans le cas contraire, l’utilisateur est ignoré et le script passe au prochain.



Figure 56 Recherche d'un utilisateur

##### Génération de la description

Chaque utilisateur doit avoir une description qui mentionne la classe dans laquelle il est. Cette description est générée en remplaçant le mot "[classe]" dans la description par défaut par l’actuelle classe de l’utilisateur.

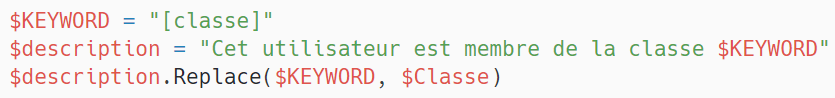


Figure 57 Génération de la description

##### Génération du script de connexion

A l’ETML, Il existe des scripts de connexion propres à chaque sections / professions. Selon la profession de l’utilisateur, le bon script de connexion doit lui être attribué. Ceci est fait en deux étapes.

1. **Dictionnaire des professions**

Un dictionnaire contenant les différentes professions ainsi que leurs scripts respectifs est déclaré. De cette manière il est simple d’attribuer un script selon la profession de l’utilisateur.

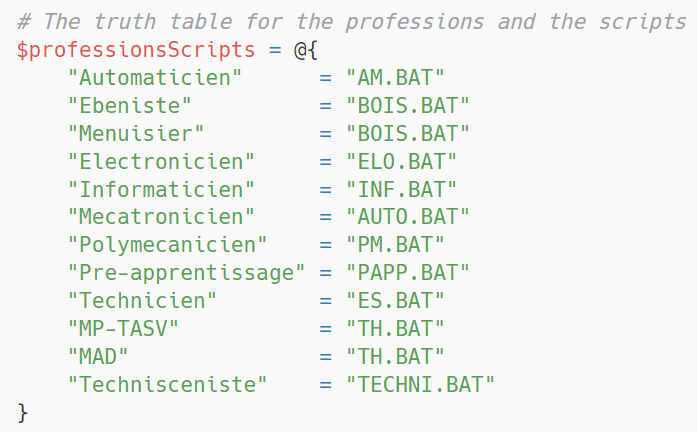


Figure 58 Dictionnaire des professions

1. **Attribution du script**

Tous les champs du dictionnaire sont parcourus afin de trouver la profession de l’utilisateur dans le dictionnaire. Dès que celle-ci est trouvée, le script correspondant est retourné. Dans le cas où la profession ne serait pas retrouvée, la valeur null est retournée.

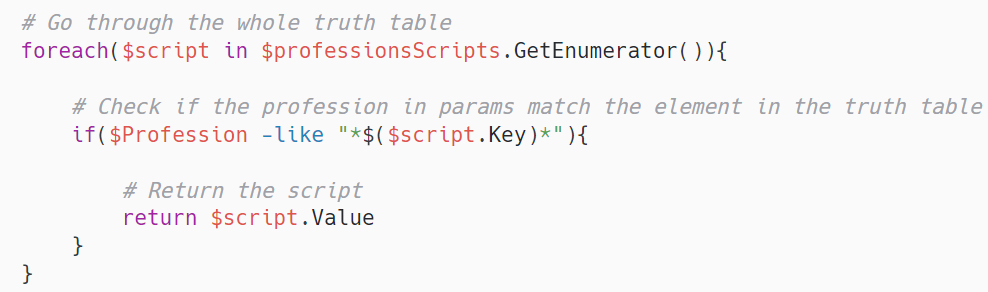


Figure 59 Attribution du script

##### Modification de l’utilisateur

Une fois toutes les étapes précédentes effectuées, le compte utilisateur est prêt à être modifié. Pour ce faire le script procède en deux étapes.

1. **Définition des données utilisateur**

Un dictionnaire est créé contenant toutes les données nécessaires à la modification de l’utilisateur.

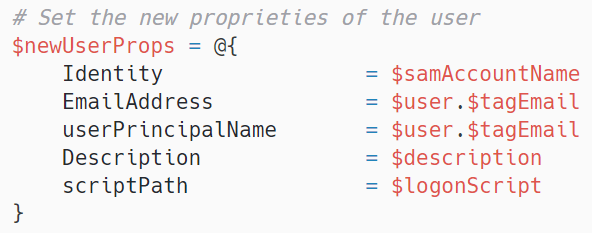


Figure 60 Définition des données utilisateur

1. **Modification du compte**

Le compte est modifié avec les données définies dans le dictionnaire. Ces données sont fournies avec l’opérateur "@" qui permet de passer les propriétés du dictionnaire en tant qu’arguments.

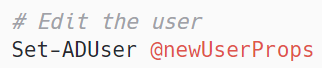


Figure 61 Modification du compte

##### Ajout dans les groupes

Une fois compte utilisateur modifié, il faut maintenant l’ajouter dans les groupes. Le script reçoit en entrée la liste des groupes dans une chaine de caractères séparés par un caractère "%".



Figure 62 Exemples de chaine ce caractères de groupes

Il faut d’abord retirer l’utilisateur de tous ses groupes.

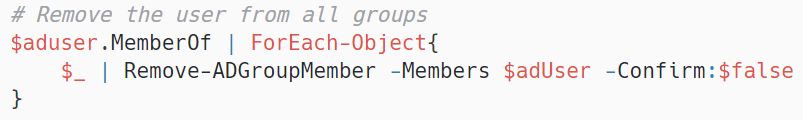


Figure 63 Retirer l'utilisateur des groupes

La liste des groupes est divisée aux endroits où il y’a un "%", puis l’utilisateur est ajouté dans chaque groupe. De plus, le préfix "GUS\_ETML\_" est ajouté au nom du groupe.

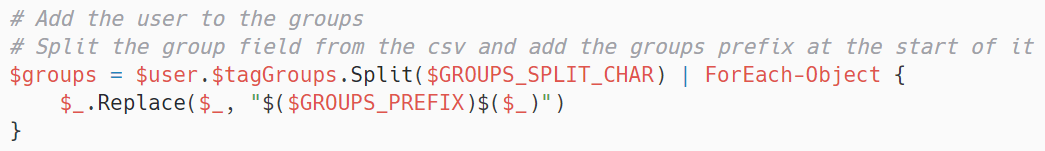


Figure 64 Obtention de la liste des groupes



Figure 65 Ajout dans les groupes

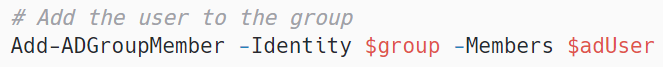


Figure 66 Fonction d'ajout dans les groupes

#### Génération du CSV

Afin de journaliser ce qui se passe lors de la modification des comptes, un fichier CSV contenant ce qui se passe est généré. Tout d’abord un dictionnaire contenant les titres des informations à journaliser est déclaré. Toutes les valeurs sont vides sauf l’utilisateur.

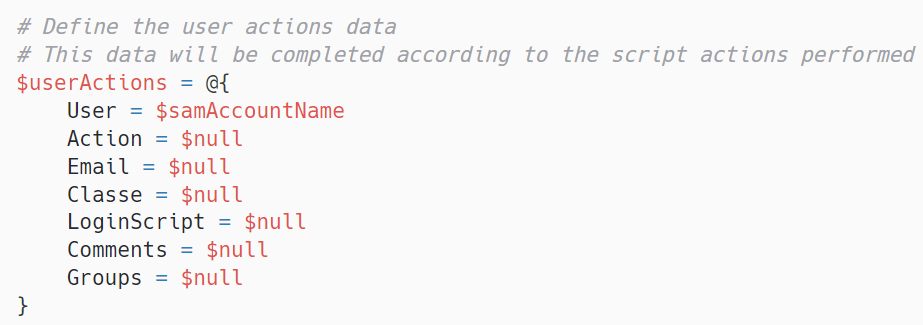


Figure 67 Déclaration du dictionnaire

Selon ce qui se passe durant la modification du compte, les informations de journalisation sont remplies.

Ici, l’utilisateur n’existe pas. L’action est définie sur "Failed" et le commentaire ajouté.



Figure 68 Erreur utilisateur n’existe pas

Ici, il y’a eu une erreur lors de la modification de l’utilisateur. L’action est définie sur "Failed", le commentaire ajouté.

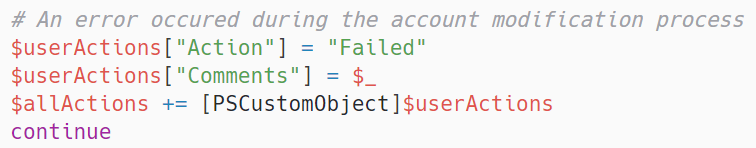


Figure 69 Erreur lors de la modification de l'utilisateur

Ici, le compte a été modifié avec succès. Tous les champs sont donc renseignés. De plus, le dictionnaire est ajouté à la liste contenant toutes les actions des utilisateurs créés.

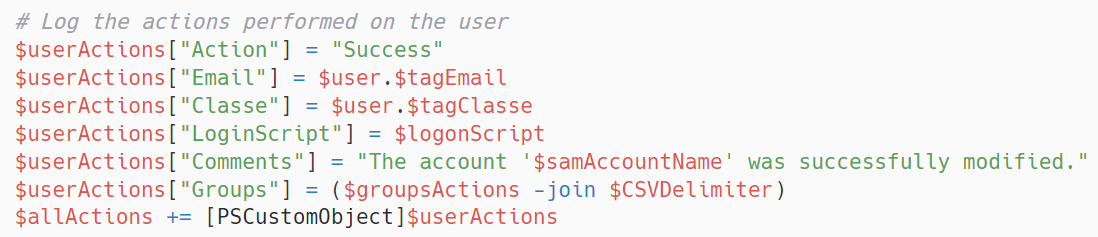


Figure 70 Compte modifié avec succès

Une fois tous les comptes modifiés, le fichier csv est exporté avec les données des utilisateurs.

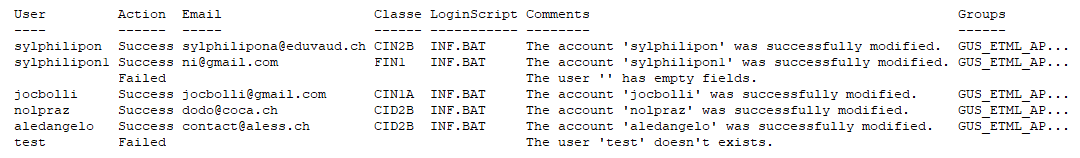


Figure 71 Données CSV exportés

### Suppression des comptes ✔

Le processus de suppression des comptes utilisateur commence tout d’abord par le remplissage d’un fichier CSV avec les données nécessaires. Ce fichier contient des informations telles que le nom d’ouverture de session, une indication si le compte doit être supprimé ou bien désactivé.

Ensuite, le script de suppression des comptes est exécuté. Pour chaque compte, le script vérifie les données du fichier CSV et effectue plusieurs opérations afin de supprimer les comptes utilisateurs. Désactivation, suppression du répertoire personnel et suppression du compte utilisateur. De plus, toutes les actions effectuées sont enregistrées dans un fichier CSV.

Dans cette section, nous détaillerons les différentes étapes du processus de suppression des comptes utilisateur, depuis le remplissage du fichier CSV jusqu'à la génération du fichier de journalisation.

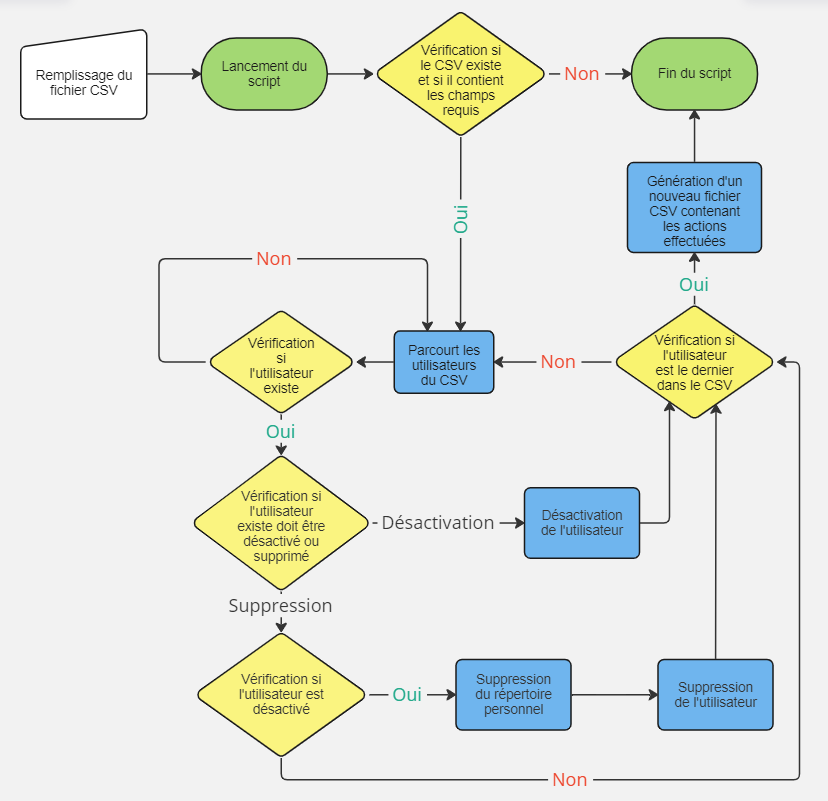


Figure 72 Diagramme de flux de la modification des comptes

#### Remplissage du fichier CSV

Afin de pouvoir modifier les comptes utilisateur, la première étape consiste à remplir un fichier CSV avec les nouvelles données des comptes. Les données nécessaires à la modification des comptes sont les suivantes :

* **Identifiant de connexion**
* **Désactivation**
* **Suppression**

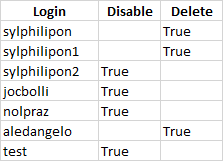


Figure 73 Exemple de données dans le CSV

Les champs "Disable et "Delete" doivent contenir la valeur "True" pour effectuer l’action, ou bien n’importe quelle valeur pour ne pas l’effectuer. Il absolument faut un champ en "True" et l’autre avec une autre valeur. Dans le cas où les deux champs ont la valeur "True", le script prioritise la désactivation. Un compte doit obligatoirement être supprimé avant de pouvoir être supprimé.

#### Exécution du script

Une fois le fichier CSV remplis avec les informations des comptes à supprimer, l’utilisateur peut exécuter le script. Lors de l’exécution ce celui-ci, il faut renseigner au script les paramètres suivants :

* **UsersCSV :**

Ce paramètre renseigne le chemin du fichier CSV contenant les données des comptes à supprimer. Ce paramètre doit obligatoirement être renseigné.

* **ActionsCSV :**

Ce paramètre renseigne le chemin du fichier CSV qui sera créé avec toutes les actions effectuées. Ce paramètre doit obligatoirement être renseigné.

* **CSVDelimiter**

Ce paramètre renseigne le caractère délimiteur des données du fichier CSV contenant les données des comptes à modifier. Par défaut celui-ci est une virgule. Cependant certains fichiers ont en un autre comme un point-virgule par exemple. Ce paramètre est facultatif.

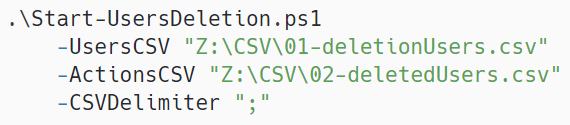


Figure 74 Exemple d'exécution du script de suppression avec des paramètres

#### Vérification des données du fichier CSV

Une fois le script de modification exécuté, le fichier CSV et ses données sont vérifiés pour garantir le bon fonctionnement du programme. Les tests effectués sont les suivants :

1. **Vérification si le fichier existe et qu’il est dans le bon format**

Le chemin du fichier est testé afin de voir s’il existe. De plus le script vérifie si l’extension de fichier est bien ".csv". Si l’un des deux tests ne passe pas, un message d’erreur est affiché à l’utilisateur et l’exécution du script s’arrête.

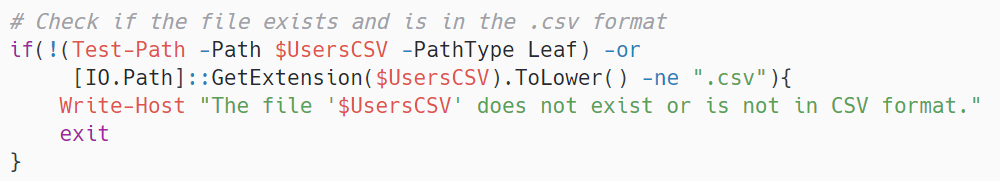


Figure 75 Vérification existence et format du CSV

1. **Vérification des en-têtes du fichier**

Le script teste si tous les champs d’entêtes de données nécessaires sont présents dans le fichier CSV. (Identifiant de connexion, désactivation…).

Pour ce faire, le script parcourt tous les champs nécessaires un par un, et vérifie si ce champ est présent dans l’entête. Si un des champs n’est pas présent dans l’entête, un message d’erreur est affiché à l’utilisateur et l’exécution du script s’arrête.

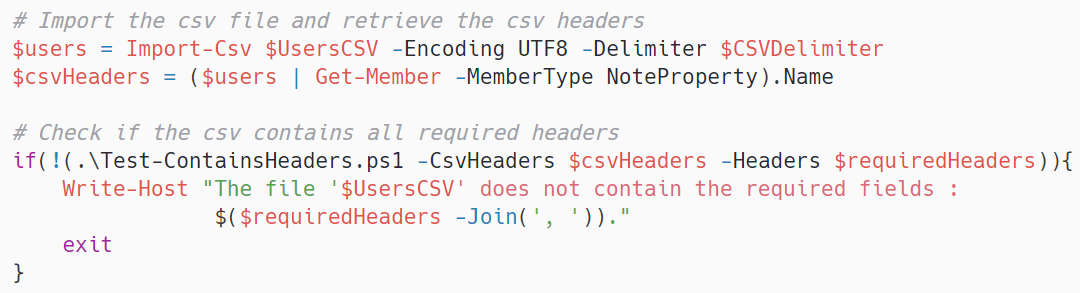


Figure 76 Vérification des en-têtes du fichier

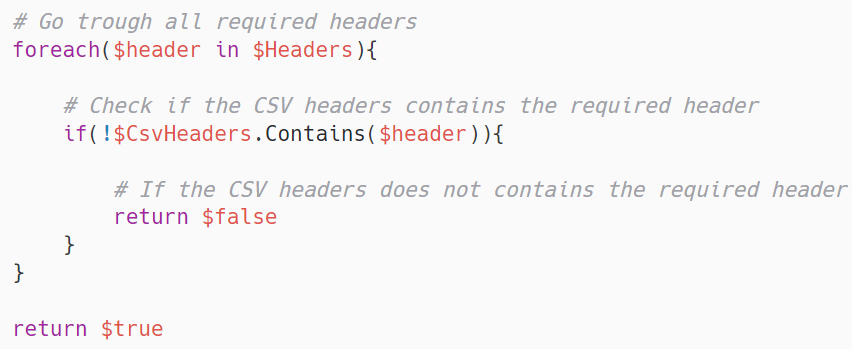


Figure 77 Fonction de vérification des en-têtes du fichier

#### Modification ou suppression des utilisateurs

##### Vérification si l’utilisateur existe

Une fois ces vérifications effectuées, le script va parcourir tous les utilisateurs du fichier CSV un par un. Le script vérifie si l’utilisateur à supprimer existe déjà dans l’annuaire. Pour ce faire, une recherche sur le nom d’ouverture de session de l’utilisateur est effectuée. Si une donnée est retournée, cela veut dire que le compte existe. Dans le cas contraire, l’utilisateur est ignoré et le script passe au prochain.



Figure 78 Recherche d'un utilisateur

##### Désactivation de l’utilisateur

Afin de savoir si le compte doit être désactivé ou bien supprimé, le script va vérifier si la valeur du champ "Disable" est "True". Dans ce cas, le script va simplement désactiver l’utilisateur. Dans le cas contraire le script va passer à l’étape de suppression du compte.



Figure 79 Désactivation d'un compte utilisateur

##### Suppression de l’utilisateur

Le script va tout d’abord vérifier si la valeur du champ "Delete" est "True". Dans ce cas, le processus de suppression du compte va commencer. Dans le cas contraire, le script passe au prochain utilisateur dans le CSV.

###### Vérification si l’utilisateur est désactivé

Le script va tester si l’utilisateur est désactivé dans l’annuaire. Dans ce cas les étapes de suppression continuent. Dans le cas contraire, le script passe au prochain utilisateur dans le CSV.

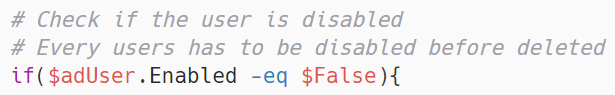


Figure 80 Vérification si l’utilisateur est désactivé

###### Suppression du répertoire personnel

Le script commence par récupérer le chemin du dossier personnel de l’utilisateur, puis supprime récursivement le dossier ainsi que ses sous dossiers et fichier. Ensuite, les champs "HomeDirectory" et "HomeDrive" sont définis avec une valeur nulle.

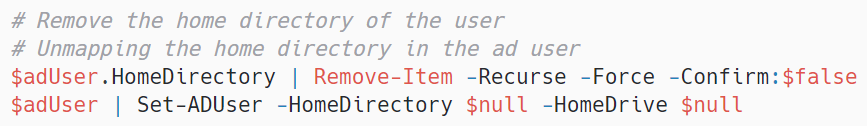


Figure 81 Suppression du répertoire personnel

###### Suppression de l’utilisateur

L’utilisateur est supprimé de l’annuaire.

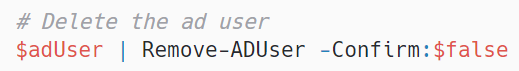


Figure 82 Suppression de l’utilisateur

##### Génération du CSV

Afin de journaliser ce qui se passe lors de la suppression des comptes, un fichier CSV contenant ce qui se passe est généré. Tout d’abord un dictionnaire contenant les titres des informations à journaliser est déclaré. Toutes les valeurs sont vides sauf l’utilisateur.

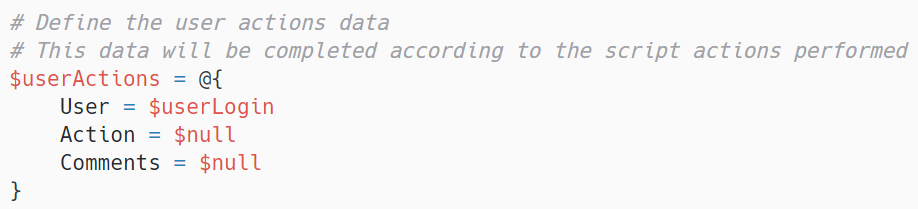


Figure 83 Déclaration du dictionnaire

Selon ce qui se passe durant la suppression du compte, les informations de journalisation sont remplies.

Ici, l’utilisateur n’existe pas. L’action est définie sur "Failed" et le commentaire ajouté.



Figure 84 Erreur utilisateur n’existe pas

Ici, l’utilisateur doit être supprimé mais n’est pas désactivé, L’action est définie sur "Failed" et le commentaire ajouté.

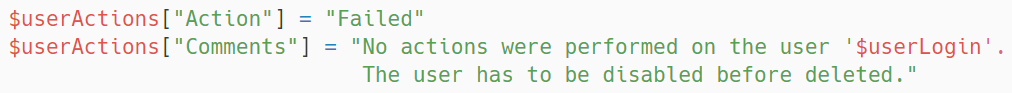


Figure 85 Erreur utilisateur pas désactivé

Ici, il y’a eu un problème avec les données de désactivation et de suppression. Cela peut être causé si aucun des deux champs ne contient la valeur "True".

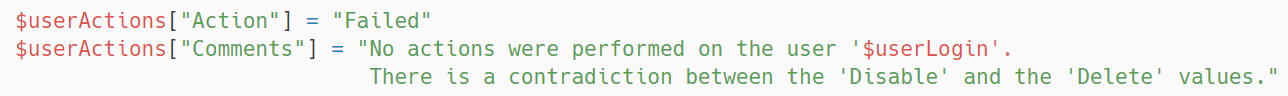


Figure 86 Erreur problèmes avec les champs "Disable" et "Delete"

Ici, l’utilisateur à bien été supprimé, cependant il n’avait pas de répertoire personnel.

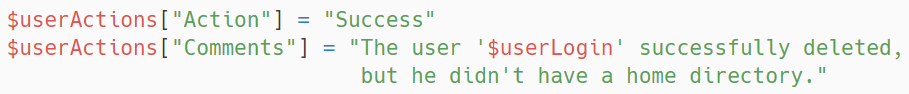
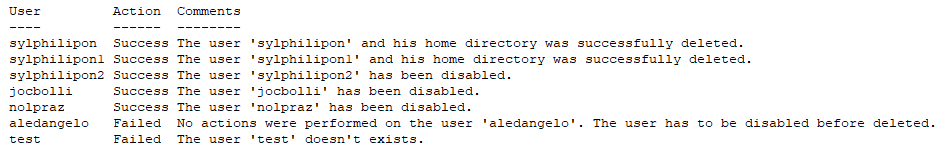


Figure 87 Utilisateur supprimé mais il n'avait pas de répertoire personnel

Une fois tous les comptes supprimés ou désactivés, le fichier csv est exporté avec les données des utilisateurs.



## Description des tests effectués

| Test | Impact en cas d’échec | Résultat attendu | Résultat obtenu | Conclusion | Temps requis pour corriger |
| --- | --- | --- | --- | --- | --- |
| Une recherche est effectuée pour vérifier si l'utilisateur existe déjà dans l'AD. | • Cela peut entrainer une réinitialisation non voulue d’un compte utilisateur. |  | OK 100% | Rien à signaler |  |
| Le nom d’ouverture de session est généré avec les 3 premiers caractères du prénom + les 8 premiers caractères du nom de famille. De plus les diacritiques, espèces et apostrophes sont retirés. | • Compromission du compte.  • Impossibilité d’utilisation du compte. |  | OK 100% | Rien à signaler |  |
| Le nom d’ouverture de session proposé est unique, dans le cas contraire un autre est généré. | • Risques de doublons d’identifiants  • Impossibilité de connexion |  | OK 100% | Rien à signaler |  |
| Un mot de passe aléatoire devant être changé à l’ouverture de session est généré. | • L’utilisateur ne peut pas changer de mot de passe.  • Risques de sécurité. |  | OK 100% | Rien à signaler |  |
| Un script d’ouverture de session est généré selon la profession de l’utilisateur. | • Accès inappropriés.  • Difficultés de gestion |  | OK 100% | Rien à signaler |  |
| Une description est générée selon la classe de l’utilisateur. | • Gestion des comptes plus compliquée. |  | OK 100% | Rien à signaler |  |
| L’utilisateur est créé dans la bonne OU. | • Gestion des utilisateurs plus compliquée.  • Problèmes de sécurité liés aux GPO. |  | OK 100% | Rien à signaler |  |
| Le dossier personnel est créé avec les bons droits d’accès. | • Impossibilité d’avoir un dossier personnel. |  | OK 100% | Rien à signaler |  |
| L’utilisateur est membre des groupes spécifiés dans le CSV. | • Impossibilité d’utilisation du compte. |  | OK 100% | Rien à signaler |  |
| Un fichier CSV contenant les actions effectuées est généré. | • Impossibilité d’obtenir un retour des actions effectuées. |  | OK 100% | Rien à signaler |  |

## Liste des documents fournis

|  |  |
| --- | --- |
| Document | Version |
| Le rapport de projet | 1.0 |
| Le code source du projet | 1.0 |
| Journal de travail | 1.0 |
| Planification détaillée | 1.0 |

Les documents cités ci-dessus sont disponible dans la partie annexe de ce document.

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

## Glossaire

1. Active Directory (AD)
2. Annuaire
3. Scripting
4. Domaine
5. CSV
6. Diacritiques
7. Merged
8. OU
9. Tests unitaires
10. Framework
11. Forme de normalisation
12. Fonction
13. Itération
14. GPO
15. Valeur null
16. Droits NTFS

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## 

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*