

<b>Procédure</b>	TP4 - Registre et Scripts
<b>Auteur</b>	Romain MIAZGA
<b>Date de création</b>	20/02/2026
<b>Nom du fichier</b>	Regedit, script batch
<b>Acteurs concernés</b>	Toute personne voulant créer des fichiers .reg dans la base de registre et faire des scripts batchs (.bat)

## Étape 1 : TP 4.1 Base de registre

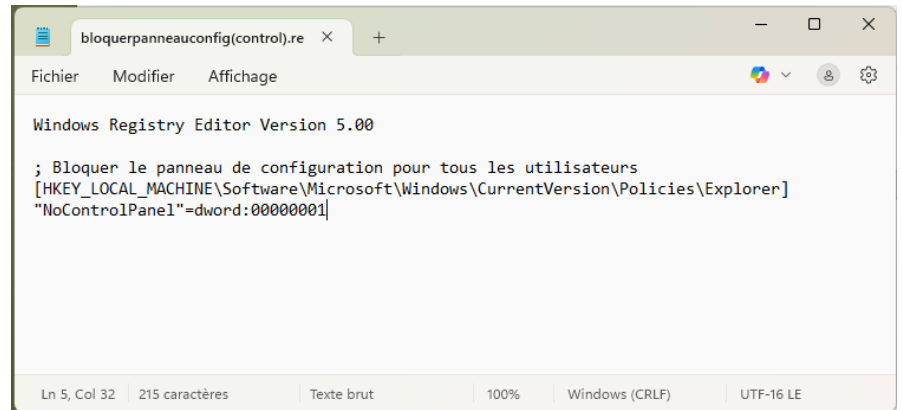
Créer 3 fichiers REG, expliquer la syntaxe d'un fichier REG

Créer un fichier REG permettant d'activer le blocage du panneau de configuration pour les utilisateurs

Pour cela indiquer le chemin et mettre la valeur

"NoControlPanel"=dword :00000001

Lancer le fichier reg pour activer le blocage du panneau de configuration

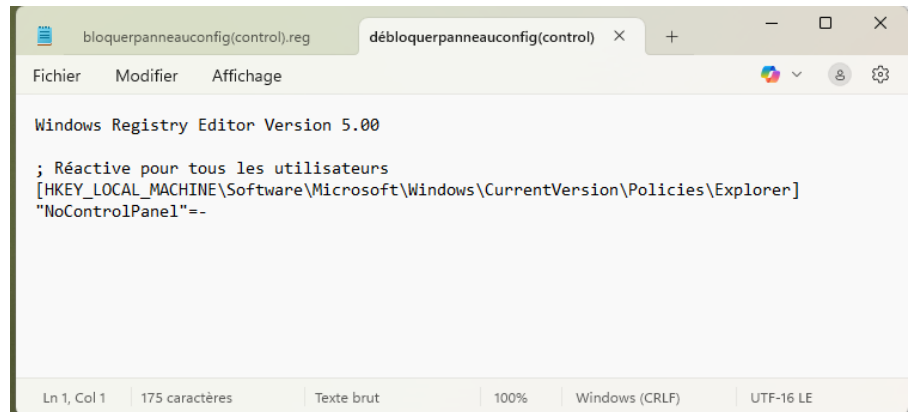


```
Windows Registry Editor Version 5.00

; Bloquer le panneau de configuration pour tous les utilisateurs
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer]
"NoControlPanel"=dword:00000001
```

Pour réactiver le panneau de configuration changer la valeur "dword :00000001" par "dword :=-" ou "dword :00000000"

Lancer le fichier reg pour débloquent l'accès au panneau de configuration



```
Windows Registry Editor Version 5.00

; Réactive pour tous les utilisateurs
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer]
"NoControlPanel"=-
```

Pour forcer l'activation du PAD Numérique au démarrage changer la valeur 2147483650 par 2147483648 depuis la ruche utilisateurs HKEY\_USERS\DEFAULT\ Dans HKEY\_CURRENT\_USER\ mettre la valeur sur 2. Lancer le fichier reg pour forcer l'activation du PAD et constater le résultat lors du redémarrage de l'ordinateur.

```
Windows Registry Editor Version 5.00




; Forcer le pavé numérique au démarrage
; Si après l'exécution du fichier .reg ça ne fonctionne pas, accéder aux paramètres d'options d'alimentation et mettre l'ordinateur en performances élevées (WIN + R > powercfg.cpl)

[HKEY_USERS\DEFAULT\Control Panel\Keyboard]
"InitialKeyboardIndicators"="2147483650"
"KeyboardDelay"="1"
"KeyboardSpeed"="31"

[HKEY_CURRENT_USER\Control Panel\Keyboard]
"InitialKeyboardIndicators"="2"
```

Nos trois fichiers REG sont maintenant configurés et fonctionnels.

Un fichier **.reg** est un fichier texte qui contient un en-tête obligatoire, des chemins de clés du registre entre crochets, et des valeurs associées, permettant d'ajouter, modifier ou supprimer automatiquement des paramètres du registre

 bloquerpanneauconfig(control).reg	20/02/2026 14:09	Fichier REG	1 Ko
 débloquerpanneauconfig(control) - Copi...	20/02/2026 14:11	Fichier REG	1 Ko
 forcerpadnumerique.reg	20/02/2026 14:42	Fichier REG	1 Ko

## Étape 2 : TP 4.2 Scripts batch

*Créer des scripts batch qui exécutent des commandes CMD automatiquement*

La commande "mkdir" permet de créer les dossiers. "& (echo %date% %time% & dir C :) > c:\log\d7\fichier.txt" va nous permettre d'ajouter à ces dossiers un fichier.txt avec la date, l'heure et le contenu de C :

```
C:\Users\ADMIN>mkdir c:\log\d7 & (echo %date% %time% & dir C:) > c:\log\d7\fichier.txt
```

Voici le contenu du fichier log.bat permettant de créer avec Notepad un fichier Batch qui réalise un fichier log incrémental en utilisant les variables d'environnement et qui note sur une même ligne le nom de l'utilisateur, la date et l'heure puis en dessous liste le contenu de «C: » puis crée un répertoire caché au nom de l'utilisateur.

La commande "set" permet de gérer les variables d'environnement en créant, modifiant ou en affichant la variable.

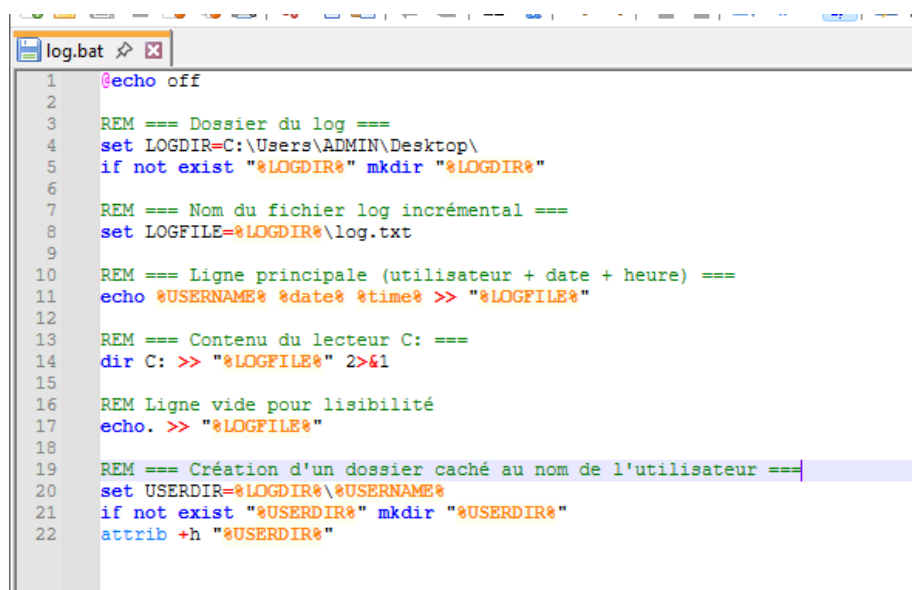
"if no exist" Elle sert à tester si un fichier ou dossier n'existe pas.

"echo." Echo avec un point permet de sauter une ligne

"Attrib +h" Permet de cacher le dossier USERDIR

">>" permet de rajouter du contenu sans supprimer

">" écrase le contenu précédent et le remplace



```
1 @echo off
2
3 REM === Dossier du log ===
4 set LOGDIR=C:\Users\ADMIN\Desktop\
5 if not exist "%LOGDIR%" mkdir "%LOGDIR%"
6
7 REM === Nom du fichier log incrémental ===
8 set LOGFILE=%LOGDIR%\log.txt
9
10 REM === Ligne principale (utilisateur + date + heure) ===
11 echo %USERNAME% %date% %time% >> "%LOGFILE%"
12
13 REM === Contenu du lecteur C: ===
14 dir C: >> "%LOGFILE%" 2>&1
15
16 REM Ligne vide pour lisibilité
17 echo. >> "%LOGFILE%"
18
19 REM === Création d'un dossier caché au nom de l'utilisateur ===
20 set USERDIR=%LOGDIR%\%USERNAME%
21 if not exist "%USERDIR%" mkdir "%USERDIR%"
22 attrib +h "%USERDIR%"
```

Maintenant nous allons créer un fichier REP.BAT qui va nous permettre de passer en paramètre, de créer un fichier de log incrémental en utilisant les variables et variables d'environnement.

Cette commande nous permet de :  
Noter sur une même ligne le nom de l'utilisateur, la date et l'heure, d'ajouter en dessous une ligne avec le nom de l'utilisateur pour lequel on crée un répertoire.

Si le répertoire existe déjà, ajoute une erreur dans le fichier de log et redemande un autre nom d'utilisateur.

### set /p

set → crée une variable  
/p → demande une saisie clavier  
la réponse est stockée pour être réutilisée dans le script.

```
1 @echo off
2 chcp 65001 >nul
3
4 :: Définition du chemin d'accès vers le Bureau
5 set "BUREAU=%USERPROFILE%\Desktop"
6
7 :: Définition du fichier de log pour qu'il se crée sur le Bureau
8 set "FICHIER_LOG=%BUREAU%\log_repertoire.txt"
9
10 :: Récupération du paramètre passé lors de l'appel du script (ex: REP.BAT Toto)
11 set "CIBLE_UTILISATEUR=%~1"
12
13 :VERIFICATION
14 :: Si aucun paramètre n'a été passé au départ, on le demande
15 if "%CIBLE_UTILISATEUR%"==" " (
16     set /p "CIBLE_UTILISATEUR=Veuillez indiquer un nom d'utilisateur : "
17     goto VERIFICATION
18 )
19
20 :: 1. Note sur une même ligne le nom de l'utilisateur courant, la date et l'heure
21 echo Utilisateur: %USERNAME% - Date: %DATE% - Heure: %TIME% >> "%FICHIER_LOG%"
22
23 :: 2. Ajoute en dessous une ligne avec le nom de l'utilisateur cible
24 echo Tentative de creation du repertoire pour : %CIBLE_UTILISATEUR% >> "%FICHIER_LOG%"
25
26 :: 3. Vérification si le répertoire existe déjà SUR LE BUREAU
27 if exist "%BUREAU%\%CIBLE_UTILISATEUR%" (
28     :: Ajout de l'erreur dans le log
29     echo [ERREUR] Le repertoire "%CIBLE_UTILISATEUR%" existe deja sur le Bureau ! >> "%FICHIER_LOG%"
30     echo Le repertoire "%CIBLE_UTILISATEUR%" existe deja sur le Bureau.
31
32     :: On vide la variable et on redemande un autre nom
33     set "CIBLE_UTILISATEUR="
34     set /p "CIBLE_UTILISATEUR=Entrez un NOUVEAU nom d'utilisateur : "
35
36     :: On repart au début de la boucle pour retenter et reloguer
37     goto VERIFICATION
38 ) else (
39     :: Création du répertoire sur le Bureau si tout est bon
40     mkdir "%BUREAU%\%CIBLE_UTILISATEUR%"
41     echo [SUCCES] Repertoire "%CIBLE_UTILISATEUR%" cree avec succes sur le Bureau. >> "%FICHIER_LOG%"
42     echo Le repertoire "%CIBLE_UTILISATEUR%" a ete cree sur votre Bureau !
43 )
44
45 echo Fin de l'operation.
```

"@echo off" : Il empêche l'ordinateur de répéter à voix haute toutes les lignes de code qu'il est en train de lire. Grâce à lui, votre écran reste propre et n'affiche que le résultat final (vos messages et vos questions).

"chcp 65001 >nul" C'est l'abréviation de *Change Code Page*. Il sert à modifier l'alphabet utilisé par la console. En utilisant chcp 65001, on force le système à comprendre l'UTF-8, ce qui permet d'afficher correctement tous les accents français (é, è, à) sans qu'ils ne se transforment en symboles bizarres.

### if (Si...)

C'est ce qu'on appelle une **condition**. Cela permet au script de prendre une décision. On dit à l'ordinateur : "*Si telle chose est vraie, ALORS fais ceci*".

### if exist (Si il existe...)

C'est une variante de la commande if, mais elle est **spécialement conçue pour vérifier le disque dur** (les fichiers ou les dossiers). Elle ne vérifie pas du texte, elle vérifie la réalité matérielle sur votre ordinateur.

**:VERIFICATION (Le marque-page)**

Attention au petit deux-points (:) au début, il est crucial ! Ce n'est pas une commande qui exécute une action, c'est **une étiquette** (ou un point de repère). Imaginez que c'est un post-it ou un marque-page que vous glissez dans un livre.

Quand le script rencontre une erreur (le dossier existe déjà), il lit la commande goto VERIFICATION. L'ordinateur cherche alors l'étiquette :VERIFICATION plus haut dans le texte et **recommence la lecture à partir de ce point**. C'est ce qui crée une "boucle" pour redemander un nom à l'infini tant que ce n'est pas bon !

**setlocal / endlocal** : Créent une "bulle" de sécurité. Les réglages faits dans ce script disparaîtront à la fin sans perturber le reste de votre ordinateur.

### 1. Configuration

**set "NOM=Chemin"** : Créé des "raccourcis" ou des "boîtes". Au lieu de retaper les longs emplacements de vos dossiers à chaque fois, le script les mémorise sous des noms simples (comme SRC\_CLASSE).

### 2. Génération de la date for /f ... powershell ... :

C'est une astuce intelligente. Elle demande à Windows la date du jour précis au format AnnéeMoisJour (ex: 20260223) pour nommer le dossier de sauvegarde. C'est la méthode la plus fiable.

### 3. Exécution de la sauvegarde

**echo** : Affiche du texte à l'écran (ex: "Lancement de la sauvegarde...") pour vous tenir informé de l'avancement.

**if not exist ... mkdir ...** : Vérifie si le dossier d'aujourd'hui existe déjà sur le PC de destination. Si ce n'est pas le cas, il le crée (mkdir pour *make directory*).

**robocopy** : C'est le "super-copieur" de Windows, bien plus puissant qu'un simple copier-coller.

```
1 @echo off
2 chcp 65001 >nul
3 setlocal
4
5 :: =====
6 :: 1. CONFIGURATION DES CHEMINS (À MODIFIER)
7 :: =====
8
9 :: Chemins locaux (poste source)
10 set "SRC_CLASSE=C:\CLASSE"
11 set "SRC_DOCPROFS=C:\DOCPROFS"
12 set "SRC_PUBLIC=C:\PUBLIC"
13
14 :: Chemin réseau vers le poste de sauvegarde
15 set "DEST_BASE=\\PC2\Users\PROF.PC2\sauvegardes"
16
17 :: =====
18 :: 2. GÉNÉRATION DE LA DATE (FIABLE)
19 :: =====
20
21 :: Génère AAAAMMJJ sans dépendre de la langue Windows
22 for /f %I in ('powershell -NoProfile -Command "Get-Date -Format yyyyMMdd"') do set "DATE_JOUR=%I"
23
24 :: Chemin final
25 set "DEST_FINALE=%DEST_BASE%\%DATE_JOUR%"
26
27 :: =====
28 :: 3. EXÉCUTION DE LA SAUVEGARDE
29 :: =====
30
31 echo Lancement de la sauvegarde du %DATE_JOUR%...
32 echo Destination : %DEST_FINALE%
33 echo.
34
35 :: Création du dossier du jour si nécessaire
36 if not exist "%DEST_FINALE%" mkdir "%DEST_FINALE%"
37
38 :: Copie avec Robocopy
39 echo [1/3] Sauvegarde du dossier "Classe"...
40 robocopy "%SRC_CLASSE%" "%DEST_FINALE%\CLASSE" /E /R:1 /W:1
41
42 echo [2/3] Sauvegarde du dossier "Docprofs"...
43 robocopy "%SRC_DOCPROFS%" "%DEST_FINALE%\DOCPROFS" /E /R:1 /W:1
44
45 echo [3/3] Sauvegarde du dossier "Public"...
46 robocopy "%SRC_PUBLIC%" "%DEST_FINALE%\PUBLIC" /E /R:1 /W:1
47
48 :: =====
49 :: 4. FIN
50 :: =====
51
52 echo.
53 echo Sauvegarde terminée avec succès dans le dossier %DATE_JOUR% !
54 pause
55 endlocal
```

\*

**/E** : Copie absolument tout, y compris les sous-dossiers (même s'ils sont vides).

**/R:1 et /W:1** : S'il n'arrive pas à copier un fichier (par exemple s'il est ouvert), il réessaiera 1 fois (R:1), en patientant 1 seconde (W:1). Ça évite que le script ne reste bloqué à l'infini.