

How the Windows Registry **Works**

[Home](#)[Site Map](#)[Products](#)[Downloads](#)[Newsletters](#)[Purchase](#)[Support](#)[Glossary](#)[Feed Back](#)

Here you'll find so basic explanation on how the Windows Registry **works**

[Explanation of Registry Hives](#)
[How Windows uses the Registry](#)
[Explanation of SubKeys](#)
[The Heart of Windows](#)
[More about Keys and SubKeys](#)
[Where are the Windows Registry files?](#)

Where are the Windows Registry files?

If you have Windows 9x or ME, known as the 9x Platform, the Registry files are stored in the Windows folder. They are named System.dat and User.dat. However, Windows ME also has Classes.dat. These files all have Hidden attributes so unless you are view hidden files you will not find them.

If you have Windows NT, 2000, 2003, Xp or Vista, known as the NT Platform, the Registry files are stored in their own folder; Windows\System32\Config. You will not be able to copy these files to make backups, as the operating system is protecting them for security reasons.

Lets start with an explanation of the Registry Keys.

Open **RegEdit**, you'll see the six Hives on the Windows 9x Platform and five on the NT Platform under the "My Computer" header:

HKEY_CLASSES_ROOT: Contains software settings about the file system, it contains shortcut information, and other user interface information. There will be a SubKey for every file association. And each Key here points to another key. The entire Hive is part of **HKEY_LOCAL_MACHINE** and can be found at **HKEY_LOCAL_MACHINE\Software\Classes**. If you change a setting in either of these two locations it is also changed in the other.

HKEY_CURRENT_USER: Contains the information for the currently logged-on user, such as settings and software information. Setting changed in this Hive will only affect the current user. This Hive is part of the **HKEY_USERS** hive.

- 1) AppEvents Key: contains the settings for which sounds to play for system sound events
- 2) Control Panel Key: Control Panel settings are stored here, similar to system.ini and win.ini in Win 3.x.

- 3) InstallLocationsMRU: contains folder paths and drives
 - 4) Keyboard layout: specifies current keyboard layout
 - 5) Network: Network connection information
 - 6) RemoteAccess: Contains information about the current log-on location using Dial-Up Networking
 - 7) Software: Software configuration settings for the currently logged-on user.
- You may find other Keys here placed by other software, that is probably should have been placed in one of the above Sub-Keys. The entire Hive is also found at the HKEY_USERS\Default or if more then one profile HKEY_USERS\(Profile name). If you change and setting in either of these two locations it is also changed in the other.

HKEY_LOCAL_MACHINE: Contains information about the hardware and software settings that are used for all users of this computer.

- 1) Config: Configuration information . Same as the Hive HKEY_CURRENT_CONFIG on Windows 9x
- 2) Enum: Hardware information (found under System in NT)
- 3) Hardware: Information passed to Windows from the BIOS (found under System in NT)
- 4) Network: information about networks installed to the machine.
- 5) Security: network security settings.
- 6) Software: Software-specific information and settings
- 7) System: System startup and device driver information, and operating system settings.

HKEY_USERS: Information about for each user that logs onto this computer is stored here. Each user will have a Sub-Key under this heading. On Windows 9x, if there is only one user, the SubKey will be ".default". When a user logs on, one of the Sub-Keys will be loaded to the HKEY_CURRENT_USER key.

HKEY_CURRENT_CONFIG : Contains info about the current hardware configuration, pointing to HKEY_LOCAL_MACHINE\Config. This hive is dynamic, meaning it is built on the fly.

HKEY_DYN_DATA: This key contains dynamic information about plug-n-play devices. The data here changes constantly. This key is rewritten every time you boot up, it is a virtual Hive. This Hive is dynamic, meaning it is built on the fly and is not used on the NT Platform.

How Windows uses the Registry

For reference:

HKCU=HKEY_CURRENT_USER
HKLM=HKEY_LOCAL_MACHINE

First you need to understand that the Shell of Windows is the Windows Explorer. Without the Explorer there is no Windows, No desktop and so on.

With Windows loaded and you are logged on, Windows now can use the information in the Registry that each Hive contains. Here is a sample of how the Registry is used.

So let's click on the Start button, which is controlled by the Explorer.exe. Now Windows reads the Key HKCU\Microsoft\Windows\CurrentVersion\Policies Keys to determine what the current user is allowed to do. What the user is allowed to access. Now scroll up to the Programs label and the Key HKCU\Microsoft\Windows\CurrentVersion\Explorer is read to determine how Windows will react. Now you open the Start menu and it opens, then you click on a program you wish to open. This is a shortcut that you are clicking on. A shortcut is a file with the extension ".lnk" so Windows now looks at the HKLM\Software\Classes Key for the extension ".lnk" which reads "lnkfile". So Windows now looks for HKLM\Software\Classes\lnkfile which reads "Shortcut" and it sees "NeverShowEx" = "" which means do not display file extension for this type of file extension. Windows then looks to see what SubKeys are there and it finds a CLSID SubKey which points to "{00021401- 0000- 0000- C000- 0000000000046}". Since the SubKey was "CLSID" Windows knows to look at HKLM\Software\Classes\CLSID Key and finds the matching SubKey. This Key also reads "Shortcut" and Windows now looks for what SubKeys are available. It finds the "ProgID" Key which points back to the "lnkfile" Key. and it finds the server file at the "InProcServer32" file. So Windows now knows to serve this file to Shell32.dll.

If the Explorer did not find the CLSID key listed it would have to search the CLSID Keys for a matching ProgID Key that pointed to the "lnkfile" Key.

Shell32.dll now takes over, the Explorer makes an API call to the Shell32.dll and tells it to Shell (run, start) the file that the that you clicked on. Now the Shell32 knows that this is a Shortcut so it reads the file and determines that you wish to activate the program QikFix. So it starts QikFix now needs and starts loading the other DLLs it needs to run. QikFix searches its' own directory and if not found there looks to the Windows Folder then the System folder. It finds the DLLs it knows it needs to run, and then now sees that it needs an interface. So as with all Visual Basic programs it needs the Msvbvm50.Dll or Msvbvm60.Dll to do the work for it. So then Msvbvm50.dll draws the plain window and then starts adding the text boxes and the tabs. To do this it may need help from another dll as with the case of the tabs. It knows it needs tabctl32.ocx because when I selected to use the tabs I need to include the DLL name in my source code, which in this case is an Active X control (ocx). Now it needs to draw Tabs but the Msvbvm50.dll has no idea what a tab is.

If there were TypeLib Keys under the CLSID Key then the Explorer would also get read the matching HKLM\Software\Classes\TypeLib Key. The TypeLib Key may point to an Interface Key (HKLM\Software\Classes\Interface) and this Key will tell the Explorer what version is available and so on. To understand more you need to read the next section about SubKeys.

Explanation of SubKeys

You will find a vast amount of different SubKeys, some are user (read) only by the software program, while most of the SubKeys are read by Windows.

So for this example let's use the * Key. If you open **RegEdit** and go to the HKEY_CLASSES_ROOT Hive the first Key down will be the *. This Key is a wildcard Key for File extensions, all files. You will find a SubKey Shellex which should have no value set. The Shellex Key tells the Explorer what to do when you right click on a file. The context menu should be displayed when you right click a file, the Explorer will use the ContextMenuHandlers SubKey. You will also find a PropertySheetHandlers SubKey there also. This SubKey tells the Explorer what to display when you right click a file and select Properties.

My ContextMenuHandlers SubKey has three SubKeys, Open With, Quick View, and WinZip. I added the Open With SubKey so that I always have the option of the Open With when I right click a file. If you have the WinZip installed you'll have the WinZip SubKey, so let's follow the WinZip SubKey. It points to a CLSID Key, HKEY_CLASSES_ROOT\CLSID\ {E0D79304- 84BE- 11CE- 9641- 444553540000} Key, and this Key has the SubKey InProcServer32 which points to WZSHLSTB.DLL. So Windows now knows what DLL to load if you click on WinZip in the context menu.

Now let's open **RegEdit** (so you can follow) and go to HKEY_CLASSES_ROOT\.gif, then right click on a file we all have, C:\Windows\Cloud.gif, this is a standard Windows file. The .gif Key does have a Shellex SubKey, but not a ContextMenuHandler SubKey, so only the default context menu will be displayed. It does have the SubKey {BB2E617C- 0920- 11d1- 9A0B- 00C04FC2D6C1} which points to the CLSID\{3F30C968- 480A- 4C6C- 862D- EFC0897BB84B} Key which is now telling the Explorer if Thumb nails are use what to do, what file will handle this operation.

Now right click the Cloud.gif file and select Open. You will notice that the (Default) value of the .gif Key points to giffile. Windows will now go to HKEY_CLASSES_ROOT\giffile to see what it must do. The (Default) value of this Key is GIF Image, this is what is displayed in the Explorer under the "Type" column if you view files in Detail. The giffile Key has 3 SubKeys, CLSID, DefaultIcon, and Shell. The DefaultIcon SubKey tells the Explorer what Icon should be displayed for a .gif file. The Shell SubKey should have 3 SubKeys, Open, Print, and Printto. Since we select Open on the Context Menu which would be the same as double clicking on the file, Windows will use the Shell\Open subkey. The Shell\Open Key has one or two SubKeys Command, and maybe a ddeexec SubKey. If you have just the Command Key, Windows stops there and shells (starts - run) the file listed at the (Default) value of the Shell\Open\Command SubKey with the parameter (command line) C:\Windows\Cloud.gif . This will tell the program, possibly Fast View to display the file Cloud.gif.

If there is no ddeexec SubKey the (Default) value would look something like this.
"C:\Easy Desk Utilities\Fast View\Fastview.exe" %1. The %1 means to pass the file path and name only on to Fast View. This would be the same as typing in the Run box at the Start menu C:\EasyDe~1\FastVi~1\Fastview.exe C:\Windows\Cloud.gif. The Run box needs to have 8.3 format. Now Fast View is displaying the Cloud.gif file.

If you have a ddeexec SubKey the (Default) value might be :
"rundll32.exe C:\WINDOWS\SYSTEM\SHIMGVW.DLL ,ImageView_Fullscreen "
and ddeexec SubKey value is "[open(%1)]", In the Run box you would type:
rundll32.exe C:\WINDOWS\SYSTEM\SHIMGVW.DLL, ImageView_Fullscreen
C:\Windows\Cloud.gif

The Heart of Windows

The Registry **works** the same on both the Windows 9x platform, and the NT Platform.
However it is laid out a little differently, especially when it comes to the hardware.

The real heart of the Windows Registry is found at the Key
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows, controlling the software.
The System Drivers are controlled at HKEY_LOCAL_MACHINE\System. The
hardware is controlled at HKEY_LOCAL_MACHINE\Enum in Windows 9x and ME
And HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum in the NT Platform.

[About Us](#) | [Privacy Policy](#) | [Contact Us](#) | [FAQ](#) | [Help Files](#) | [Humor](#) | ©2010 Easy Desk
Software