**The Registry**

The registry is a **centralised listing of settings for Windows 7, its applications, hardware and user preferences**. It is stored on the hard drive. It stores a myriad of information e.g. your screen resolution, file type associations, where user recent documents, favorites will be stored, settings for your software, etc. Windows 7 and Windows applications can make changes to the registry. E.g. when you change a setting in Control Panel or change your default font in Word, values are created or edited in the registry.

You can also edit the registry yourself. **Making changes to the registry directly, as opposed to letting your software do it for you, is hazardous**. An errant edit can bring your system down. That’s why Microsoft Windows is set up so that you normally don’t have to get involved with the registry. When you change some detail about your system’s configuration using Control Panel, Control Panel writes the necessary updates to the registry for you, and you needn’t be concerned with how it happens. When you install a new piece of hardware or a new program, a myriad of registry modifications take place; again, you don’t need to know the details.

But because the designers of Windows couldn’t provide a user interface for every conceivable customization you might want to make, sometimes working directly with the registry is the only way to get a job done. And sometimes, even when it’s not the only way, it might be the fastest way. So Microsoft Windows 7 provides you with a **registry editor** that you should know how to use—safely.

**The registry can be modified**

**– By software**

**– By the OS**

**– Using the registry editor: regedit.exe**

To edit the registry, choose Start. Then type regedit in the search box. Click regedit.

The registry has 5 collections of settings called hives, which control different aspects of operation. The hives are:

**HKey\_Classes\_Root**: contains information on file types. This is a subset of information in HKey\_Local\_Machine i.e. also contained at HKey\_Local\_Machine\software\classes .

**HKey\_Current\_User**: contains info on the currently logged-on user. This is a subset of information in HKey\_Users

**HKey\_Local\_Machine**: All the information about your computer is contained here

**HKey\_Users**: the master-hive for all user-related information. When you log on your info is copied from here to HKey\_Current\_User

**HKey\_Current\_Config:** contains pointers to other keys in other hives.

Each hive has keys that contain data in a particular format. Some keys contain other keys and some contain keys and values and some just contain values. Each key has at least one *value*. In Registry Editor, that value is known as the *default value*. Many keys have additional values. The names, data types, and data associated with values appear in Registry Editor’s right pane.

The registry is stored on disk as several separate hive files. The appropriate hive files are read into memory when the operating system starts (or when a new user logs on) and assembled into the registry. You can see where the hives of your system physically live by examining the values associated with HKey\_Local\_Machine\System\CurrentControlSet\Control\HiveList.

Notice that one hive, \Registry\Machine\Hardware, has no associated disk file. This hive,

which records your hardware configuration, is completely volatile; that is, Windows creates it fresh each time you turn your system on. Notice also the path specifications for the remaining hive files. Windows assigns drive letters after assembling the registry, so these paths do not specify drive letters.

Exercise:

Find and give path\name for any 2 hives on your machine:

Hive : HKEY\_CLASSES\_ROOT

Path\name Computer\ HKEY\_CLASSES\_ROOT

Hive: HKEY\_CURRENT\_USER

Path\name Computer\ HKEY\_CURRENT\_USER

**File types and HKEY\_CLASSES\_ROOT**

In HKEY\_CLASSES\_ROOT choose .html

Look at its default Data value. It says htmlfile

Look for htmlfile under HKEY\_CLASSES\_ROOT

Note the default value in the AppUserModelID for htmlfile: Microsoft.InternetExplorer.Default (or maybe Firefox.HTML)

Now under htmlfile drop down to Shell then drop to Open, then to Command.

Note the default value: "C:\Program Files\Internet Explorer\IEXPLORER.EXE"

These two sets of keys make up a file type in Windows 7 i.e. the default application for a file type

Exercise

Find out the default application which will open for a .java file extension\_IntelliJ IDEA\_\_\_\_\_\_\_\_\_\_\_\_\_

Find out the default application which will open for a .txt file extension\_\_Notepad\_\_\_\_\_\_\_\_\_\_\_\_\_

**Editing the registry**

**When editing the registry you must know three things:**

**The registry keys exact name and location**

**The value’s data type**

**The value or permissible values for the key**

Each of the following tweaks points to a Registry path, and then describes the value (or

values) that need to be modified. If any of the keys or values specified don't exist on your

system, simply create them as directed. The topics covered include Files, Folders, and

File Types (next), Performance Tweaks, and User Account and Network Settings.

**Exporting keys**

This should be done to any registry keys before changing it. It will give you an easy way of changing back any changes that you make while editing the registry. Right click the key that you want to export, and select **Export**. This creates an ordinary text file called a ‘patch’

Then enter a name for your export file

**Importing keys**

This reverses what you have done by exporting. To revert to what you saved by exporting: usually a double-click on a registry file will merge it for you unless you

have changed the default action.

If you have changes the default you have to right-click the file and select Merge.

There are many places to find suggestions of keys to edit in the registry. One is <http://www.theeldergeek.com/registry_edits.htm> (though right now this is for Windows XP – some should work also in Windows 7)

The following are samples for you to try:

**Note: before doing any changes, export the key as specified above, so that you can return to the original value by importing.**

**Disable access to the Control Panel**

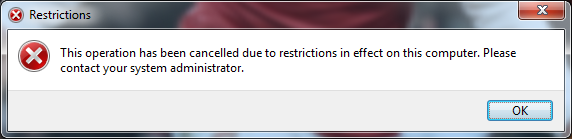
1) Launch Regedit

2) Navigate to this key:  
HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer

3) Create a new VALUE.  Note choose: 'DWORD (32-Bit) Value' Name the new value: NoControlPanel

4) Edit the 'Value data'.  What I do is double click NoControlPanel, then click in the 'Value data:' set to 1.  If it displays as  0x00000001 (1) this is a good sign it will disable your Control Panel.

5) To check the fruits of your labours, close the registry editor and Logoff.  Next Logon and **try starting the Control Panel. It’s not in the Start Menu. Try right-clicking on the Desktop and Personalise. You should receive a 'Restrictions' message similar to the one shown below:**



**Turn off CD AutoPlay**

Open regedit and browse to the following key: HKEY\_LOCAL\_MACHINE\SYSTEM\CURRENTCONTROLSET\SERVICES\CDROM

Set the Autorun DWORD value to 0 to disable CD AutoPlay.

**Enable/Disable Recycle Bin**

For those confident enough that you will never need a deleted file or folder, this tweak will disable the Recycle Bin and permanently delete items with no chance of recovery.

Open regedit and browse to the following key:

HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\BitBucket\Volume\{GUID}

Select the first Volume (if there are more than one)

Modify the Value Data Type(s) and Value Name(s) as detailed below.  
Data Type: DWORD // Value Name: NukeOnDelete  
Setting for Value Data: [0 = Use Recycle Bin / 1 = Permanently Delete]  
Exit Registry and Reboot (do all the edits first before rebooting)

##### Change IE Title Bar Text

This tweak allows you to customize Internet Explorer by adding your own window title.

Open regedit and browse to the following key: HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\Main  
Create the Value Name [Window Title] according to the Value Data listed below.  
Data Type: REG\_SZ [String Value] // Value Name: Window Title  
Value Data: [Enter The Text Desired In Title Bar]  
Exit Registry and Reboot (do all the edits first before rebooting)

##### Launch Programs at Login without Using the Startup Folder

This tweak allows a program to be launched when the user logs onto the computer. It can be set to run minimized and there is no trace of the executable in the Startup folder on the Start Menu nor in the Documents and Settings folder accessible via Windows Explorer.

Open regedit and browse to the following key:

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run

Create the Value Data Type(s) and Value Name(s) as detailed below.  
Data Type: REG\_SZ [String Value] // Value Name: [Enter the Name of Program Executable]  
Value Data: [Enter the Path to the Program Executable]  
Exit Registry and Reboot (do all the edits first before rebooting)

**When you have all the edits done, reboot your machine to see the effect. Then import all the keys back to return to original state.**

**Group Policy Editor (gpedit)**

Securing computers and users' desktops is an important responsibility of the IT administrator. Today's computing environment provides users with hundreds, if not thousands, of configurable settings. Some of these settings are harmless while others could keep help desk staff busy. Domain administrators solve these tough problems using **Group Policy**. How do you solve this problem for stand-alone computers? Microsoft Windows 7 solves this problem by **Multiple Local Group Policy objects**. MLGPOs allow an administrator to **apply different levels of Local Group Policy to local users on a stand-alone computer**. This technology is ideal for shared computing environments **where domain-based management is not available**, such as shared library computers or public Internet kiosks

## Technology Review

Local Group Policy is a subset of a broader technology known as Group Policy. **Group Policy is domain based** **while Local Group Policy is specific to the local computer**. Both technologies allow administrators to **configure specific settings in the operating system and then force those settings to computers and users**. **Local Group Policy is not as robust as Group Policy**. For example, Group Policy allows administrators to configure any number of policies that could affect some, all, or none of the users of a domain-joined computer. Group Policy could even apply policies to users that have specific group memberships. However, Local Group Policy could only apply one policy to the computer and all the local users of the computer, even the local administrator. This made managing the stand-alone computer difficult because the same policy applied to the administrator and the users.

Windows Vista introduced **Multiple Local Group Policy** objects, an improvement over the previous version of Local Group Policy that gives stand-alone computer administrators the ability to apply different Group Policy objects to stand-alone users. Windows Vista provides this ability with **three layers of Local Group Policy objects**: **Local Group Policy, Administrator and Non-Administrators Group Policy, and user specific Local Group Policy**. These layers of Local Group Policy objects **are processed in order, starting with Local Group Policy, continuing with Administrators and Non-Administrators Group Policy, and finishing with user-specific Local Group Policy**.

### Local Group Policy

The Local Group Policy (also known as **Local Computer Policy**) layer is the topmost layer in the list of Multiple Local Group Policy objects. Local Group Policy is the only Local Group Policy object that **allows computer settings**. Besides computer settings, you can select user settings. However, **user settings contained in the Local Group Policy apply to all users of the computer**, even the local administrator. Local Group Policy behaves the same as it did in Windows XP.

### Administrators and Non-Administrators Local Group Policy

Each stand-alone computer running Windows 7 has a list of built-in groups and users. Windows Setup creates this list of users and groups during the installation or upgrade to Windows Vista. One of these groups is the administrators group. The administrators group is a built-in group created by Windows and by default has only one member, the administrator. Windows considers all members of the administrators group to be administrators of the computer. If the user is not a member of the local administrators group, then Windows considers the user to be a member of the local users group (non-administrators).

Administrators and Non-Administrators Local Group Policy objects act as a single layer and logically sort all local users into two groups when a user logs on to the computer. The user is either an administrator or a non-administrator. Users that are members of the administrators group receive policy settings assigned in the Administrators Local Group Policy object. All other users receive policy settings assigned in the Non-Administrators Local Group Policy objects. The Administrators and Non-Administrators Local Group Policy objects are new in Windows Vista.

### User-Specific Group Policy

Administrators of stand-alone computers can create new local user accounts. When created, Windows stores these new accounts with the list of built-in groups and users on the local computer. Local administrators can use the last layer of the Local Group Policy object, Per-User Local Group Policy objects, to apply specific policy settings to a specific local user.

### Processing order

The benefits of Multiple Local Group Policy objects come from the processing order of the three separate layers. The Local Group Policy object applies first. This Local Group Policy object may contain both computer and user settings. User settings contained in this policy apply to all users, including the local administrator. Next, Windows applies Administrators and Non-Administrators Local Group Policy objects. These two Local Group Policy objects represent a single layer in the processing order, and the user receives one or the other. Neither of these Local Group Policy objects contains computer settings. Windows finishes processing Local Group Policy objects by applying user-specific Local Group Policy. This last layer of Local Group Policy objects contains only user settings, and you apply it to one specific user on the local computer.

**To summarize, Windows applies Local Group Policy objects first, then the Administrators or Non-Administrators Local Group Policy objects, and finally the user-specific Local Group Policy objects**.

### Conflict resolution between policy settings

Available user settings are the same between all Local Group Policy objects. It is conceivable **a policy setting in one Local Group Policy object can contradict the same setting in another Local Group Policy object**. Windows resolves these conflict by using the **"Last Writer Wins"** method. This method resolves the conflict by overwriting any previous setting with the last read (most current) setting. The final setting is the one Windows uses.

For example, an administrator enables a setting in the Local Group Policy object. The administrator then disables the same setting in a user-specific Local Group Policy object. The user logging on to the computer is not an administrator. Windows reads the Local Group Policy object first, followed by the Non-Administrators Local Group Policy object, and then the user-specific Local Group Policy object. The state of the policy setting is enabled when Windows reads the Local Group Policy object. The policy setting is not configured in the Non-Administrators Local Group Policy object. This has no affect on the state of the setting, so it remains enabled. The policy setting is disabled in the user-specific Local Group Policy object. This changes the state of the setting to disabled. Windows reads the user-specific Local Group Policy object last; therefore, it has the highest precedence. The Local Computer Policy has lowered precedence.

### Domain member computers

Stand-alone computers benefit the most from Multiple Local Group Policy objects, wherein managing each computer is local. **Domain-based computers apply Local Group Policy first and then domain-based policy**. Windows continues to use the "Last Writer Wins" method for conflict resolution. Therefore, **policy settings originating from domain Group Policy overwrite any conflicting policy settings found in any Local Group Policy** to include administrative, non-administrative, and user specific Local Group Policy. **Domain administrators can disable processing Local Group Policy objects on clients** running Windows 7 by enabling the **"Turn off Local Group Policy objects processing"** policy setting in a domain Group Policy object. **You can find this setting under Computer Configuration\Administrative Templates\System\Group Policy.**

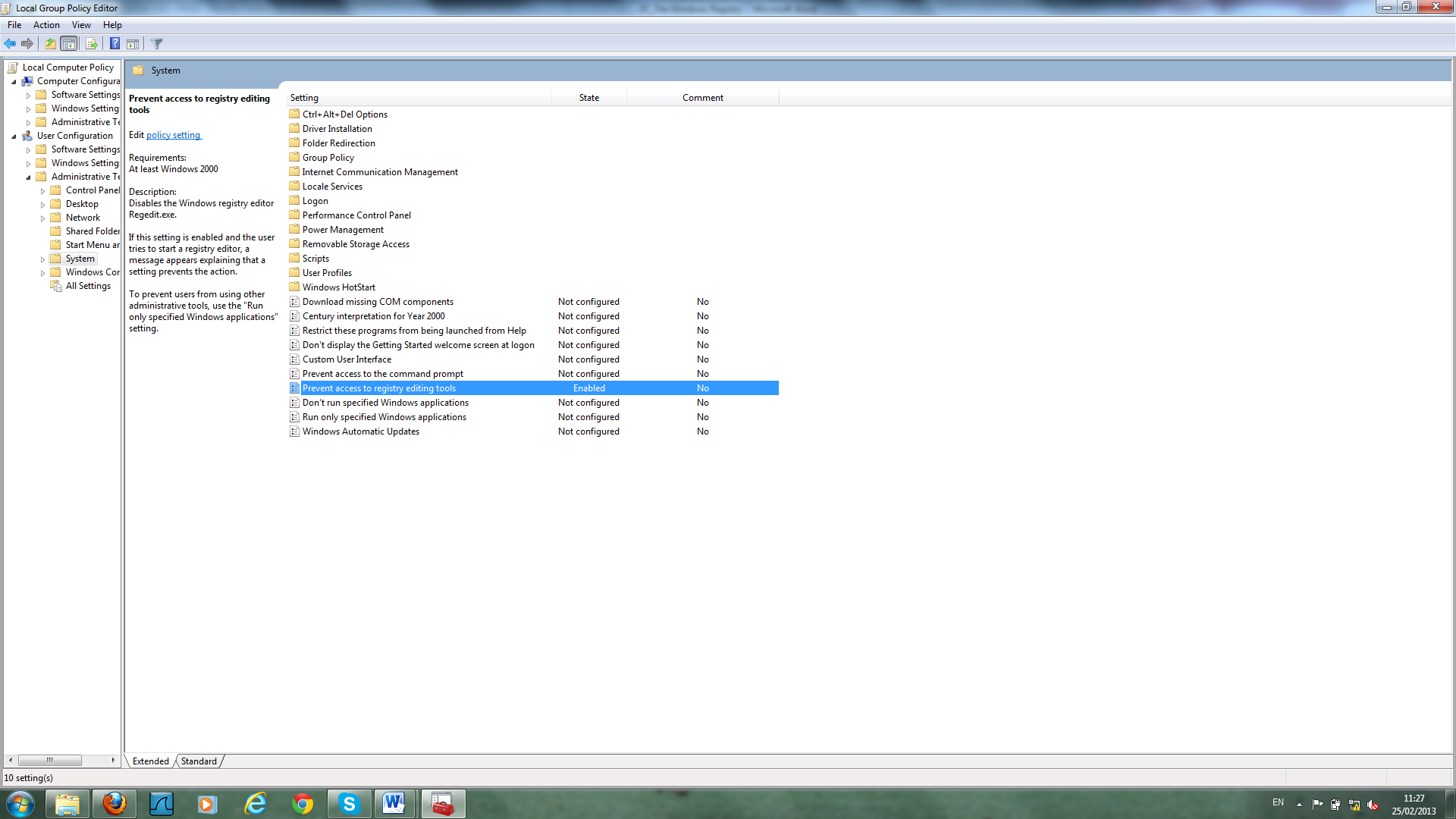
**Local Group Policy Examples**

**You can turn off access to the Registry Editor (regedit) as follows:**

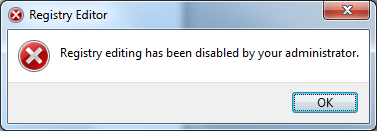
Start Group Policy Editor – **Click Start > Run > gpedit.msc**

Click on the **Local Computer Policy –>User Configuration > Administrative Templates > System**

**Double-click on Prevent access to registry editing tools and select Enable**.



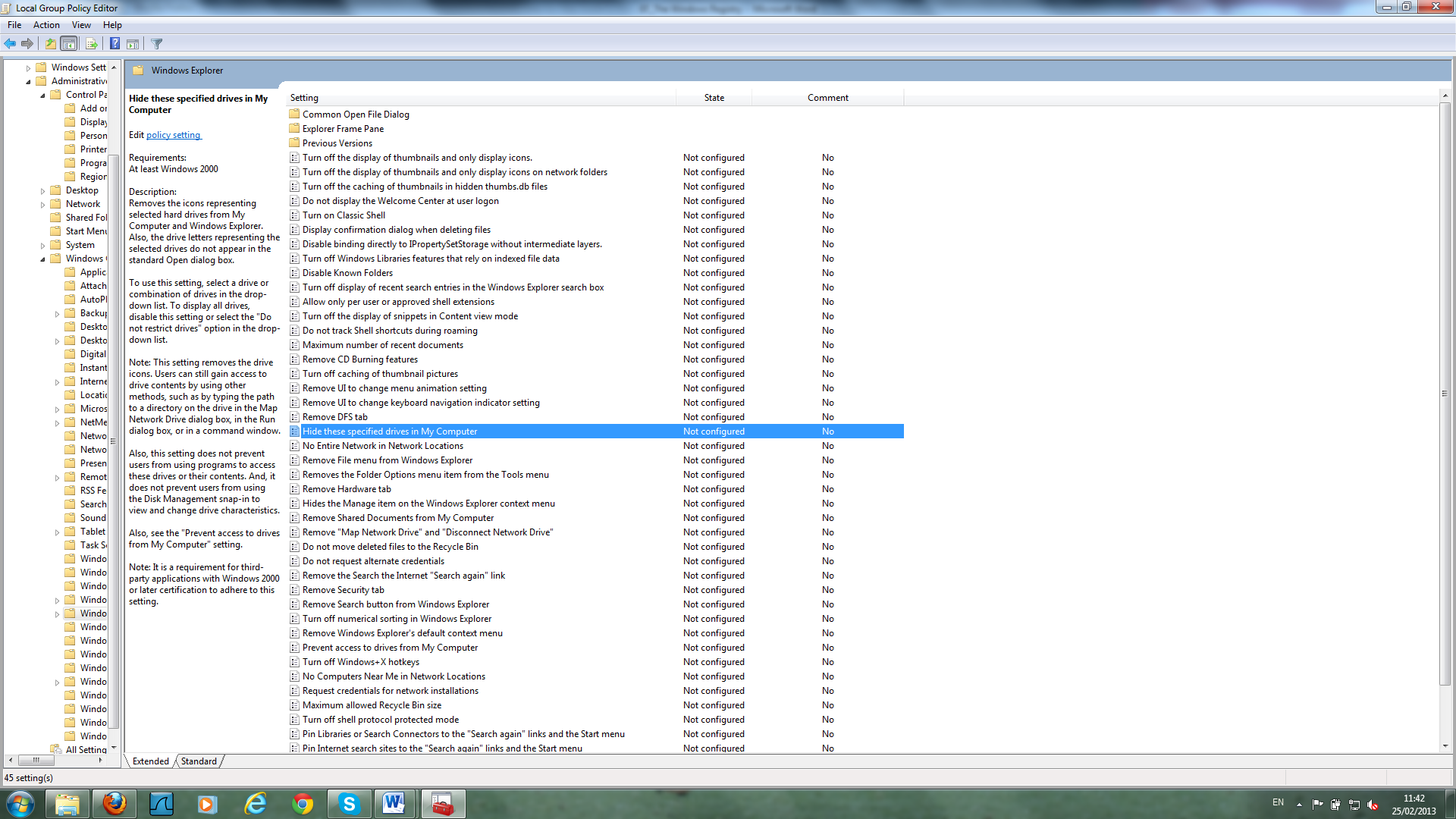
Try to run the Registry Editor and you should get a message as follows:



**How to hide Drives on your computer**

**Navigate to User Configuration > Administrative Templates > Windows Component > Windows Explorer**. Now look for the **Hide these specified drives in My Computer** setting. Double-click on it and Enable it, choose to hide the CD-ROM drive.

When done, open Windows Explorer, and try to see the CD-ROM drive.



**How to remove Change Password from Ctrl+Alt+Del Options**

**Navigate to User Configuration > Administrative Templates > System > Ctrl-Alt-Del Options** and **Enable the Remove Change Password Setting.** Test that the changes have been applied.

**How to restrict access to the Control Panel**

**Navigate to User Configuration > Administrative Templates > Control Panel. Enable the Prohibit access to the Control Panel setting.** This disables all Control Panel programs (prevents Control.exe from starting). Removes Control Panel from the Start Menu. Removes the Control Panel folder from Windows Explorer. If a user tries to select a Control Panel item from the Properties item on a context menu, a message like the following will appear. **Try right-clicking on the Desktop and Personalise**.

