Understanding The Registry

Editing the registry is not as hard as you might think, but you need to understand what you're doing, and it's essential to make a backup before you make any changes so that you can back them out if necessary.

By Mike Lewis

Put simply, the Windows registry is a central repository of information about all aspects of the computer - in particular, its hardware, operating system, applications and users. It can be accessed and updated under software control and also directly by users.

The registry first appeared in Windows 3.1. In that system it was a single file, called REG.DAT, and was mainly used to store information about OLE objects. Most other configuration data was held in various INI files, of which WIN.INI and SYSTEM.INI were the most important.

The modern registry, as found in Windows 9x and NT, brings together all the information that was previously held in REG.DAT and the separate INI files.

The registry has several advantages over INI files. Because the information is centralised, it is easier for applications to access it. It is more hierarchical than INI files, and so better suited for storing large amounts of structured data. It is also free of the size limitations which affect INI files (although there is still a maximum total registry size limit).

Storage

Although the registry is usually considered to be a single entity, its contents are in fact stored in more than one physical file. In Windows 9x, there are two such files: SYSTEM.DAT and USER.DAT. These hold computer-specific and user-specific information respectively. In Windows NT, the registry is spread over a series of files, sometimes called hives.

SYSTEM.DAT and USER.DAT are usually held in the Windows directory. However, it is also possible to place USER.DAT in the user's login directory on a network, thus allowing the user to log in at other workstations. In NT, the hive files are located in the SYSTEM32\CONFIG directory, which is off the Windows directory.

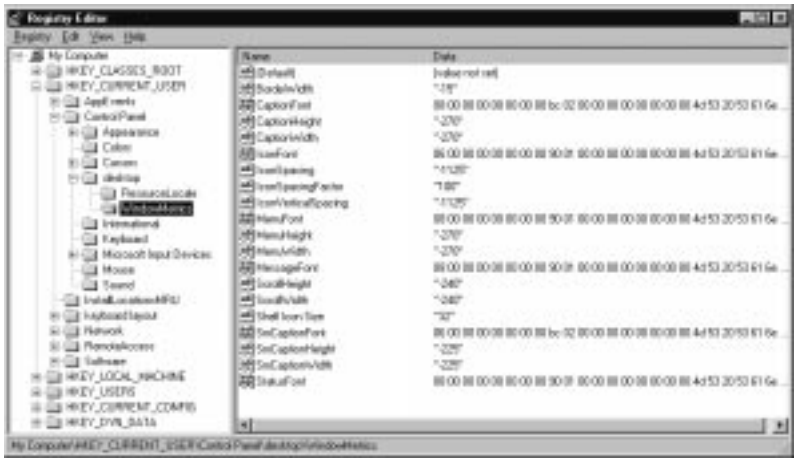

Figure 1 - The Microsoft Registry Editor shows the registry's hierarchical structure.

Architecture

When you view the registry in the Microsoft Registry Editor its hierarchical nature becomes obvious. (To launch the editor, run REGEDIT.EXE from the Start/Run menu. I'll describe it in more detail later in the article.) The editor presents an Explorer-like view of the registry, with a tree in the left pane and data in the right (see Figure 1).

The registry tree is divided into six broad sections (five in NT). These sections, which all have names beginning with HKEY_, are called root keys or top-level keys (see Figure 2). Each root key contains sub-keys, which might in turn contain further sub-keys and so on. The lowest level keys along a given branch are called values.

Taking a hard disk as an analogy, keys are like directories and values are like files. Keys and values both hold data, which can either be binary values or ASCII strings.

Each item of data has an associated name. As a minimum, each key and value holds a single data item, named Default. The data and names are displayed in the right pane of the editor,

along with an icon which shows whether the data is binary or string.

Continuing with the hard disk analogy, you can identify any key or value by specifying the path along its branch, using the familiar backslash notation. For example, information about installed dial-up networking connections is held in HKEY_CURRENT_USER\RemoteAccess\Addresses. If you drill down through this path in the left pane, you will see the relevant data in the right pane. In this example, each data item corresponds to one DUN connection.

Aliases

I said earlier that the registry is divided into six broad sections, one for each root key. This is certainly how the registry is usually regarded, but it is not strictly true. The reason is that all but two of the root keys are in fact aliases for other parts of the tree.

To see an example of this, drill down from HKEY_CLASSES_ROOT. You will see that this root key contains a large number - perhaps many hundreds - of sub-keys at the first level down. The first group of these sub-keys have names which look like file extensions, while the names of the remainder resemble those of applications.

Now locate HKEY_LOCAL_MACHINE\Software\Classes. As you can see, this contains exactly the same sub-keys, values and data as HKEY_CLASSES_ROOT. That's because HKEY_CLASSES_ROOT is an alias for HKEY_LOCAL_MACHINE\Software\Classes.

```
HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS
HKEY_CURRENT_CONFIG
HKEY_DYN_DATA
```

Figure 2 - The six root keys.

An alias is not a copy. Rather, it is another view of the same information. If you edit the data in the alias, the change is immediately reflected in the part of the tree to which the alias refers, and vice versa. Only one edit actually takes place, but you are seeing it from two different viewpoints. Figure 3 lists the aliases in the Windows 9x registry.

One of the root keys, HKEY_DYN_DATA, works slightly differently. This key is essentially a RAM-resident copy of certain parts of the registry which Windows needs to get at quickly. It is created at boot time and discarded at shut-down; it never gets written back to disk.

Because aliases only exist while Windows is running, they will not get backed up if you create your backup copies from DOS. This is not a problem as the information in the aliases is all available elsewhere in the registry. Windows always re-creates the aliases during startup.

Registry Editors

The main tool for viewing and editing the registry is the Microsoft Registry Editor, REGEDIT.EXE. Although third-party editors exist, you will probably want to stick with the official Microsoft product, given the critical nature of the registry editing process. (That's not to say that REGEDIT.EXE is itself completely reliable; the Microsoft Knowledge Base notes several bugs in the Windows 95 version, but these are unlikely to cause problems in day-to-day operations.)

Windows NT 4.0 comes with a second editor: REGEDT32.EXE. This supports certain NT-specific features which REGEDIT.EXE does not know about, such as the ability to maintain security settings. However, it lacks the very useful search function found in the standard version. NT 4.0 also includes REGEDIT.EXE, although this might not be the same as the one found in Windows 9x. If you upgraded from Windows 3.1 to Windows NT, you will have the original 3.1 version of REGEDIT.EXE.

As far as the Windows 9x version is concerned, its operation is completely straightforward, with all its functions being easily accessible from the registry and Edit menus. You can also right-click on an item to edit, delete or rename it, or to create new keys or values.

When you edit a data item in the editor, the change is written to the registry almost immediately - you do not explicitly save the file. If you make a mistake, the only recourse (apart from restoring from a backup) is to edit the same item again.

Conversely, if another process changes a registry item while the editor is open, the editor will pick up the new setting straight away - although you might need to refresh the display in order to see it (to do so, select View, Refresh, or press F5).

Remote Registries

As well as letting you view and edit the registry on your local machine, the Microsoft Registry Editor can also access registries on other computers on the network. If your machine and the remote computer are both running NT 4.0, this operation is completely straightforward. But if either or both machines have Windows 9x, you must first install the Remote Registry service, which in turn depends on having user-level security enabled and Remote Administration services installed. For step-by-step instructions on setting this up, see Article Q141460 in the Microsoft Knowledge Base.

Once you have installed the necessary components, you can access the other computer's registry by selecting Connect Network Registry from the

Root key	Alias for
HKEY_CLASSES_ROOT	HKEY_LOCAL_MACHINE\Software\Classes
HKEY_CURRENT_USER	User's branch within HKEY_USERS
HKEY_CURRENT_CONFIG	Hardware profile within HKEY_LOCAL_MACHINE\Config

Figure 3 - Aliases on the Windows 9x registry.

The Registry

registry menu within the editor. Having done so, you will be able to view and edit the remote registry in the same way as the local registry. When you have finished, go back to the registry menu and select Disconnect Network Registry.

Registry Backup

Backing up the Windows registry presents a specific problem: you cannot directly copy the relevant files while they are open, and they are always open while Windows is running. However, there are a couple of techniques you can use to work round this.

Backup Utilities

For Windows 95 users, the easiest approach is to use the Configuration Backup utility (Figure 4). This copies the registry to a compressed backup file, the name of which is REGBACKn.RBK, where n is a sequence number. Up to nine generations of backup can be made. You are prompted to enter a description for the backup to help you subsequently identify it. The backup is always created in the Windows directory, but you are free to move it elsewhere.

The same utility can be used to restore and delete backups. It can only restore from the Windows directory so, if you have moved the file to another directory, you must move it back before running the utility.

The Configuration Backup utility is not installed by default. You will find it on the Windows CD-ROM, in the \OTHER\MISC\CFGBACK directory. You can copy the two files (CFGBACK.EXE and a help file) from this directory to your hard disk, or you can run the executable directly from the CD-ROM.

In Windows 98, the best way of backing up the registry is to use the Registry Checker (SCANREGW.EXE). This creates a backup automatically each time the computer starts, but it can also be run on demand. The backup is held in a CAB file, named RBn.CAB (where n is a sequence number), in the SYSBCKUP directory (this is a hidden directory off the Windows directory). By default, five generations of backup are maintained, but this number can be varied by editing SCANREG.INI.

Windows NT does not include a specific registry backup tool. However, the standard NT backup utility, NTBACKUP.EXE, is able to back up the registry, but only to supported tape drives.

Manual Backups

Another way of backing up the registry is simply to copy the relevant files. You cannot do this while Windows is running but, in the case of Windows 9x, you can work round this either by booting to DOS (hold down F8 during startup, then select Command Prompt Only) or by exiting to DOS from the Shut Down dialog.

The two registry files, SYSTEM.DAT and USER.DAT, are flagged as hidden, system and read-only. Before copying them, you will need to use the ATTRIB command to switch off these flags. Once that's done, you can copy the two files from the Windows directory to another suitable location. Finally, use ATTRIB again to restore the flags.

In the case of NT, if the system is configured for dual-booting you should boot to DOS or Windows 9x before copying the registry files. Alternatively, boot to DOS from a startup floppy. The files which you should copy are those stored in the SYSTEM32\CONFIG directory, which is off the Windows directory. Note that you cannot use this method if the Windows directory is on an NTFS partition, as the booted operating system will not be able to access it.

Whatever the operating system, you can restore the registry by reversing the above process.

Exporting The Registry

Another approach to backing up the registry is to export it. Exporting the registry is not the same as copying it. Instead, the process creates a text file which contains the registry data in a format similar to that of an INI file (see Figure 5). If you need to restore the registry, you can do so by re-importing the text file.

An advantage of this approach is that you do not have to export the en-

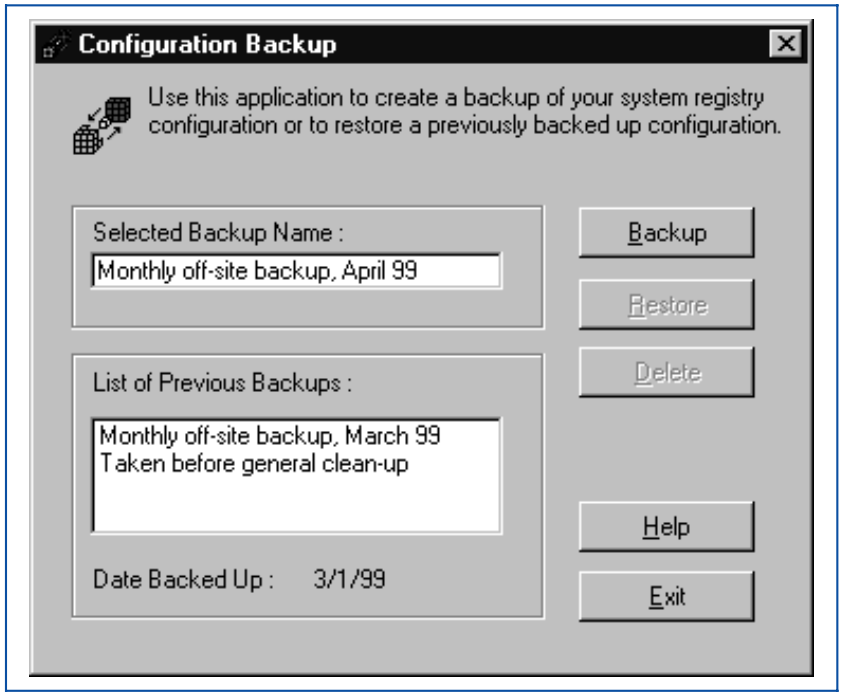

Figure 4 - The Configuration Backup tool provides the simplest way of backing up and restoring the registry in Windows 95.

tire registry. If you want to try out changes which only affect one branch, you can limit the export to that branch. Another benefit is that you can perform both the export and import operations from within Windows.

To start the export process, launch the Microsoft Registry Editor, select the branch that you wish to copy, choose Export Registry File from the Registry menu, and specify the name and location of the export file. Note that the Save dialog includes a choice between exporting the selected branch and exporting "all", that is, the whole registry.

The resulting file has the extension REG. You can view its contents by opening it in a text editor. When working with this file, take care not to double-click on it, as this will re-import it. You can also import the REG file by selecting Import Registry File from the Registry menu.

Automatic Backups

If the worst happens and you find yourself with a damaged registry and no recent backup, there is an escape route. As soon as Windows has successfully booted, it automatically creates a backup, which you can then use to restore the registry if the need arises. This is not always an ideal solution, as you can only restore the registry as it was at the start of the session, but it should be enough to get you out of trouble.

In Windows 98, these automatic backups are held in the CAB files created by the Registry Checker. If you need to restore from them, boot to DOS, then type SCANREG /RESTORE to launch the command-line version of the utility. You will see a list of the available backups, from which you can select the one you wish to restore.

The Registry Checker offers a couple of extra benefits. As its name suggests, it performs a check, albeit a rudimentary one, on the integrity of the registry. It does this at boot time. If it detects a problem, it will automatically restore the most recent backup. It will also defragment the registry if it detects more than half a megabyte of empty space.

In the case of Windows 95, only one generation of automatic backup is maintained. This consists of two files, named SYSTEM.DA0 and USER.DA0. They are hidden, system read-only files in the Windows directory. If you need to restore from them, boot to DOS, change the attributes (on the backup and the existing registry files), and copy the backups over the existing files. This will only work if you have not booted to Windows since the registry became corrupted.

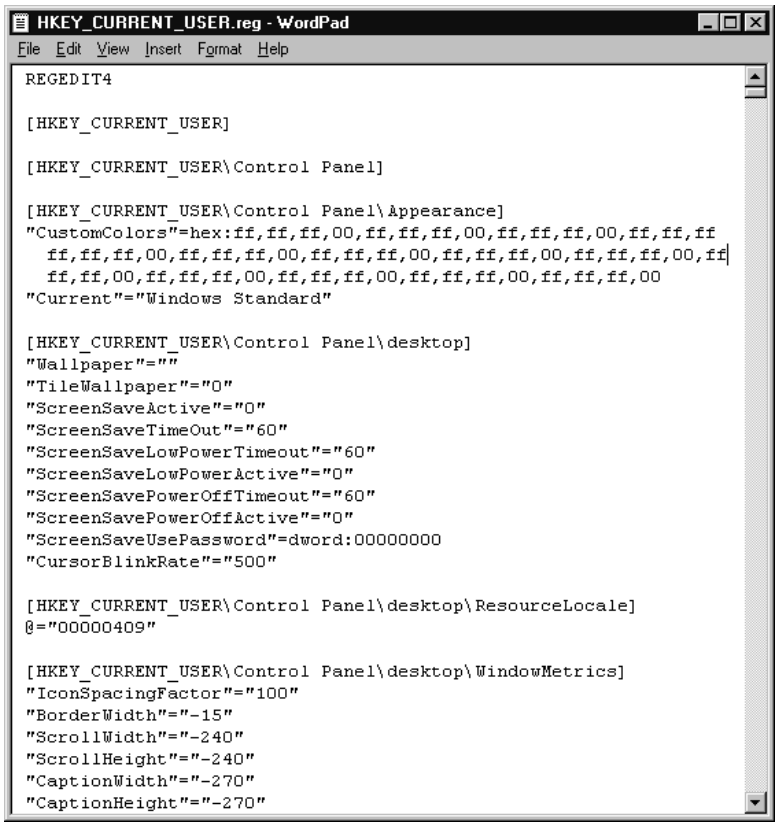

Figure 5 - You can export the registry to a text file, in INI format.

Registry Contents

For the remainder of the article, I will describe the most important keys and values in the registry of a typical PC. As you read this, you might want to follow along by having your own registry open in the editor. For convenience, I'll deal with the root keys in the order in which they appear in the editor.

HKEY_CLASSES_ROOT

This branch is an alias for HKEY_LOCAL_MACHINE\Software\Classes (see below), and is a direct descendant of the REG.DAT file found in Windows 3.1. It is mainly used to keep track of file extensions and their associated applications, documents and OLE objects. It is a particularly large branch, with a very large number of sub-keys at the first level down (I counted over a thousand on my own PC).

The first group of these first-level sub-keys have names that look like file extensions: .JPG, .XLS and the like. There is one of these for each "registered" document type, that is, for each type of file listed in the File Types tab in the Options dialog in Windows Explorer. As a minimum, the sub-key's data contains a reference to the class definition associated with the document.

The Registry

The class definitions themselves are held in the remaining first-level sub-keys. These contain a descriptive name for the document type (as it appears in the Type column in folder windows), a pointer to the default icon and, where relevant, information about how the application handles the documents as OLE objects and how the documents are manipulated from the Windows shell - for example, the actions available from the menu which appears when you right-click on the file.

Although HKEY_CLASSES_ROOT is updated automatically as applications are installed and uninstalled, there might be times when you need to edit it yourself. For example, you might want to restore a file association which a new application has taken over from an existing one. However, rather than editing the registry directly, it is easier and safer to make this type of change from the File Types tab in the Options dialog.

HKEY_CURRENT_USER

This root key contains information specific to the user, and is an alias for the user's branch within HKEY_USERS (described below). If user profiles are enabled, it relates to the user who is currently logged on. The key contains seven first-level sub-keys.

The first of the first-level sub-keys is named AppEvents, and contains details of the sounds which the user has associated with system or application events. It is organised into two subsidiary keys: EventLabels contains the names of the events, and Schemes contains references to the corresponding sound files. Schemes is itself organised by application, and for each event within the application there is a current and a default setting.

The second of the first-level sub-keys is named Control Panel. This contains the settings that used to be made from Control Panel in Windows 3.1: colour schemes, screen savers, wallpaper, keyboard repeat rate, mouse speed and so on. These settings are spread over a number of subsidiary keys, each of which roughly corresponds to one of the old Control Panel modules.

The next first-level sub-key is called InstalledLocationsMRU. It is used by certain installation routines to create a history list for the control which prompts the user for the location of the source files.

This is followed by Keyboard Layout, which contains settings from the Language tab in Keyboard Properties. It includes a key named Preload, which in turn holds a key for each installed keyboard layout. These keys act as pointers to keys within HKEY_LOCAL_MACHINE\System\Current-ControlSet\Control\Keyboard Layouts, which in turn contain references to the keyboard drivers.

The next first-level sub-key is Networks. It in turn contains two keys: Persistent lists the mapped drives which are configured for reconnection at logon; Recent holds a key for each share on a connected computer which has been accessed from this computer. In each case, this shows the connection type and provider name.

Next, the RemoteAccess sub-key contains details of the user's Dial-Up Networking connections. The key itself contains settings common to all connections, such as the area code and the number of redial attempts. Below this, the Addresses and Profile keys contains settings for specific connections.

The last of the first-level sub-keys in HKEY_CURRENT_USER is easily the largest. It is named Software, and it is one of the two parts of the registry specifically intended for use by applications (the other is also named Software, and is in HKEY_LOCAL_MACHINE).

Immediately below HKEY_CURRENT_USER\Software, there is a key for each vendor which has applications installed on the computer. This in turn contains a key for each of the vendor's installed applications and, in some cases, a further sub-key for each installed version of the application. Beyond that, the content of each key is for the vendor to decide. Typically, they contain user preferences, histories and the like.

As an example, my own registry includes a key named HKEY_CURRENT_USER\Software\JASC\Paint-Shop Pro 5, which in turn contains 43 sub-keys. As well as my preferences for PaintShop Pro, these store the position and state of every toolbar and window, a recently-used file list, the recent locations for opening and saving each of the file types, and quite a lot more. This is an unusually large example - most applications don't store as much as this.

Although HKEY_CURRENT_USER\Software is mainly intended for third-party vendors, Microsoft also has a presence there. The key includes sub-keys for each installed Microsoft application (for example, HKEY_CURRENT_USER\Software\Microsoft\Office\8.0\PowerPoint) and also for Windows itself (HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion) . The latter holds user-specific settings for the Windows applets, Internet Explorer, Task Manager and other components.

In Windows NT, there are some additional first-level keys below HKEY_CURRENT_USER. They include Console (settings for the Command Prompt window), Environment (environment variables read at logon) and Unicode (references to applications that support Unicode).

HKEY_LOCAL_MACHINE

This is another large root key. It is the home of all the computer-specific information, including details of the hardware configuration and any machine-specific settings for the installed applications. Whereas each user who logs onto the PC sees different settings in HKEY_CURRENT_USER, they all see the same information in HKEY_LOCAL_MACHINE. It contains seven first-level sub-keys.

The first of the first-level sub-keys, named Config, contains all the hardware profiles which have been set up for the machine (do not confuse these with user profiles, which are in HKEY_USERS). Each hardware profile has its own key, one level down from HKEY_LOCAL_MACHINE\Config; these are named 0001, 0002, etc. Each profile contains configuration details for the monitor, printers and other devices present in the profile, as well as certain Internet-related settings.

The second of the first-level sub-keys is Enum. This holds information about all the devices and peripherals

installed in the computer, including such details as the device type, drive letter, hardware ID and manufacturer. It might also include devices that are not currently available. For example, if you have changed your monitor, both monitors might have an entry (in HKEY_LOCAL_MACHINE\Enum\Monitor), with a further entry, named Default_Monitor, used to point to the one currently installed.

Enum contains a key for each class of hardware. These vary according to the installed devices, but will typically include: BIOS (devices used with a plug-and-play BIOS), ESDI (installed ESDI drives), Flop (floppy disk drives), LptEnum (plug-and-play printers), MF (multi-function boards), Monitor (monitors), Network (network protocols and bindings), PCI (PCI devices), Root (certain legacy devices), SCSI (SCSI devices) and SerEnum (serial plug-and-play devices).

The next first-level sub-key, named Hardware, contains a few details about the CPU, floating-point processor and serial ports. This is followed by Network, which stores information about the current network logon (if any), including the user name and the name of the primary network provider. Next, the Security sub-key contains details of any security provider.

The largest of the first-level sub-keys comes next. It is named Software, and it closely parallels the Software key in HKEY_CURRENT_USER. However, while HKEY_CURRENT_USER\Software contains user-related settings for the installed applications, the HKEY_LOCAL_MACHINE version contains computer-specific settings. For example, HKEY_CURRENT_USER\Software\Microsoft\Office\8.0\PowerPoint includes the current user's preferences for PowerPoint; the corresponding branch in HKEY_LOCAL_MACHINE contains the application's directories, details of the installed filters and so on.

In addition, HKEY_LOCAL_MACHINE\Software includes a key named Classes, which holds information about registered file types and their associated applications. This key is aliased by HKEY_CLASSES_ROOT, which is described above.

The last of the first-level sub-keys in HKEY_LOCAL_MACHINE is named System. It contains a single key (in Windows 9x), named CurrentControlSet, which in turn contains two keys: Control and Services. The former stores certain information needed at boot time, including the computer name, file system settings, multimedia resources, descriptions of network providers and information about national language support. The Services key lists the device drivers which Windows must load during booting.

In Windows NT, HKEY_LOCAL_MACHINE does not have Config, Enum or Network sub-keys; some of their settings can be found under the System key instead. The Hardware key contains more extensive information about hardware devices and their current status (roughly corresponding to the details shown in the Windows NT Diagnostics applet). The Security key is also more extensive; it contains the settings which are configured from User Manager. And there is one additional first-level key in NT: the SAM key holds user and group account information.

HKEY_USERS

This root key contains a sub-key for each user profile. There is a further sub-key, named .Default, which provides default values for new user profiles. If user profiles are not enabled, .Default stores the settings for the actual user.

When a user logs on, Windows creates the HKEY_CURRENT_USER alias from the corresponding profile. The contents of the profile key within HKEY_USER are therefore identical to that of HKEY_CURRENT_USER (described above).

HKEY_CURRENT_CONFIG

As mentioned earlier, HKEY_LOCAL_MACHINE\Config contains details of the installed hardware profiles (this applies only to Windows 9x). Each profile has its own key within Config - named 0001, 0002 etc - which holds configuration details for the profile. There is always at least one profile key.

The HKEY_CURRENT_CONFIG root key is an alias for the current hardware profile. Its content is therefore identical to HKEY_LOCAL_MACHINE\Config\nnnn, where nnnn is the profile number.

In Windows NT, hardware profiles are stored in HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles, and HKEY_CURRENT_CONFIG is an alias for HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\Current.

HKEY_DYN_DATA

This final root key (which is not present in NT) is a memory-resident copy of certain other registry items. It contains information which Windows needs to retrieve particularly quickly.

The root key contains two sub-keys. The first, named Config Manager, holds details of the current hardware configuration as seen by the Plug-and-Play Configuration Manager. Windows builds this information (which is sometimes referred to as the hardware tree) by examining the hardware during booting; the information is then updated dynamically as plug-and-play devices are installed and removed.

The other sub-key is named PerfStats. This contains performance information about network components.

PCSA

The Author

Mike Lewis is a freelance technical journalist and a regular contributor to PCSA. You can contact him by email at mike.lewis@itp-journals.com.