UAC

publié par JCB le mar, 13/09/2011 - 18:40

"**UAC**" (*User Account Control*) est une fonctionnalité qui a été introduite dans Windows **Vista**, c'est à dire depuis le **30/11/2006**.
Elle est souvent sujette à controverses, surtout chez les utilisateurs habitués des précédentes versions de la famille Windows NT, à cause des bouleversements qu'elle a entrainés!

Rappels sur les privilèges

Depuis **Windows NT 3.1** (1993), tout compte utilisateur possède un certain nombre de "**privilèges**", dépendant de la catégorie à laquelle il appartient :

- administrateurs
- utilisateurs ordinaires
- invités

A chaque privilège, identifié par un nom, correspond une action spécifique, par exemple :

Nom	*Rôle*
SeSecurityPrivilege	Gérer le journal d'audit et de sécurité
SeTakeOwnershipPrivilege	Prendre possession de fichiers ou d'autres objets
SeLoadDriverPrivilege	Charger et décharger les pilotes de périphériques
SeSystemtimePrivilege	Modifier l'heure

	système
SeBackupPrivilege	Sauvegarder les fichiers et les répertoires
SeRestorePrivilege	Restaurer les fichiers et les répertoires
SeShutdownPrivilege	Arrêter le système
SeCreateSymbolicLinkPrivilege	Créer des liens symboliques
...	...

Lorsqu'un compte utilisateur ouvre une session, le sous-système de gestion de la sécurité (**LSA =** *Local Security Authority*) attribue à ce compte un "jeton" (*token* en anglais, = un objet mémoire) qui contient la liste des privilèges auxquels le compte a droit.
Si une opération relevant de la **sécurité** a lieu (création d'un dossier, arrêt de l'ordinateur, création de compte, changer le fuseau horaire, ...), le "jeton" est consulté par le système qui **autorise** ou **interdit** le processus.

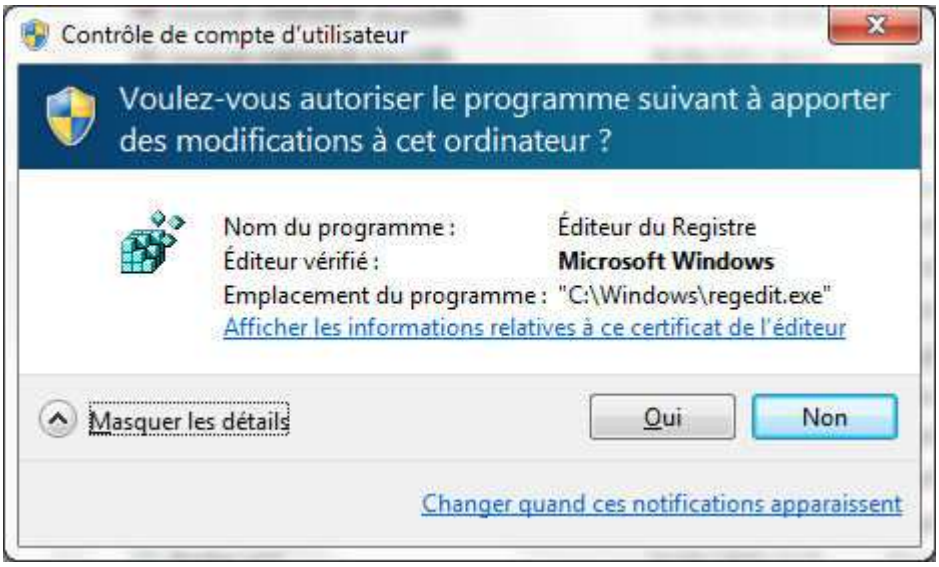
Ce qui change à partir de VISTA ...

Ce qui précède est toujours valable, **sauf** en ce qui concerne l'attribution du jeton par **LSA** aux comptes **administrateurs** .
Appartenir au groupe des **administrateurs ne signifie plus forcément** avoir les **privilèges d'administrateur**!

Ainsi, quand un compte **administrateur**, **autre** que **le** compte *Administrateur* (= celui dont le *Security IDentifier* se termine par **500**, cf. plus bas), ouvre une session, il lui est attribué non pas **1** mais **2 jetons** :

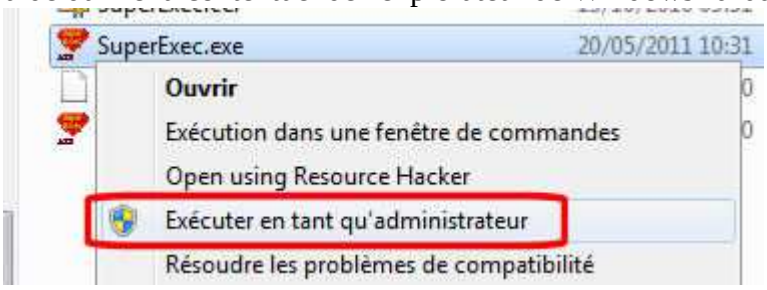
- l'un est celui d'administrateur (avec **tous** les privilèges)
- l'autre celui d'un compte ordinaire (avec des privilèges **limités**).

Mais par défaut, c'est le jeton *ordinaire* qui est **actif**, alors que le jeton *administrateur* est **inactif**.
Donc un compte **administrateur** se retrouve avec les privilèges réduits d'un compte *ordinaire.*
Cela a pour but de **renforcer la sécurité**, en limitant volontairement **par défaut** les privilèges effectifs des **administrateurs** (et éviter des erreurs et/ou des attaques provenant de "malwares").
Si un processus a besoin de privilèges élevés (notifiés p.ex. dans un fichier "xxxxx.manifest" par la ligne *<requestedexecutionlevel level="requireadministrator" /> "* , le jeton en cours est insuffisant, et il y a alors demande d'élévation de privilèges, signifiée, par défaut, par une boite de dialogue.



Si l'utilisateur (du groupe des administrateurs) confirme cette demande (parce qu'il sait que c'est lui qui est à l'origine de la demande, et non pas un processus inconnu), il y a alors utilisation du jeton *administrateur*, si bien que le processus peut se poursuivre.
Dès que le processus est terminé, le jeton *administrateur* est désactivé, et le jeton *ordinaire* est à nouveau actif.

On peut forcer l'exécution d'un exécutable quelconque avec les privilèges administrateurs à l'aide du menu contextuel de l'explorateur de Windows "*exécuter en tant qu'administrateur*"



Il s'ensuit l'ouverture de la boite de dialogue de **confirmation** l'élévation de privilèges.

Ainsi, pour pouvoir modifier ou supprimer certains dossiers, sous-dossiers et/ou fichiers (p.ex. dans %systemroot%\System32 ou %programfiles%), ou encore des clefs du Registre, il faut **au préalable** avoir lancé l'explorateur (*explorer.exe*), la fenêtre de commandes (*cmd.exe*), l'éditeur du Registre (*regedit.exe*),..., **en tant qu'administrateur**, puis avoir **confirmé** l'élévation de privilèges.

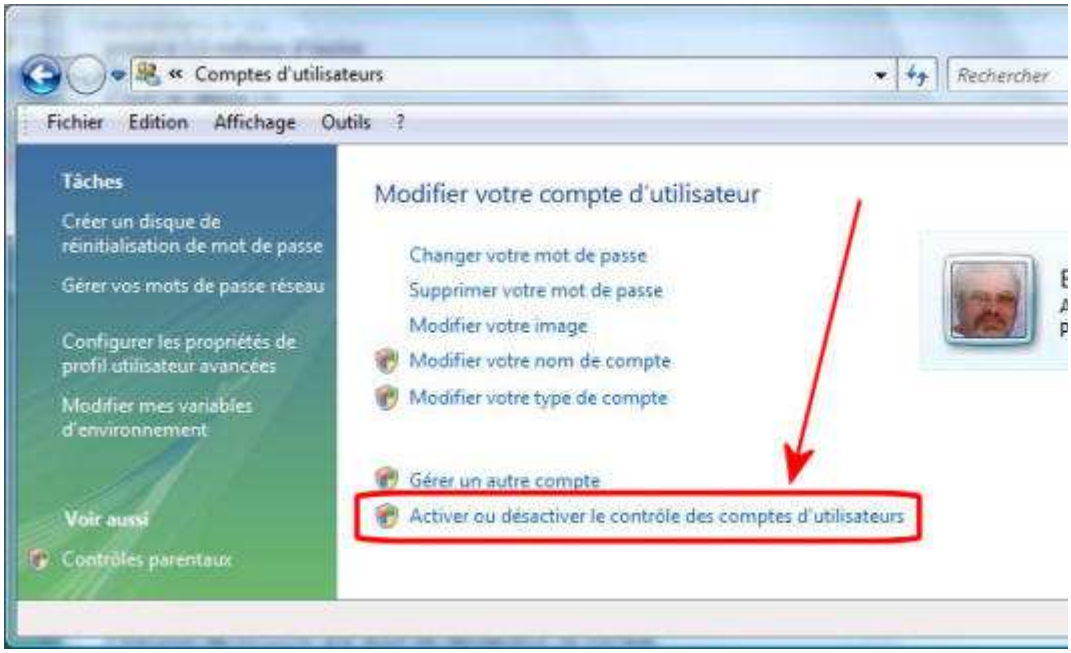
Le **problème** avec ce système est que ces demandes de confirmation d'élévation de privilèges sont demandées en **permanence**, à chaque fois que le lancement d'un processus exige d'être réalisé en tant qu'administrateur. Par exemple, si on a besoin, au cours de la même session, d'exécuter une vingtaine de fois l'éditeur du Registre "*regedit.exe*", il faudra confirmer l'élévation de privilèges une vingtaine de fois!

Si on ne veut plus être dérangé par ces incessantes confirmations d'élévation de privilèges, il faut **paramétrer** en conséquence **UAC**.
Les méthodes **diffèrent** suivant que l'on est sous **Vista** ou sous une **version ultérieure**.

Paramétrage de UAC sous Vista

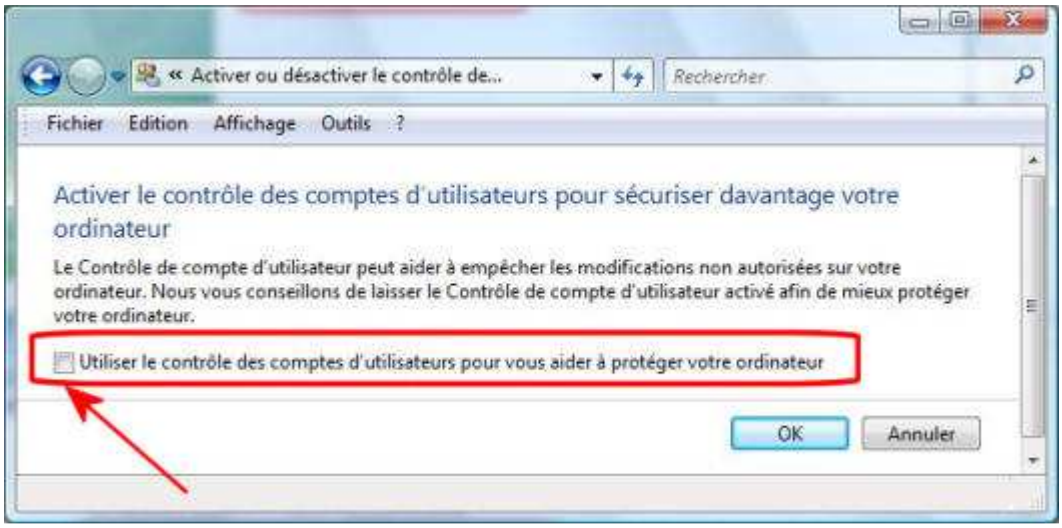
L'interface de Vista ne permet qu'un fonctionnement "**tout ou rien**", qui se choisit à partir du panneau de configuration "*Comptes d'utilisateurs*"
(des captures d'écran ayant été réalisées sous une version US de Vista, certains libellés sont en anglais)

Ouvrir le panneau de configuration ***Comptes d'utilisateurs***

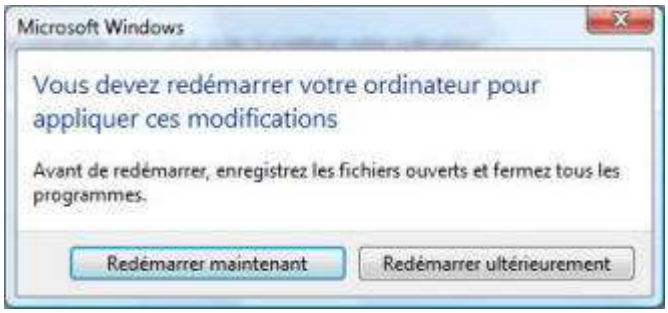


Cliquer sur le lien ***Activer ou désactiver le contrôle des comptes d'utilisateur***

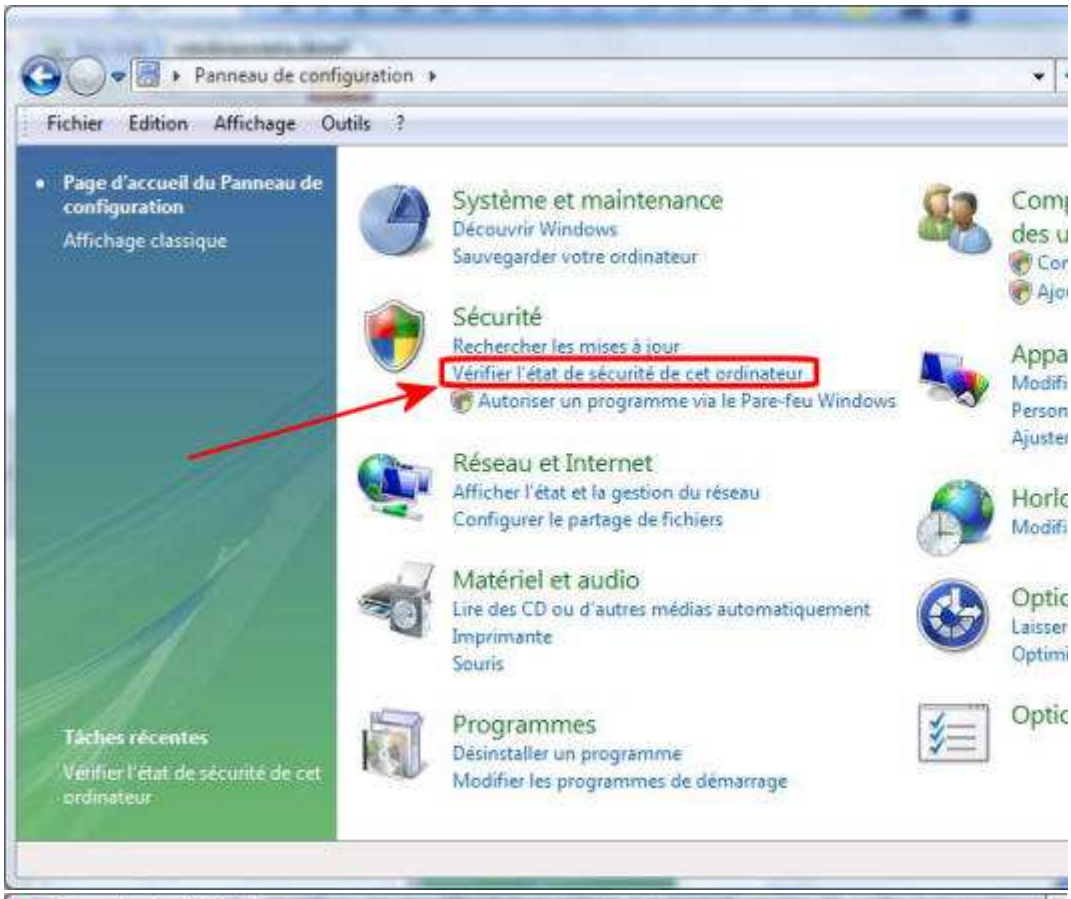
Décocher la case d'utilisation de **UAC**



Redémarrer Windows pour que la modification soit prise en compte

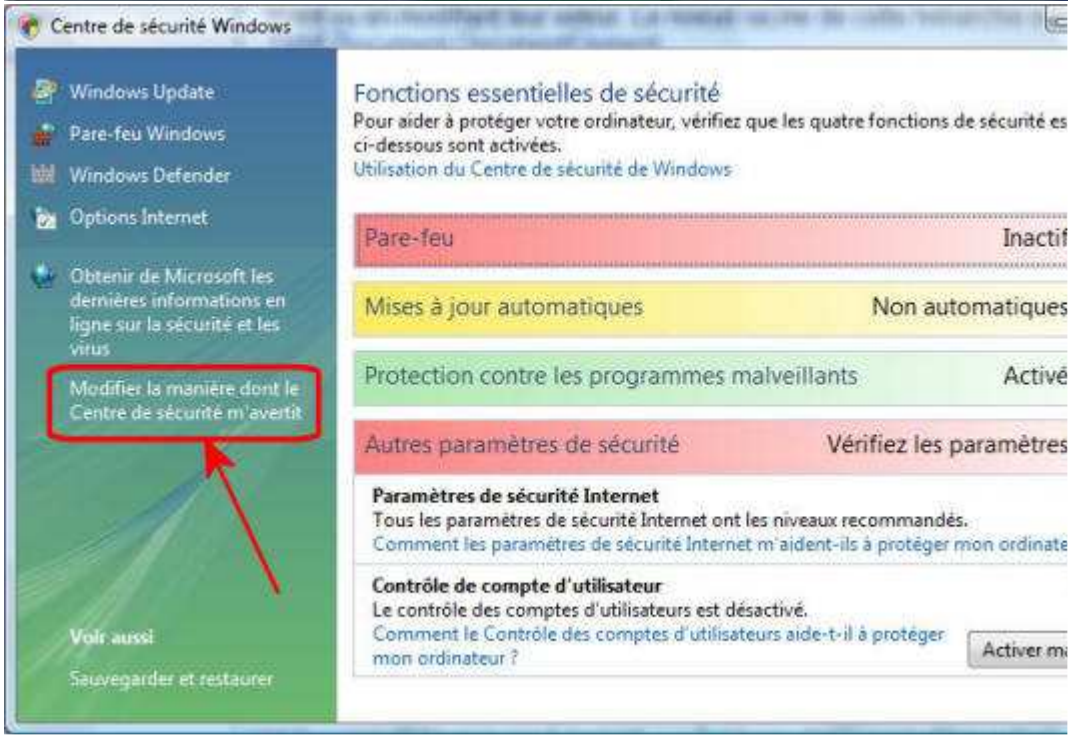


Après redémarrage, ouvrir le panneau de configuration ***Sécurité / Vérifier l'état de sécurité de cet ordinateur***

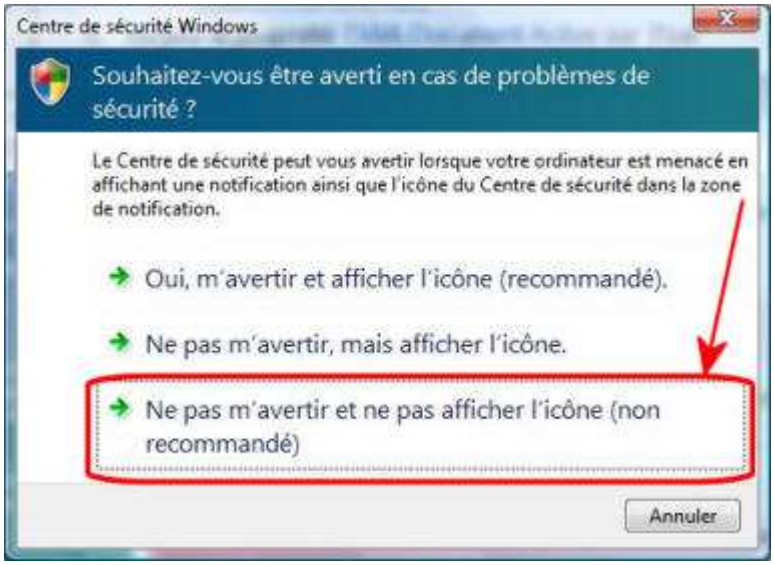


Vérifier que UAC est bien **désactivé**

Cliquer sur le lien ***Modifier la manière dont le Centre de sécurité m'avertit***



Sélectionner la 3ème option "***Ne pas m'avertir et ne pas afficher l'icône***"



et surtout ne pas se laisser impressionner ni dissuader par la remarque "*non recommandé*" !

Ce paramétrage évitera les rappels incessants et se voulant culpabilisants signalant la désactivation d'**UAC**.

Ces paramètres sont stockés dans les clefs suivantes de la **Base de registres** :

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA
0 UAC désactivé
1 UAC activé
HKLM\SOFTWARE\Microsoft\Security Center\Svc\<CLSID du compte concerné>\EnableNotifications
0 Notifications désactivées et aucune icône

1 Notifications désactivées mais icône affichée

- Notifications activées

☞Il existe **d'autres paramètres** qui permettent de régler **UAC** plus en détail (en gras la valeur par défaut), mais ils **ne sont accessibles** que dans la base de **registres** :

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\ConsentPromptBehaviorAdmin

0 Élévation des privilèges sans demander confirmation

1 Demande d'identification d'administrateur (nom + mot de passe)

2 Demande de consentement (accepter/refuser)

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\ValidateAdminCodeSignatures

0 Exécution normale des exécutables non signés

1 Seuls les exécutables signés pourront subir une élévation de privilèges

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\PromptOnSecureDesktop

0 Il n'y a pas de changement de bureau quand on élève les privilèges

1 On passe sur bureau sécurisé quand on élève les privilèges [(1)]

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\FilterAdministratorToken

0 Le compte "Administrateur" n'est pas soumis aux approbations

1 Le compte "Administrateur" est soumis aux approbations d'élévation de privilèges comme les autres administrateurs.

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\ConsentPromptBehaviorUser

0 Sous un compte standard, toute demande d'élévation de privilèges sera refusée

1 Sous un compte standard, toute demande d'élévation de privilèges sera soumise à la saisie d'un nom et mot passe administrateur

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableVirtualization

0 la virtualisation [(2)] des dossiers et clefs est désactivée

1 la virtualisation des dossiers et clefs est activée

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableInstallerDetection

0 l'installation d'un programme se fait comme sous XP et précédemment

1 si l'installation d'un programme nécessite d'être sous un compte administrateur, une demande d'élévation de privilège aura lieu.

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableSecureUIAPaths

0 L'élévation de privilège pour des applications **UIAccess** [(3)] aura lieu même si elles ne sont pas installées dans des emplacements sécurisés [(4)].

1 L'élévation de privilège pour des applications UIAccess n'aura lieu que si elles sont installées dans des emplacements sécurisés.

[(1)] Cela occasionne un écran noir fugitif.

[(2)] La virtualisation permet à une application lancée sous un compte standard de ne pas générer d'erreur si elle tente d'écrire dans des dossiers ou clefs interdits en écriture (p.ex. %programfiles%, HKLM, ...).
Ainsi une écriture de fichier dans **%programfiles%** se fera en réalité dans **%LOCALAPPDATA%\VirtualStore** .

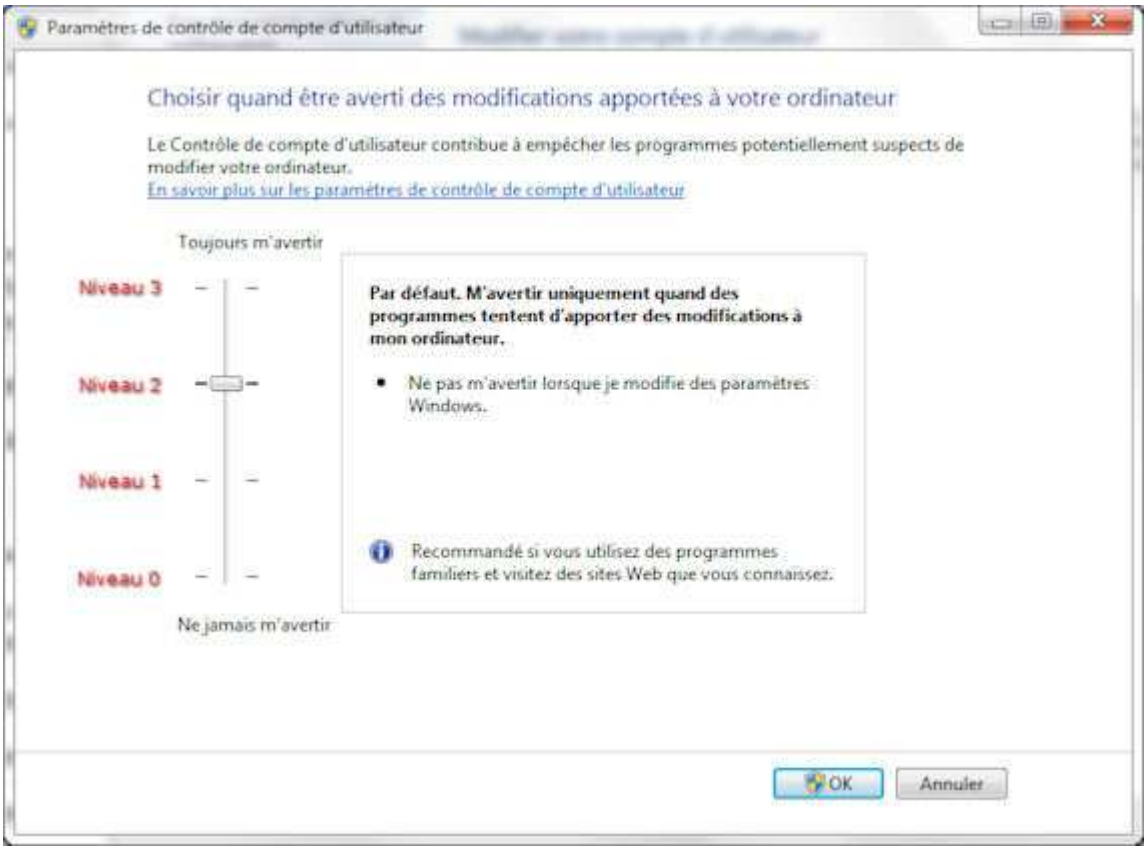
[(3)] Une application **UIAccess** ("Accès à l'interface utilisateur") est une application qui nécessite un haut niveau de privilèges, lequel est défini dans son fichier ***manifest*** par la présence de l'attribut *uiAccess* égal à "*true*" dans la balise *requestedExecutionLevel* :

```
...
<trustInfo
 xmlns=”urn:schema-microsoft-com:asm.v3”>
 <security>
  <requestedPrivileges>
   <requestedExecutionLevel
    Level=”requireAdministrator”
    uiAccess=”true”/>
  </requestedPrivileges>
 </security>
</trustInfo>
```

[(4)] Les emplacements sécurisés désignent les dossiers **%systemroot%\system32** et **%programfiles%** (sous-dossiers compris)

Paramétrage de UAC sous Windows 7 et au delà

Alors que sous **Vista** on ne peut que activer ou désactiver **totalement** UAC (à moins de passer par la base de Registres), sous **Windows 7** (et +) on peut le régler plus finement depuis le panneau de configuration "*comptes utilisateurs*" en cliquant sur le lien "*modifier les paramètres de contrôle de compte d'utilisateurs*".
Cela provoque l'ouverture d'une boite de dialogue "*Paramètres de contrôle de compte d'utilisateur*", comportant une réglette sur **4 niveaux** :



Suivant le niveau choisi, on fixe les valeurs des entrées (déjà rencontrées sous Vista) **ConsentPromptBehaviorAdmin**, **EnableLUA** et **PromptOnSecureDesktop** de la clef **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System** :

Entrées / Niveaux	0	1	2	3
ConsentPromptBehaviorAdmin	0	5	5	2
EnableLUA	0	1	1	1
PromptOnSecureDesktop	0	0	1	1

☞**ATTENTION !**

En ce qui concerne **ConsentPromptBehaviorAdmin** , les valeurs respectives **ne sont pas les mêmes** suivant que l'on est sous **Vista** ou **Windows 7** (et +)!

ConsentPromptBehaviorAdmin		
Vista	***7 et +***	*Effet*
0	0	Élévation des privilèges sans demander confirmation
1	2	Demande d'identification d'administrateur (nom + mot de passe)
2	5	Demande de consentement (accepter/refuser)(valeur par défaut)

☞**ATTENTION !**

En ce qui concerne **EnableLUA**, les valeurs respectives **ne sont pas les mêmes** suivant que l'on est sous **Vista et Windows 7** ou sous **Windows 8**!

En effet, sous **Windows 8**, même au **niveau 0**, **EnableLUA** reste égal à **1**, ce qui signifie qu'il est **impossible** de désactiver complètement UAC sous Windows 8 **sauf** en modifiant directement l'entrée de la Base de registres!

☞Si on a **désactivé totalement UAC** (EnableLUA à 0), il devient **impossible** d'accéder aux **tuiles** concernant le réseau (météo, Windows store, Internet explorer, ...).

Les applications de l'interface "bureau" ne sont pas concernées par cette restriction.

Les autres entrées citées dans le paragraphe consacré à Vista sont **identiques**.

LE compte "Administrateur"

Ce compte, dont le SID (Security IDentifier) se termine par 500 (p.ex. *S-1-5-21-1164180264-3434086188-853729041-500*) est créé automatiquement lors de l'installation de Windows.
Il possède les propriétés suivantes :

- Il **ne peut pas être supprimé**
  Si on tente d'exécuter la commande

  ```
  NET USER Administrateur /DELETE
  ```

  l'erreur suivante est générée :

  ```
  L'erreur système 1371 s'est produite.
  Impossible d'accomplir cette action sur des comptes prédéfinis.
  ```

- il est **désactivé par défaut**.
  Pour l'activer, il suffit d'exécuter (sous un autre compte administrateur) la commande :

  ```
  NET USER Administrateur /ACTIVE:YES
  ```

- bien que désactivé, il peut ouvrir une session, mais seulement en **MODE SANS ÉCHEC** et à la condition qu'il n'existe **aucun autre** compte administrateur.

- il **n'est pas soumis à UAC par défaut**, car il ne reçoit de LSA qu'un seul jeton, celui d'administrateur.
  (**sauf** si on a **volontairement** demandé qu'il soit soumis à la règle de confirmation d'élévation de privilèges, en fixant la valeur de l'entrée **FilterAdministratorToken** à 1)

Donc une autre solution pour se soustraire à UAC est d'ouvrir une session sous **le** compte "**Administrateur**", sous réserve qu'il ait été activé au préalable.

Comment programmer le niveau requis de privilèges

Si on veut **forcer** l'élévation de privilèges (au niveau "*administrateur*" en général), il n'y a pas d'autre solution (dans le cas où processus ne réclame pas lui-même ce niveau grâce à un fichier "*.manifest*" - ou compilé avec - contenant l'instruction <*requestedexecutionlevel level="requireadministrator"* />) que de **relancer** le processus via un appel à la fonction ***ShellExecuteEx*** (de l'API shell32.dll)
cf. http://msdn.microsoft.com/en-us/library/bb762154(VS.85).aspx

Cette fonction admet en paramètre une structure ***SHELLEXECUTEINFO***
cf. http://msdn.microsoft.com/en-us/library/bb759784(v=VS.85).aspx

Parmi les membres de cette structure on notera :

LPCTSTR **lpFile**;	chemin du fichier exécutable (ou document)
LPCTSTR **lpParameters**;	paramètres éventuellement transmis
LPCTSTR **lpVerb**;	la commande désirée. Jusqu'à Windows XP, on avait le choix

entre

- ***edit***
- ***explore***
- ***find***
- ***open*** (cas le plus fréquent)
- ***print***
- ***properties***

A partir de Vista, avec l'apparition de UAC, s'ajoute la commande :

- ***runas*** qui est donc celle à utiliser dans le cas précis pour forcer **l'élévation de privilèges**.

Voici un **script VBS** qui fait la même chose :
(il admet en paramètre l'exécutable ou le document suivi de paramètres éventuels)
Fichier "elevate.vbs"

```
---------- couper ici ----------
Set args = Wscript.Arguments
nbargs=args.count
If nbargs=0 Then Wscript.quit
Appli=args(0)
Params=""
For i= 1 To nbargs-1
   Params=Params+" "+args(i)
Next
Set Shell=Wscript.CreateObject("Shell.Application")
Shell.ShellExecute Appli,Params,"","runas"
---------- couper ici ----------
```

Il existe aussi la fonction ***SHCreateProcessAsUserW***
cf. http://msdn.microsoft.com/en-us/library/bb762138(VS.85).aspx
qui joue le même rôle que ***ShellExecuteEx*** mais avec **implicitement** le verbe **runas**

Le **problème** est qu'elle n'existe pas sous XP et précédents, donc si on veut une application portable sous toutes les versions de Windows, on ne peut pas l'utiliser (à moins de faire des tests supplémentaires). C'est pourquoi je lui préfère ***ShellExecuteEx***.

☞**ATTENTION** aux erreurs dans le MSDN !!!

P.ex. on lit, à propos de ***SHCreateProcessAsUserW*** :

"...SHCreateProcessAsUserW is not supported under Windows XP. "
et 4 lignes plus bas :
"Minimum supported client : Windows 2000 Professional, Windows XP" !!!

et en ce qui concerne ***ShellExecuteEx*** (qui fonctionne **très bien** sous Windows 2000) :

"Minimum supported client : Windows XP "

☞Pour les programmeurs **Delphi** : **création** d'un fichier *.manifest* et **compilation** du programme avec ce fichier

1. **Créer** (avec un éditeur texte tel le bloc-notes p.ex.) un fichier nommé (p.ex.) *RequireAdmin.manifest*

```
------------ couper ici ------------
<?xml version="1.0" encoding="utf-8" ?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1"
manifestVersion="1.0">
<assemblyIdentity version="1.0.0.0" processorArchitecture="X86"
name="Project3" type="win32" />
<description>Privilèges admin obligatoires</description>
<trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
<security>
<requestedPrivileges>
<requestedExecutionLevel level="requireAdministrator" />
</requestedPrivileges>
</security>
</trustInfo>
</assembly>
------------ couper ici ------------
```

2. **Créer** un autre fichier texte nommé (p.ex.) *RequireAdmin.rc* contenant cette unique ligne :

```
------------ couper ici ------------
1 24 "RequireAdmin.manifest"
------------ couper ici ------------
(pour information, 1 est l'ID de la ressource, et 24 son type)
```

3. Dans une fenêtre de commandes, en se plaçant dans le dossier contenant les 2 fichier .manifest et .rc, **exécuter** la commande suivante :

```
brcc32 RequireAdmin.rc
```

```
Cela va générer un fichier "RequireAdmin.res"
```

4. **Insérer** la ligne suivante dans l'**unité principale**, p.ex. après la déclaration "*Uses*" de "*implementation*", :

```
{$R 'RequireAdmin.RES'}
```

5. **Compiler** l'application

En résumé

Lors d'une ouverture de session, **LSA** (le sous-système de gestion de la sécurité) attribue :

1. si et seulement si l'utilisateur est **LE** compte "**Administrateur**" :
   Un "jeton" d'administrateur activé (privilèges étendus)
2. si et seulement si l'utilisateur appartient au **groupe des administrateurs** et **n'est pas LE** compte "Administrateur"
   Un jeton d'administrateur désactivé (privilèges étendus)
   ET
   Un jeton de compte ordinaire activé (privilèges restreints)
3. si et seulement si l'utilisateur n'appartient pas au groupe des administrateurs :
   Un jeton de compte ordinaire activé (privilèges restreints)

Si un **processus** nécessite des privilèges **administrateur** :

- Dans le **cas 1**, il ne se passe **rien** de spécial, puisque le compte administrateur **possède déjà** les privilèges en question grâce à son jeton administrateur activé.
  (en faisant évidemment abstraction du cas très spécial où on aurait demandé que le compte "administrateur" soit traité comme les autres comptes administrateurs)
- Dans le **cas 2**, il y a **demande** d'élévations de privilèges, avec le **même** compte, qui se traduit par l'activation temporaire (le temps du processus) du jeton d'administrateur.
- Dans le **cas 3**, il y a **changement** de compte vers un compte du groupe des **administrateurs** (avec saisie de **nom** de compte et **mot de passe**)

Ce compte reçoit temporairement son jeton d'administrateur activé.

☞Si on **désactive** totalement **UAC** (ce qui correspond à **EnableLUA** mis à **0),** et donc revient à un fonctionnement de style XP (et précédents), il faut être **très prudent**!
Car **tout** processus pourra alors s'exécuter avec les privilèges administrateur, donc si p.ex. on a cliqué sans réfléchir sur un lien dans un spam, un "malware" pourra s'exécuter sans aucun contrôle!

‹ Plages horaires d'ouverture de session haut Base de registres ›