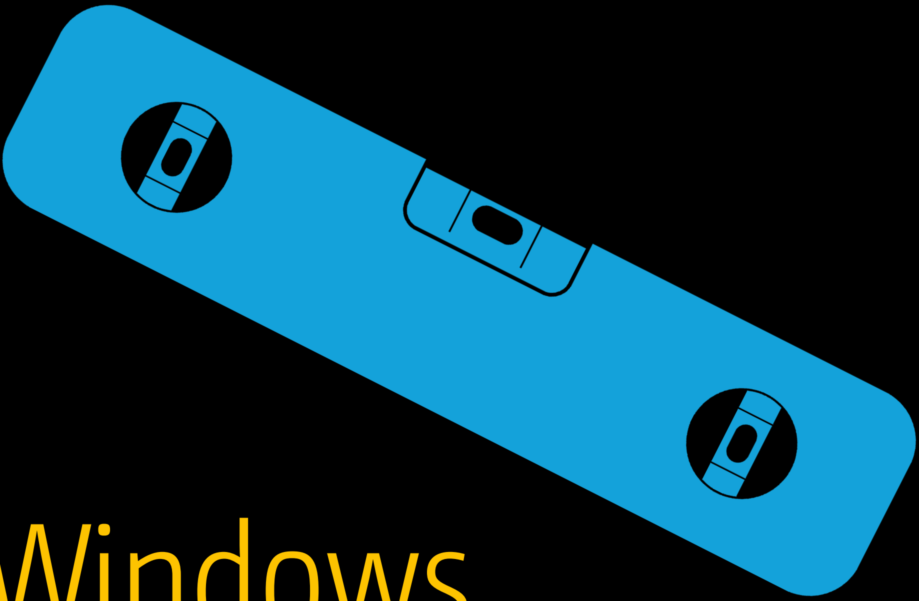
WINDOWS TROUBLESHOOTING SERIES



Windows Registry Troubleshooting

Diagnose and repair problems with any Windows version from Vista to Windows 8.1

—

Mike Halsey, MVP
Andrew Bettany, MVP

Apress®

For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.

friendsof ™

Apress®

Contents at a Glance

About the Authors xiii

About the Technical Reviewer xv

Introduction xvii

■ Chapter 1: The Architecture of the Windows Registry 1

■ Chapter 2: Registry Tools and Utilities 11

■ Chapter 3: Troubleshooting a Corrupted Registry 33

■ Chapter 4: Working with Other Users' Registry Files/Advanced Troubleshooting 51

■ Chapter 5: Securing the Registry 73

■ Chapter 6: Hacks, Tweaks, and Common Changes 93

Index 111

Introduction

Whatever you do on your Windows PC, from installing software and apps, to using GPS hardware, creating 3D models, or intricate embedded security systems and games, the Registry is at the heart of what you do.

Consequently, it's the one common element that's guaranteed to run through all troubleshooting problems. This makes a good knowledge of what the Registry is, how it works, and how you can work with it, an essential skill for any support technician or IT pro.

That's where this book comes in. There have been other Windows troubleshooting books, but none have gone into as much detail about the Registry as you'll find here.

It's not all dry technical code and complex structures, though. There are also a lot of fun and cool things you can do with the Registry to enhance your productivity and enjoyment of using your PC. In this book we've included over 60 of our favorite hints, tips, and tweaks for Windows Vista, Windows 7, and Windows 8.1.

With a good knowledge of the Registry—and there's no better way to learn than to dive in with some hacks—you'll be in a much better position to quickly diagnose and repair Windows and PC problems as you encounter them.

CHAPTER 1

The Architecture of the Windows Registry

The most relevant simile for the Windows Registry in your PC is the human brain It stores all of the key information and code that allows the different parts of your computer, and its operating system and software, to communicate with one another. This includes details about hardware addresses and communications protocols, software installation and configuration files and, of course, the inner workings of Windows itself.

No matter what version of Windows you are using, you will have not one but many registry files, as Windows splits the registry into several main files, which contain all the information relevant to the overall operation of the computer and its hardware, and other files specific to each individual user and their software, settings, and preferences.

Registry files are databases, often large and complex, that the operating system loads into the PC's memory as the computer starts or the user signs in. Without this readily available store of settings, files, and options, nothing can start or operate on the PC at all.

The Registry is similar to the brain in another way, too. It's able to heal itself, but only to a limited extent. While the human body is full of all the building blocks required to repair and rebuild damaged cells and tissue, the brain, like the spinal cord, mostly lacks this self-repair ability. Should the wrong part of the brain is damaged, it simply would be unable to repair itself In the case of the Registry, should the parts of the databases containing the information required to load the Registry Editor, any third-party repair tools, or the System Recovery Options become corrupt, self-repair will be out of the question.

Fortunately, however, the Registry is unlike the human brain in one significant way: you can rebuild and repair it from the outside, and even reset it to an earlier state or wipe it completely and start from scratch. While the prospect of a surgeon or doctor performing such an operation on your own mind might seem disturbing at the least, this book will guide you through the Windows Registry in depth, and show you how to protect configure, and repair your PC's brain under any circumstances.

What Is the Windows Registry?

If, like me, you used Windows 3.1 back in the day, you might remember having to manually open a `.ini` file in the folder for a program or driver and changing or configuring settings that would enable that program or piece of hardware to work.

Every program and hardware driver had its own `.ini` file, which was a plain ASCII text document, and each one needed configuring individually for each individual PC. This is because there were always minor differences between PCs, such as different drive letter or serial port assignments, so if you imported a `.ini` file from one Windows 3.1 PC to another it was unlikely to work.

The Registry was Microsoft's answer to this chaos, and it effectively pulled all these individual `.ini` files together into a single, manageable database.

The Registry as we know it now was first introduced in Windows 95 and Windows NT, and as a core component of the operating system it hasn't changed much since. The reason for this is compatibility with legacy hardware and software is crucial to business users of PCs, and changing the Registry too much would either break everything, or require a complex virtualization engine, effectively running a PC inside a PC (which would present significant security implications if overall performance wasn't to be affected as well).

Over time the Registry has been made more robust and secure, with features such as User Account Control (UAC) preventing malware from writing itself deep into the folder structure where it might never be found.

Despite its size and the tens or even hundreds of thousands of keys it contains, it's quite a straightforward, simple, and robust system to work with, consisting of a fairly simple structure of folders, keys, and values.

Windows Registry Files

If you are using a *modern* version of Windows—by which I mean Vista, Windows 7, Windows 8 or 8.1, and Windows 10—then the Registry, how it is structured and operates, and how you can edit and work with it will be the same. As a core component of the operating system, it doesn't change very often, though there were slight differences in Windows XP and Windows Server 2003 that, where applicable, we will detail in notes throughout this book.

Windows has several stores for Registry files, one for the settings that are relevant to all users on the PC, such as hardware and general operating system options, and more for each individual user.

The main registry files are located in the `%systemroot%\System 32\Config\` folder (see Figure 1-1) and consist of the following:

- `SAM` (Security Accounts Manager)
- `SECURITY`
- `SOFTWARE`

- SYSTEM
- DEFAULT
- USERDIFF (used only for OS upgrades)

Additionally, each user will have their own Registry files:

- %userprofile%\ntuser.dat
- %userprofile%\AppData\Local\Microsoft\Windows\UsrClass.dat

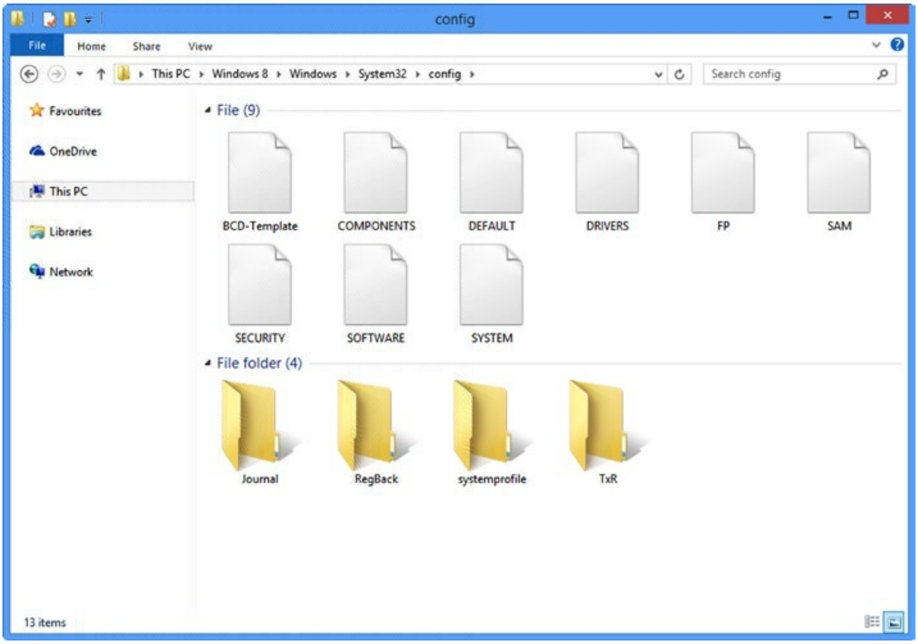


Figure 1-1. The main Windows Registry files

The first of these files, ntuser.dat, contains the user's general software, customization, and configuration options. The second, UsrClass.dat, contains additional settings such as user-specific file associations and COM (Component Object Model) information.

■ **Note** If you are still using Windows XP or Windows Server 2003, please stop! However, the UsrClass.dat file can be found in the %USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\ folder.

The Registry files themselves are binary format files, and as such are not readable without interpreter software. This is different from `.reg` files, which are exportable from the Windows Registry Editor and are plain text, so as to be viewable and editable by administrators.

Registry Keys and Values

The Registry, when viewed in its entirety, is split into five main sections, or groups. This helps organize the different keys and folder in the Registry to make them easily navigable.

When you modify or create items in the Registry you will create both keys and values, but what are the differences between the two?

■ **Note** Registry keys are containers that will contain values for software, hardware or the Windows OS itself. Keys can be thought of as folders, in which the values are individual files. You can navigate the keys using a folder-type hierarchy. Values are the nuts and bolts of the Registry. These are the settings, integers, and strings that provide the data and information that helps Windows load and effectively work with your hardware and apps.

HKEY_CLASSES_ROOT (HKCR)

This section stores information about registered applications, OLE Object Class IDs, and file associations. Sometimes you may see it abbreviated to `HKCR`.

Should a subkey be added that duplicates a key that already exists in `HKEY_CURRENT_USER\Software\Classes`, Windows will use the `HKEY_CURRENT_USER\Software\Classes` key as its master.

HKEY_CURRENT_USER (HKCU)

This section contains the configuration options for the currently signed-in user's profile, including the disk locations for user folders, control panel settings, and specific app configuration settings.

HKEY_LOCAL_MACHINE (HKLM)

As you might expect, this key contains all the settings specific to the PC on which Windows is installed. It contains the subkey files `SAM`, `SECURITY`, `SYSTEM`, and `SOFTWARE` that I listed earlier. A fifth file, `HARDWARE`, is created afresh each time the PC starts, and contains information about currently detected plug-and-play hardware.

This key also contains the files `COMPONENTS` and `BCD`, which can be found in the `%systemroot%\System 32\Config\` folder, and which contain information specific to the PC's boot configuration data.

A few of the subkeys in the HKEY_LOCAL_MACHINE section are worthy of note, because this collection of Registry files is most commonly modified and edited by users.

SAM

The Security Accounts Manager keys section will commonly appear empty unless the viewing user has all appropriate administrative permissions. It contains the security information for all the domains the PC connects to, including the local domain, which is also called SAM.

Each SAM database contains the username used to log in to the domain, along with a UID (Unique Identifier) for the domain, a cryptographic hash of the user's password, the location of the user's Registry Hive, on the server, and various other settings and flags that may be required by the connection.

SECURITY

This subkey also appears empty for most users unless they have appropriate administrative permissions. Upon connection to a domain, it is linked to the security database in the Server Registry Hive, which will contain all security policies applicable to the current user and their installed apps.

SYSTEM

This subkey contains information about Windows setup, settings, and details for currently mounted devices, and for drives that contain a file system.

SOFTWARE

This subkey contains settings for the currently installed Windows installation and its installed software and apps. The keys are organized by software vendor and include subkeys for file extensions, MIME types, and Object Class and Interface IDs (such as ActiveX controls).

HKEY_USERS (HKU)

The HKU key contains settings and options for the currently-loaded user profile. You cannot access settings for any user account other than the currently signed-in user through these keys.

HKEY_CURRENT_CONFIG (HKCC)

Last, the HKCC keys contain information gathered when the PC starts and Windows loads. This information is relevant only to the current powered-session and is discarded when the PC is switched off.

Within each of the Registry sections, the individual keys include many different types, to allow the greatest flexibility for managing settings and options for the Windows OS and your apps and hardware.

HKEY_PERFORMANCE_DATA

This section is invisible when using the Windows Registry Editor. It contains runtime and performance data that is provided by the Windows kernel, drivers, apps, and services. It is discarded when the PC is switched off and is rebuilt at the next boot.

Registry Value Types

When you create a new Registry value, you will be presented with the following options:

- `REG_BINARY`: This key type stores raw binary data.
- `REG_DWORD`: A variable-length 32-bit integer.
- `DWORDS`: Commonly used to define the parameters for settings, device drivers, and software configuration.
- `REG_SZ`: A fixed-length string value.
- `REG_EXPAND_SZ`: An expandable-length string value, also used to contain environment variables.
- `REG_MULTI_SZ`: A multiple string that may contain a list of values, normally separated by commas or a space.
- `REG_RESOURCE_LIST`: A list of resources in a nested array, used by device drivers.
- `REG_RESOURCE_REQUIRMENTS_LIST`: An array list of hardware resources, used by device drivers.
- `REG_FULL_RESOURCE_DESCRIPTOR`: These are nested arrays used to store the resource lists for physical hardware devices.
- `REG_LINK`: A symbolic link (UNICODE) to another Registry key that specifies both the root key and the path to the target key.
- `REG_NONE`: Data that does not have a specific type.
- `REG_QWORD`: A variable-length 64-bit integer.

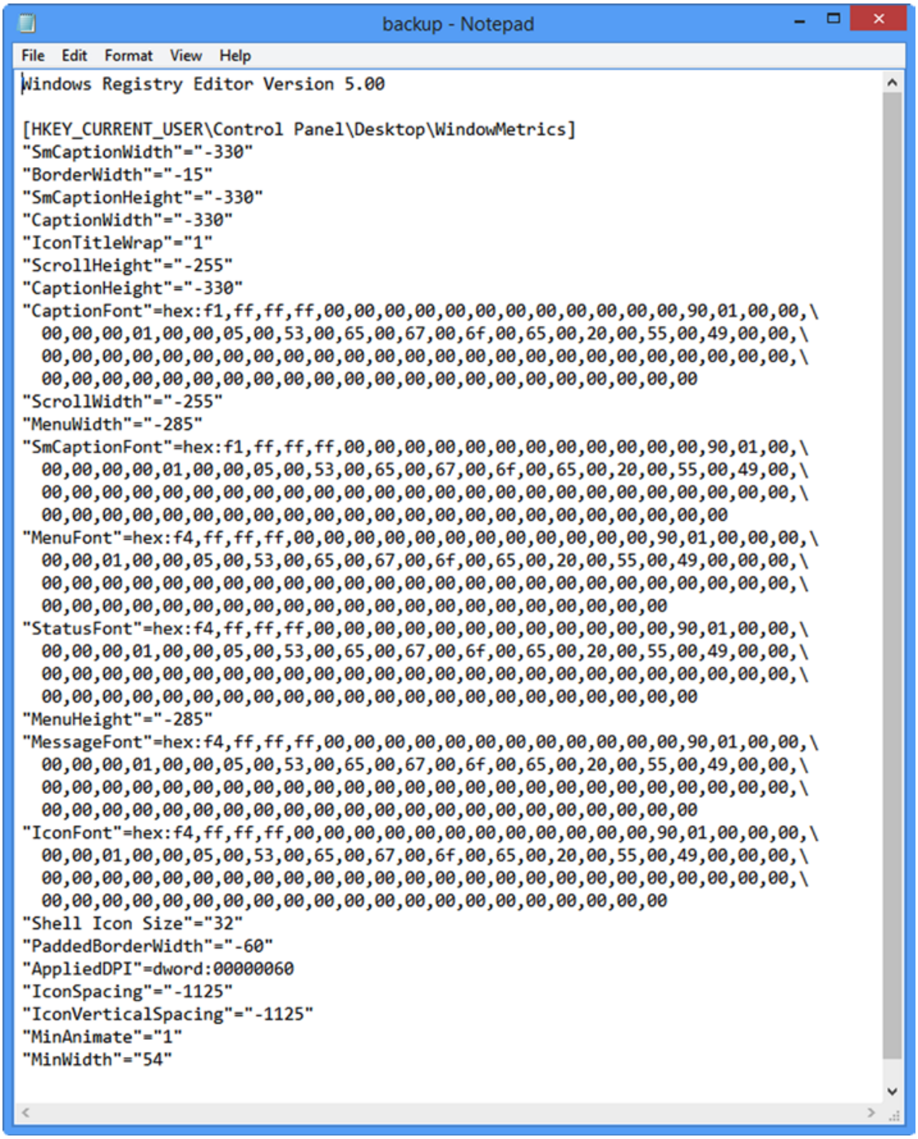
■ **Note** There are some differences between 32-bit (x86) and 64-bit (x64) Windows Registries. The 64-bit `QWORD`s are not supported by 32-bit versions of the Windows operating system. Also, the Registry handles 32-bit and 64-bit keys in the same intelligent way that the file system handles multiple versions of the same `DLL` files, but maintains a compatibility folder that you can find in the `HKEY_LOCAL_MACHINE\Software\WOW6432Node` section of the Registry.

.reg Files

Earlier in the chapter I made a reference to `.reg` files, which are different from the Registry in several important ways. First, while they store Registry keys and values, they are intended for backup and transport between different PCs and Windows installations of those values and keys. Also, `.reg` files are stored in plain ASCII text, unlike the Registry database files, which are stored in binary.

`.reg` files will be automatically opened (or at least they should be if the file associations are set correctly on a PC) by the Windows Registry Editor when double-clicked. You can also export the entire Registry, or individual Registry keys, to `.reg` files in the Registry Editor itself; we will show you how to perform these actions in Chapter 2.

You can manually edit the contents of a `.reg` file in Windows Notepad, or any other text file editor (see Figure 1-2). Right-clicking a `.reg` file in File Explorer (known as Windows Explorer in Vista and Windows 7) will reveal an Edit option in the context menu that appears.



```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Control Panel\Desktop\WindowMetrics]
"SmCaptionWidth"="-330"
"BorderWidth"="-15"
"SmCaptionHeight"="-330"
"CaptionWidth"="-330"
"IconTitleWrap"="1"
"ScrollHeight"="-255"
"CaptionHeight"="-330"
"CaptionFont"=hex:f1,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,90,01,00,00,\
  00,00,00,01,00,00,05,00,53,00,65,00,67,00,6f,00,65,00,20,00,55,00,49,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
"ScrollWidth"="-255"
"MenuWidth"="-285"
"SmCaptionFont"=hex:f1,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,90,01,00,\
  00,00,00,00,01,00,00,05,00,53,00,65,00,67,00,6f,00,65,00,20,00,55,00,49,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
"MenuFont"=hex:f4,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,90,01,00,00,00,\
  00,00,01,00,00,05,00,53,00,65,00,67,00,6f,00,65,00,20,00,55,00,49,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
"StatusFont"=hex:f4,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,90,01,00,00,\
  00,00,00,01,00,00,05,00,53,00,65,00,67,00,6f,00,65,00,20,00,55,00,49,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
"MenuHeight"="-285"
"MessageFont"=hex:f4,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,90,01,00,00,\
  00,00,00,01,00,00,05,00,53,00,65,00,67,00,6f,00,65,00,20,00,55,00,49,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
"IconFont"=hex:f4,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,90,01,00,00,00,\
  00,00,01,00,00,05,00,53,00,65,00,67,00,6f,00,65,00,20,00,55,00,49,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
"Shell Icon Size"="32"
"PaddedBorderWidth"="-60"
"AppliedDPI"=dword:00000060
"IconSpacing"="-1125"
"IconVerticalSpacing"="-1125"
"MinAnimate"="1"
"MinWidth"="54"
```

Figure 1-2. *Editing the contents of a .reg file in Notepad*

In the example seen in Figure 1-2 the key is listed as `HKEY_CURRENT_USER\Control Panel\Desktop\WindowMetrics`, and I have highlighted it here as it includes an addition to the default configuration.

At the end of the key sits a subkey called `"MinWidth"` that has an integer value of 54. This key changes the behavior of the Never Combine, Hide Labels option for the Windows Taskbar (not Vista) so that program icons appear separately when multiple instances of that program are run, but without their text labels.

We will include many more hints, tips, hacks, and tweaks for the Windows Registry in Chapter 6.

Summary

Overall, while the Windows Registry is a huge behemoth bloated with many thousands of complex binary, hexadecimal, and ASCII values and codes, it's fairly simple and straightforward to work with. In the next chapter, we'll look how you can do exactly that, and what tools and utilities exist to help you.