

# WINDOWS NT REGISTRY

By Lance Jensen

## THE REGISTRY: BASICS

The Registry is a unified database containing most of the information about your hardware and the installed software and the settings for their use, set up in a tree hierarchy. You can view and edit the contents using the registry editors (regedit.exe or regedt32.exe), but many changes can also be done using the existing administrative tools such as Control Panel. It is better to use the administrative tools whenever possible, as they will store the changes correctly. You can easily make a mistake while using the registry editors, and they will not warn you if you do.

CAUTION!

EDITING THE REGISTRY CAN BE EXTREMELY DANGEROUS, AND CAN DISRUPT YOUR SYSTEM TO THE POINT WHERE YOUR ONLY OPTION IS TO RE-INSTALL WINDOWS NT.

Even if you know exactly what you are doing and are completely certain what the results will be, it is good policy to always back up your registry before making any changes. If you observe all precautions and don't "experiment" the Registry itself may become one of your favorite tuning tools.

The Windows NT Resource Kits contain the programs **regback.exe** and **regrest.exe**. Use **regback.exe** to back up the registry and **regrest.exe** to restore the registry.

The registry editor **regedit.exe** has a more sophisticated search capability than the other one, **regedt32.exe**, but cannot be used to enter all of the value types that the Registry supports. It is often more efficient to use **regedit.exe** to find values in the Registry, then switch to **regedt32.exe** to make changes. There is also a read-only switch in **regedt32.exe** which I strongly recommend you use. In **regedt32.exe**, click Options on the Menu Bar, then click Read Only Mode if it is not checked. If you try to make a change, you will be notified that Registry Editor is operating in Read Only mode, and no changes you make will be saved.

## **A Brief Description**

Each sub-tree, or set of keys, sub-keys and values, is called a "hive". Within each hive there are keys, which may have sub-keys, and sub-sub-keys, and so on. At the lowest level there is a value entry comprised of a name, a data type, and a value. For example, one value entry has the name "SecondLevelDataCache", the data type "REG\_DWORD", and the value "0" (by default).

Each hive is rooted at the top of the Registry hierarchy, and most are backed by a main file, a save file and a log file in the folder **%systemroot%\system32\config**. The main file has no extension, the others have the extensions .sav and .log. Exceptions are **KEY\_LOCAL\_MACHINE\HARDWARE**, which has no files, and **HKEY\_CURRENT\_USER**, which stores its files in **%systemroot%\Profiles\**, where **<username>** is the name of the current user.

The hives and their files are:

**HKEY\_LOCAL\_MACHINE** - This has information about the local machine. It contains five hives:

- **HKEY\_LOCAL\_MACHINE\HARDWARE** - Contains information about your hardware, including cards in expansion slots, connections through ports, and the related interrupts. This data is determined and stored on boot-up, so it is not saved in any files. You should never need to edit

# WINDOWS NT REGISTRY

By Lance Jensen

any data here, and probably couldn't understand much of it because it is in binary format. If you do happen to change something, don't worry about it; just reboot and the correct data will be determined.

- **HKEY\_LOCAL\_MACHINE\SAM** - Security Accounts Manager, containing user account names and passwords and security settings. You should never need to change anything here, as it is maintained on Workstations via User Manager, or on Servers by User Manager For Domains. Files: Sam, Sam.sav and Sam.log.
- **HKEY\_LOCAL\_MACHINE\SECURITY** - Contains the security information for the local machine. This is also maintained via User Manager. Files: Security, Security.sav and Security.log.
- **HKEY\_LOCAL\_MACHINE\SOFTWARE** - When you install an application or package, its configuration is stored here under the manufacturer's name. For example, when you install the Executive Software Network Undelete utility, a sub-key \Executive Software is created, with a \Network Undelete sub-key within it. If you then install the Diskeeper defragmenter, a \Diskeeper sub-key will be created within \Executive Software. There is also a sub-key called \Classes which lists all file extensions. Files: software, software.sav and software.log.
- **HKEY\_LOCAL\_MACHINE\SYSTEM** - This is probably the most useful as well as the most dangerous hive, because it contains the startup data that cannot be calculated during startup. This data is stored in ControlSet sub-trees. One of these, CurrentControlSet, is actually a link to one of the others (ControlSet001, ControlSet002, etc.) which contains the data set currently in use. This data is normally modified via utilities in Control Panel. Files: system, system.sav and system.log. There is also system.alt, which is a backup of the system hive, and makes it possible to undo changes that had unexpected side-effects.

Subsets are:

- **HKEY\_CLASSES\_ROOT** - Points to a child (or sub-set) of HKEY\_LOCAL\_MACHINE, at \SOFTWARE\Classes. It contains the Object Linking and Embedding (OLE) and file-class association data.
- **HKEY\_CURRENT\_CONFIG** - Points to subset of CurrentControlSet (as described above), containing the current configuration. It is thus stored in the files called system, system.sav and system.log (the same files as for **HKEY\_LOCAL\_MACHINE\System**).
- **HKEY\_USERS** - This contains the user profiles of all users currently loaded on the system. File names: default, default.sav and default.log.
- **HKEY\_CURRENT\_USER** - Points to a child of HKEY\_USERS, being the user who is currently logged on. File names: ntuser.dat and ntuser.dat.log.

## CONTROL SETS

The Control Sets contain the parameters for the system's services and devices. They are located in **HKEY\_LOCAL\_MACHINE\SYSTEM**. You will likely have two of the numbered sets (ControlSet001, ControlSet002, etc.), though you may have as many as four, and you will always have **CurrentControlSet**, which is a link to the numbered set which contains the data set currently in use.

# WINDOWS NT REGISTRY

By Lance Jensen

When the system starts, the numbered set used (usually ControlSet001) is copied into **HKEY\_LOCAL\_MACHINE\SYSTEM\Clone**, and **CurrentControlSet** is linked to that numbered set. The copy in \Clone also replaces the **LastKnownGood** configuration, once the startup is declared "good" (generally meaning there were no Severe or Critical errors, and a successful logon was done).

This is a very important piece of information, because it shows what to do if you accidentally botch registry changes, and wish to revert your Registry to the way it was prior to the changes. Here's what you do:

1. Reboot.
2. Select the operating system to boot to.
3. Invoke the Configuration Recovery menu.
4. Select Last Known Good.

Any changes you made since the last "good" startup will vanish. Note that this will only work if you have not fully rebooted since the changes. If you have, then your changes to the Registry will have already been saved. A way to be sure every time is to back up your Registry prior to making any changes, so that you always have your last good copy of the Registry to fall back on.

**HKEY\_LOCAL\_MACHINE\SYSTEM>Select** contains the value entries Current, Default, Failed and **LastKnownGood**. Their values are the corresponding numbered sets. For example, you will probably see Current and Default as "REG\_DWORD: 0x1". This means ControlSet001 is the default set and is the set currently in use. "0x2" refers to ControlSet002, and so on. If you have never had a failed boot, Failed will be "REG\_DWORD: 0". While you can manually set LastKnownGood to any existing Control Set, this is not recommended because if you make a mistake in this setting, you won't be able to select an alternate boot. If your default boot then fails, you'll have to do an emergency repair and may have to re-install Windows NT! It's best to let Windows NT handle this default.

## The Control Sub-Key

Each Control Set has the sub-keys **\Control** (for controlling the system) and **\Services** (for controlling the services). **\Control** contains parameters necessary for the system to start. While most of these parameters are controlled through utilities in Control Panel, this is where most "tweaking" would be done.

There are several sections here that you should leave alone, specifically, **\Lsa**, **\ProductOptions**, **\VirtualDeviceDrivers** and **\WOW**. Changes to these can prevent the system from starting or running or can make it impossible for anyone to log in, so let the system maintain these. There are two others, **\HiveList** (paths to the Registry files) and **\Windows** (paths to the Windows NT and System folders). Don't fool with these, either, but theoretically you could make some changes here that would allow you to move files off the system partition in an emergency (for example, if you run out of free space). I'm doing some experimenting on a test system, and will report anything useful, but any changes here will certainly interfere with an operating system upgrade or Service Pack installation. Play it safe, and don't change these unless you absolutely must.

# WINDOWS NT REGISTRY

By Lance Jensen

## The Other Control Set Sub-Keys

**\Services** contains data on drivers and on their associated hardware. You maintain this data from Control Panel, using the Devices, Network, Services and UPS icons. I have never come across a need to make changes manually, except deleting keys while manually uninstalling an application when Add/Remove Programs fails.

- **\Hardware** contains five entries, 001 through 004 and Current, which correspond to ControlSets. These contain data defining hardware that is run by drivers listed in \Services. These are also maintained entirely from Control Panel.
- **\Enum**. This shows up on my machine, but I have not been able to find any references to it.

**HKEY\_CURRENT\_CONFIG** - Points to a subset of **CurrentControlSet** (as described above), containing the current configuration.

By now you should have a good understanding of what we are dealing with. The next few articles will start on details: What specific value entries there are, what they mean, and how you can use them to tune your system. Do you have your Registry backup tools ready?

## SESSION MANAGER

At the **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager** sub-key, you will find the global variables that are used to control sessions. When you click **\Session Manager**, you will see on the right half of the window several value entries that contain global variables. Most of these variables must be left alone, but you can change GlobalFlag. If you have applications that can run under both OS2 and MS\_DOS, they will run under OS2 if GlobalFlag is set to the default 0x21100000 or under MS-DOS if you change the value to 0x20100000.

Note that you may have another sub-key called **\SessionManager** (no space between the words). Leave this one alone and just work in **\Session Manager**.

## Session Manager Sub-Keys

There are also sub-keys under **\Session Manager**:

- **\AppPatches**: This contains sub-keys containing value entries which document patches that have been applied to various applications.
- **\DOS Devices**: These are links that Windows NT creates at startup. You shouldn't change these.
- **\Environment**: Paths to various subsystems such as OS2. You can change these if you want to move the subsystems off the system partition, but be very careful: If you misspell a path, the subsystem it refers to won't run. This can be disastrous, as the value entry "Path" refers to Windows NT logon, and "windir" points to the Windows NT folder. If you get this wrong, you may have to do a repair or re-install Windows NT. We do not recommend that you move parts of the Windows NT operating system off of the system partition.
- **\Executive**: These value entries are for advanced system tuning such as creating additional process threads. (A thread is an agent of a process, which runs program code. A process can

# WINDOWS NT REGISTRY

By Lance Jensen

- **\FileRenameOperations:** System files that are locked cannot be changed while Windows NT is running. However, there are ways to copy, move or rename them. When this is done, the change is not completed till the system is rebooted. The value entries at this location are used to complete the change when you reboot. There is nothing here that you will ever need to change manually.
- **\KnownDLLs:** Dynamic Link Libraries (DLLs) are essentially subroutines that applications use during execution. The DLLs listed here are loaded into memory during startup, and stay there. Theoretically you could save memory by removing from this subkey any DLLs that are never used. The hard part is determining which, if any, are safe to take out, so don't bother; it's not worth the danger. Buy more memory if you need it, but leave these entries alone.
- **\SubSystems:** These are paths for starting various subsystems. It's a lot like \Environment, with the major difference that these values are set during startup. If you do change a value in **\Environment**, the corresponding value in **\Subsystem** (if there is one) will be automatically set when you reboot. You should never need to change anything here.
- **Memory Management:** This is the most likely area to need tuning. Most of the value entries are maintained from **Control Panel\System\Virtual Memory**, but there are a couple you may tweak manually.

## Memory Management Value Entries

Some of these entries may not appear in your Registry. If so, you can add them, but be sure you spell them EXACTLY as presented here, without changing the case of any character.

- **ClearPageFileAtShutdown:** When this is set to a Value Type of REG\_DWORD and a value of 1, all data in the paging file will be cleared upon system shutdown. You must reboot for this change to take effect. The paging file will not be cleared on this reboot, but will clear on each subsequent one. This is a modification to make if you're concerned about security - when this modification is made, if an unauthorized user should access the disk, they won't be able to read data in the paging file.
- **DisablePagingExecutive:** When set to zero (default), this allows Windows NT to page the kernel pools to the paging file; set it to one, and the kernel pool will stay in memory. If you have a large amount of unused memory, or if for some reason your paging file is slow, this might be of value. Disabling the Paging Executive may slow your system to a crawl, so if you are going to try changing this, pick a time when your system can be out of production for a while.
- **IoPageLockLimit:** This value is the maximum bytes of memory that can be locked for I/O operations. A value of 0 defaults to 512KB. Raising this value can give you a significant performance boost. The procedure is detailed in the article "Memory Usage" (eLetter Volume 3, Issue 6).
- **LargeSystemCache:** 0 tells the system to favor the processes working set, non-zero means to favor the system-cache working set. For most systems, your applications will run faster if this

# WINDOWS NT REGISTRY

By Lance Jensen

value is set to zero; if it is non-zero, your paging file may be over-active (if you have a noisy hard drive, check to see if LargeSystemCache is non-zero). Thus most Workstations should have this set to zero, and most Servers should have it set to one.

- **PagedPoolSize:** Also Min, Max, and others, and all of these for NonPagedPool. "Pool" is all of the system memory. "Paged" means it can be paged, or written, to the disk. "NonPaged" means it can't be written to the disk. The values in the Registry are normally zero, which tells Windows NT to calculate default values relative to the amount of RAM on your computer. You should leave these alone because changing these values can cause Windows NT to miscalculate other resource allocations, and incorrect values can cause Windows NT to malfunction and possibly even cause file system corruption.

A professional who knows what side-effects will occur may benefit from reducing the poollocations (setting values larger than the defaults will have no effect), but I'm sure that very few people outside Microsoft know enough to safely tinker with this. If you're interested in the details of the default calculations, see the Microsoft article Q126402 in the Microsoft Knowledge Base. You can find Knowledge Base articles by going to [www.microsoft.com](http://www.microsoft.com), clicking Support in the menu bar, then clicking support online in the first paragraph. Click the radio button "Specific article ID number" then enter the article number.

- **PagingFiles:** Data about existing paging files (location and sizes) is stored here. You should use **Control Panel\SystemPerformance** to adjust your paging files, but this value can be handy if you get in trouble. For example, if your paging file is smaller than your physical memory or your system partition does not have enough free space to record a crash dump file, then if you get a bug check (the blue screen crash), your system may go into a continuous series of reboots. The Microsoft Knowledge Base article Q174630 details how to handle this.
- **SecondLevelDataCache:** This is the amount of L2 cache Windows NT will use. It defaults to 0, which is the correct value for 256KB of L2 cache. If it is set to 0, but you have more than 256KB cache, you should change it:
  1. Double-click SecondLevelDataCache to bring up the DWORD Editor.
  2. Click the Decimal radio button.
  3. Enter the amount of L2 cache you have, e.g., "512" for 512KB of cache.
  4. Click OK.

This will probably give you a noticeable performance increase. Some people have reported no change, some say their performance more than doubled.

- **SystemPages:** Here you specify the number of page table entries available. The default is almost always sufficient, but if you install a PCI card with a very large amount of on-board memory (like a very sophisticated video card), and you cannot access all of the card's memory, this is probably where the solution will be. Contact the card's manufacturer for the correct value to enter.

## THE REGISTRY: CONTROL, PART 1

At **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\** we find the values for controlling the system. Some of these were covered in the previous two Registry articles, "Control

# WINDOWS NT REGISTRY

By Lance Jensen

Sets" and "Session Manager". In this article and the next two we will look at the rest of these variables.

Many of the **\Control** value entries are necessary for the system to start, so you must be careful if you make any changes, and always make sure you have backed up the Registry first. Most of these parameters can be controlled through Control Panel; where that is possible, it is the best way to make changes. Changes in this area almost always require a reboot to take effect. Note that some of these variables may not appear in your Registry; they can be added if needed, but be sure you spell and capitalize them correctly.

## The Control Value Entries

- **CurrentUser, REG\_SZ.** This is for holding the username of you, the user who is currently logged on.
- **RegistrySizeLimit, REG\_DWORD, Default 8MB, 25% of PagedPoolSize** (PagedPoolSize is located at **\CurrentControlSet\Control\SessionManager\MemoryManagement**). This is the amount of memory that can be used for Registry data. It can range from 4 MB up to 80 percent of **PagedPoolSize**. The value is entered as the number of bytes, not the number of MB. If you increase **PagedPoolSize**, this value will also increase. A value of 0xFFFFFFFF sets RegistrySizeLimit to 80% of PagedPoolSize.
- **SystemStartOptions, REG\_SZ.** If the firmware passes system arguments to the system, they are listed here. You will not need to change anything here.
- **WaitToKillServiceTimeout, REG\_SZ, Default 20,000 milliseconds (ms).** Sets how long the service control manager will wait for each service to complete the shut-down request.

## The Control Sub-Keys

There are more than thirty sub-keys under **\Control**. Some will not appear in your Registry, but can be added if you need them. Some have enough sub-keys of their own to fill a whole article. I am not covering them alphabetically, so don't be concerned if I seem to skip a few; there are several more articles coming.

**\BootVerificationProgram:ImagePath, REG\_SZ or REG\_EXPAND\_SZ,** Defaults to blank. This value entry contains the path and filename of the program which the service controller uses to verify the Last Known Good configuration. If you change this from the default, you must also go to **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\WinLogon** and set the value entry ReportBootOK (REG\_SZ) to 0. These sub-keys should be left at the default unless you are quite certain you know what you are doing. For one thing, you'll probably have to write the program it calls.

## \Class

You'll find a dozen or more sub-keys under **\Class**, each with a cryptic name. Don't worry about them, because, like most **\Control** entries, you should not modify them. These sub-keys define devices such as keyboard, mouse, modem, etc., and are modified from Control Panel.

**\ComputerName:** This has two sub-keys, **\ActiveComputerName** and **\ComputerName** (yes, the name is identical). The value entry ComputerName, REG\_SZ, will be in the first sub-key, and may be

# WINDOWS NT REGISTRY

## By Lance Jensen

in the second. This is the network name of the computer. You can change it in Network\Identification in Control Panel.

**\FileSystem:** Contains four value entries:

- **NtfsDisable8dot3NameCreation, REG\_DWORD, Default 0.** Allows long file names on NTFS partitions. If you set this value to 1, long file names can not be used on your NTFS partitions. If Windows NT is taking a very long time to process directories, it may be due to having a large number of long file names. If so, setting this value to 1 may speed up the directory processing. On the other hand, you will not be able to use long file names, and you will not be able to use MS-DOS shortcuts that have long file names.
- **NtfsDisableLastAccessUpdate, REG\_DWORD, Default 0.** Whenever Windows NT accesses a file or folder, even if it's just to display the name in a list of folder contents, the Last Accessed Date is updated. If you normally deal with large numbers of files and folders, this could slow you down. To disable this feature, set this value to 1.
- **Win31FileSystem, REG\_DWORD, Default 0.** Controls whether the FAT will allow creation, enumeration, opening, or querying of long file names, and whether extended time stamp information (CreationTime and LastAccessTime) is stored and reported. This value defaults to 0; set it to 1 (true) to revert to basic Win3x (and Windows NT 3.5) semantics. Note: Changing this value does not change any disk structures. It simply changes how the system behaves from now on. You must reboot the system for a change to this value to take effect.
- **Win95TruncatedExtensions, REG\_DWORD, Default 0.** When set to 0, this makes all file extensions look like 3-character extensions. Thus Windows NT would consider all of these extensions to be the same: .LIS, .LIST, .LISTS, .LISTED, .LISTING, and so on. Any action done on \*.LIS would be performed on all of these files. To disable this feature, set this value to 1.

**\Keyboard Layout:**

**KeyboardLayout, REG\_SZ.** This key contains the name of the .DLL file which the system loads to map your keyboard. You will probably never need to change this. It contains two sub-keys.

**\DosKeybCodes:** This contains a set of value entries, each of which is an MS-DOS style layout name. The system uses it to convert Windows NT layout names. Each value entry is the code. For example, US is 00000409. Note that these are text strings, so the value type is REG\_SZ.

**\Substitutes:** If a particular user prefers a keyboard layout which is different from the default, the code for the layout is recorded here. When that user logs in, the system loads the corresponding .DLL file. As under **\DosKeybCodes**, each value entry is the code. The type is REG\_SZ, Default is blank.

**\Keyboard Layouts:** Under this key we have a sub-key for each layout name, (as listed in \Keyboard Layout \DosKeybCodes). Each sub-key contains two value entries:

- **LayoutFile, REG\_SZ.** Contains the name of the .DLL file.
- **LayoutText, REG\_SZ.** Contains the name of the keyboard layout.

# WINDOWS NT REGISTRY

By Lance Jensen

## THE REGISTRY PART 5: Control\Print

Note that some of these variables may not appear in your Registry; they can be added if needed, but be sure you spell and capitalize them correctly.

### The Print Key

Here we have the data pertinent to your printers. There will be sub-keys for DLLs and drivers that are necessary for the printers and print spoolers, and possibly sub-keys installed by OEMs.

- **The \Environments Sub-key**

Here you will find several environment, or operating system, descriptions. For example, on my machine, I have:

**\Windows 4.0**  
**\Windows NT Alpha\_AXP**  
**\Windows NT PowerPC**  
**\Windows NT R4000**  
**\Windows NT x86**

Each of these sub-keys contains a value entry Directory whose value is the driver directory. In \Windows NT X86 this value is W32X86. There are also two sub-keys:

**\Drivers:** For each printer that you have configured on this system, there will be one or more sub-keys. They will contain value entries for data that applies to the printer, such as the names of the configuration files and driver DLLs. The files will reside under the driver directory mentioned in the prior paragraph.

**\Print Processors:** This contains a value entry Driver whose value is the name of the print DLL.

- **The \Monitors Sub-Key**

**\Local Port: Driver, REG\_SZ.** Contains the name of the local monitor DLL.

**\PJM Language Monitor:** PJM stands for Printer Job Language. This sub-key contains the value entries Driver, whose value is the PJM DLL file name, and EOJTimeout, REG\_DWORD, whose value is the number of milliseconds to End-of-Job timeout.

**\Provider Network Port** contains the value entry Driver, whose value is the name of the DLL for the print monitor. It also has a sub-key **\Options** which contains several value entries defining connection, buffers, timers, etc.

- **The \Printers Sub-Key**

**\Printers** has no sub-keys, but several useful value entries.

**DefaultSpoolDirectory, REG\_SZ.** This is the path to the default print spooler directory, used by all of the printers. If you want a particular printer to use a different spooler directory, add the following value entry:

# WINDOWS NT REGISTRY

By Lance Jensen

**SpoolDirectory, REG\_SZ.** Enter the path to your alternate print spooler directory. Note that if you misspell the path, or the directory does not exist, the default print spooler will still be used.

**JobPrintsWhilstSpooling, REG\_DWORD, 0=disabled, 1=enabled.** See below.

**FastPrintWaitTimeout, REG\_DWORD, Default 24,000ms.** This is the time the port thread will wait for data. If it times out, then the print job will be paused, and the next print job will start. NOTE: If **JobPrintsWhilstSpooling** is enabled, the port thread must synchronize with the spooling application.

**FastPrintSlowDownThreshold, REG\_DWORD. Default: FastPrintWaitTimeout divided by FastPrintThrottleTimeout.** If JobPrintsWhilstSpooling is enabled, your printer may pause if no data is received for a specified period. **FastPrintSlowDownThreshold** is used to prevent this pause.

**FastPrintThrottleTimeout, REG\_DWORD, Default: 2,000ms.** When the **FastPrintSlowDownThreshold** is reached, the print spooler cuts the speed at which it sends data, so that there will not be a long enough period between data packets to allow the printer to pause.

**NetPrinterDecayPeriod, REG\_DWORD, Default: 3,600,000ms (1 hour).** There is a list of printers available to the browser. This value specifies how long a network printer will be kept on that list.

The last three value entries set the priorities for printing. They are all REG\_DWORD and default to 0 (Normal), but can be set to 1 (High) or 0xFFFFFFFF (Low).

**PortThreadPriority:** Sets the priority of the threads that carry data to the printer.

**SchedulerThreadPriority:** Sets the order that threads get access to the printer (High threads go first, then Normal, then Low).

**SpoolerPriority:** Sets the priority of the spooler as an application.

- **The \Providers Sub-Key**

There are two value entries:

**EventLog, REG\_DWORD , default 1.** When a print job completes, an entry is made in the event log. Set this to 0 to disable the logging, then go into Control Panel \Services and stop and start the spooler.

**NetPopup, REG\_DWORD, default 1.** When a print job completes a notification pops up. Set this to 0 to disable the notification.

The **\ Providers sub-key** also has a tree of sub-keys, similar to this:

## **\LanMan Print Services**

# WINDOWS NT REGISTRY

By Lance Jensen

**\Monitors**  
**\LanMan Print Services Port**  
**\Servers**  
**\SERVERNAME**  
**\Forms**  
**\Printers**  
**\HP LaserJet**  
**\PrinterDriverData**

The default top-level sub-key for a Windows NT network is **\LanMan Print Services**. It has two value entries:

Name, REG\_SZ, whose value is the name of the DLL file for the service.  
DisplayName, REG\_SZ, whose value is the name which is displayed to identify the service.

**\Monitors** has no value entries, but its sub-key **\LanMan Print Services Port** has a value entry Driver, REG\_SZ, whose value is the name of the printer driver DLL.

The next sub-key, **\Servers**, has one sub-key for each server in the network; the sub-key name (where I wrote **\SERVERNAME** in the tree diagram above) is the server name. Under the server is **\Forms**, which has a REG\_BINARY value entry for each defined print form. At the same level is the sub-key **\Printers**.

Under **\Printers** there is a sub-key for each installed printer, each with a sub-key **\PrinterDriverData**. These contain many value entries defining the printer and its driver. I'll not describe these value entries because it would take another two pages, and they are all set automatically through Control Panel **\Printers**.

Now for the rest of the minor Registry entries at **HKEY\_LOCAL\_MACHINE \SYSTEM \CurrentControlSet\Control**. Most of these values are controlled through Control Panel\Printers, but there are a few which you may want to modify manually. Changes in this area almost always require a reboot to take effect.

**\PriorityControl** has one value entry, Win32PrioritySeparation, REG\_DWORD, default 2, which controls the relative priority between foreground and background applications. This should be controlled through Control Panel \System \Performance. On Windows NT Workstation, a value of 0 means foreground and background threads get the same amount of processor time; 1 and 2 give more time to foreground threads, but I have not been able to find out just how much more. (Anybody out there know?) On a Windows NT Server, the processor time that threads get is fixed. The Win32PrioritySeparation value instead determines the priority boost given to foreground processes, with 2 being the highest boost.

**\SecurePipeServers** has one sub-key, \winreg, whose Class is REG\_SZ. It is used primarily to define who may have access to the Registry itself. In Windows NT 4.0, by default, only members of the Administrators group can access the Registry. You can alter the default in several ways:

- To change the default, go to \winreg and add the value entry Description, REG\_SZ, and set the value to Registry Server. Highlight **\winreg**, then select Security on the menu bar, then

# WINDOWS NT REGISTRY

By Lance Jensen

Permissions. Enter the users and groups you want to add, with the type of access you want them to have.

- To allow access to certain Users or Groups, add a sub-key \AllowedPaths under **\winreg**, leaving Class blank. Then add the value Machine, REG\_MULTI\_SZ. Enter the following string values:

**System\CurrentControlSet\Control\ProductOptions**

**System\CurrentControlSet\Control\Print\Printers**

**System\CurrentControlSet\Services\Eventlog**

**Software\Microsoft\Windows NT\CurrentVersion**

**System\CurrentControlSet\Services\Replicator**

- If you want to allow access only to certain parts of the Registry, add the value name Users, REG\_MULTI\_SZ, and enter the locations.

You also use this key for allowing users to monitor server performance. First, in **HKEY\_USERS**, select the SID of the local server user. Then select **\Control Panel\International\Locale** and note the basic language ID (the value for English is 409). Subtract 400 to get the number to use below.

If your system partition is NTFS format, make sure you have read access to these server files:

**%windir%\system32\PERFCnnn.DAT**

**%windir%\system32\PERFHnnn.DAT**

by following these steps:

1. Run Windows NT Explorer
2. Right click on %windir%\system32\PERFCnnn.DAT
3. Click Properties
4. Click Security
5. Click Permissions
6. Make sure your Group has Read access or Full Control

Now highlight **\winreg** and select Security on the menu bar, then Permissions. Enter the user ID and set type of access to READ (or a higher permission). Then do the same for **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib**, but this time check the "Replace permissions on all sub-keys" box.

**\SecurityProviders** contains data regarding system security. It has one sub-key **\SCHANNEL**, which includes the sub-keys **\CertificationAuthorities**, **\Ciphers**, **\Hashes**, **\KeyExchangeAlgorithms** and **\Protocols**. Any of these that are in use on your system will contain further sub-keys. For example, under **\CertificationAuthorities** you will find a sub-key for each authority you use, such as **\AT&T Certificate Services**. Each of these sub-keys will have three value entries:

**CACert, REG\_BINARY, containing a certification code.**

# WINDOWS NT REGISTRY

By Lance Jensen

**Enabled**, REG\_DWORD, value 0x1 if the authority is enabled.

**Type**, REG\_DWORD, (I have never found a definition for this value entry).

**\ServiceProvider** contains two sub-keys. **\Order** defines the sequence in which existing providers will be used, and lists any providers to be excluded. **\ServiceTypes** contains sub-keys defining the types of service providers available, such as \Microsoft Internet Information Server. Value entries under these last sub-keys contain data defining the provider, such as the TCP port.

**\Setup** contains information used by Windows NT Setup. It has three value entries which, with their x86-based computer defaults, are:

**keyboard**, REG\_SZ, default STANDARD

**pointer**, REG\_SZ, default msrser

**video**, REG\_SZ, default VGA

**\TimeZoneInformation** has eight value entries, maintained through Control Master \Date/Time. You'll never need to change them.

**\Update**: if your Windows NT system was installed over an earlier version of Windows, you will see here the value entry UpdateMode, REG\_DWORD, with a value 0x1.

**\WebPost**, through its sub-key \Providers, lists codes for available Internet Service Providers (ISPs).

## THE REGISTRY 9: \HARDWARE

Here, we cover **HKEY\_LOCAL\_MACHINE\HARDWARE**. This hive contains information about your hardware, including cards in expansion slots, connections through ports, and the related interrupts. This data is determined and stored on boot-up, so it is not saved in any files. You should never need to edit any data here, and probably couldn't understand much of it because it is in binary format. If you do happen to change something, don't worry about it; just reboot and the correct data will be determined.

**\HARDWARE** has no value entries; it has four keys, which are detailed in the four sections below.

### \DESCRIPTION

System devices are listed in the Registry by names or codes. This is where those names and codes are defined. The source of this data depends on your computer. On a Digital Alpha RISC system, the data is copied from the ARC configuration database in the firmware. On an x86 system, the Hardware Recognizer NTDETECT.COM gathers the data during startup. On a non-x86 system the data is gathered by a version of NTDETECT.COM provided by the OEM.

**\System** is the sole sub-key of \DESCRIPTION. It contains value entries defining the System and Video BIOS and the motherboard itself. It's a convenient place to check your BIOS version and revision date. Under **\System** there are three sub-keys:

# WINDOWS NT REGISTRY

By Lance Jensen

- **\CentralProcessor** lists the CPUs, each under its own sub-key \0, \1, etc. Each sub-key has five value entries describing the CPU, including the vendor and clock speed. The first three value entries are also found under each of the "number keys" (\0, \1, etc.) under \System.

Component Information, REG\_BINARY. Contains version information.

Configuration Data, REG\_FULL\_RESOURCE\_DESCRIPTOR. Contains data such as the I/O port addresses and the IRQ number. (If this data is not available, this value entry will not appear.)

Identifier, REG\_SZ. Contains the name of the device.

VendorIdentifier, REG\_SZ. Identifies the CPU manufacturer.

~MHz, REG\_DWORD. Contains the approximate rated speed of the CPU.

- **\FloatingPointProcessor** lists the math co-processors in sub-keys, which have the same value entries as **\CentralProcessor**, describing the co-processor.
- The third sub-key, **\MultifunctionAdapter**, has three sub-keys which hold the data about the adapters in your system that are BIOS-controlled.

**\0** holds the configuration data for the PCI bus, with subkeys for any BIOS-supported devices that are plugged into this bus.

**\1** will hold the configuration data for the Plug and Play BIOS, but, since Plug and Play is not fully implemented in Windows NT 4.0, there are no sub-keys.

**\2** holds the configuration data for the ISA bus, with subkeys for any BIOS-supported devices that are plugged into this bus.

Under these "number keys" there are several more sub-keys for controllers. Which key you will find them under depends on which bus they are connected to. Each sub-key will have one or more sub-keys, depending on how many controllers you have. For example, you probably only have one keyboard controller, and thus only the **\0** subkey under **\KeyboardController**, but if you have two disk controllers, you will have **\0** and **\1** under **\DiskController**. (Note: The numbers here do not refer to the type of bus.)

**\DiskController** contains the data for your hard-disk and floppy-disk controllers. Under each "number key" it will have the sub-keys **\DiskPeripheral** and/or **\FloppyDiskPeripheral**, which will have "number keys" for each attached disk drive.

**\KeyboardController** contains the data for your keyboard controller. Under the "number key" will be a sub-key **\KeyboardPeripheral**, which contains a "number key" describing the keyboard itself.

**\ParallelController** contains the data for your parallel port controller. It has a "number key" for each installed parallel port.

# WINDOWS NT REGISTRY

## By Lance Jensen

**\PointerController** contains the data for your mouse port controller. It has a "number key" for each installed mouse port.

**\SerialController** contains the data for your serial port controller. It has a "number key" for each installed serial port.

Under each of these last three keys, if there is a device plugged in to a port, there will be a `\xxxPeripheral` subkey, such as `\PointerPeripheral` for a mouse or touchpad, which contains a "number key" describing the device.

### **\DEVICEMAP**

Here we find several subkeys, each containing at least one value entry. The value entries contain either a string defining where in the Registry the driver data is stored, or a string containing a port name. The Registry location is **HKEY\_LOCAL\_MACHINE \SYSTEM \ControlSetnnn\Services**; usually the **\ControlSetnnn** is the same control set that is mirrored in **\CurrentControlSet**. The sub-keys under **\Services** contain data on the drivers and on their associated hardware. You maintain this data from Control Panel, using the Devices, Network, Services and UPS icons.

- One sub-key, **\Scsi**, needs a bit more explanation. Here you will find a sub-key for each SCSI host device, in the order that the system discovers them. Under each SCSI host device will be a sub-key for each bus on that device. Under each bus will be subkeys for each SCSI device attached. If you are trouble-shooting an unfamiliar system, this can be useful in locating all SCSI devices on the system and exactly where they are.

### **\OWNERMAP**

If any devices are owned (controlled by another device), the device and its owner are recorded in value entries here.

### **\RESOURCEMAP**

Here you will find the connection settings and addresses for your system devices.

- **\Hardware Abstraction Layer** names in its sub-key the type of HAL in use on your system. There are many possible HALs, such as Compaq and PowerPC. On my system, this subkey is `\UP MPS 1.4-APIC platform`.
- The next sub-key, **\KeyboardPort/PointerPort**, has a sub-key defining the keyboard controller chip. If you use a standard keyboard, the sub-key will be `\i8042prt`.

### **\LOADED PARALLEL DRIVER RESOURCES and \LOADED SERIAL DRIVER RESOURCES**

Contain data on the parallel and serial port drivers, in value entries within the subkeys **\Parport** and **\Serial**.

- **\OtherDrivers** holds the data on drivers that are not standard system operations drivers. For example, I have a subkey `\sndblst` for my audio card.
- **\PointerPort** hold sub-keys containing data for pointers such as a mouse or touchpad.

# WINDOWS NT REGISTRY

By Lance Jensen

- **\ScsiAdapter** holds sub-keys for any SCSI adapters installed, with their settings.
- **\System Resources** contains memory settings, including Virtual and Reserved memory, in its subkeys **\PhysicalMemory** and **\Reserved**.
- **\VIDEO** contains your video driver information. The subkey depends on your video driver. For example, my system has **\stlth3d**. But there are two other sub-keys. **\VgaSave** describes the VGA driver which is used when the installed video card fails, or when you boot to VGA mode. **\VgaStart** notes which of the video drivers is currently in use.

Well, that's it for **\HARDWARE**. You can't really do anything with it, but it's a great source of information. The next two, **\SECURITY** and **\SOFTWARE**, though, you can change. Carefully.

## **\HKEY\_LOCAL\_MACHINE\SOFTWARE**

This hive contains data for all of the 32-bit software installed on your system. Each software package may appear as a sub-key of **\SOFTWARE**, but there will also be sub-keys which are manufacturers (such as Microsoft or Executive Software) with software packages listed as sub-keys below the company sub-key. The data under the software sub-keys includes configuration settings, file associations and OLE information. This data can include build number, registration information, paths to executable and data files, and anything else the manufacturer wants.

There are also several non-program sub-keys: **\Classes**, **\Clients**, **\Description**, **\Program Groups**, and **\Secure**.

### **\Classes**

In this sub-key, OLE (Object Linking and Embedding) and DDE (Dynamic Data Exchange) classes are defined. It contains a sub-key for each class, such as **\.exe** (executable) and **\.gif** (graphic image). Each sub-key has a value entry whose value is the program used to open this type of file; this program is what you are asked to specify when you see the Open With dialogue box.

### **\Clients**

This section defines clients such as your internet e-mail package, and other applications such as Microsoft Outlook. Sub-keys and data can vary, depending on the application, but as an example, Microsoft Outlook on my machine looks like this:

```
\Mail
  \Microsoft Outlook
    \Protocols
      \mailto
        \DefaultIcon
        \shell
          \open
            \command
      \shell
        \open
          \command
```

# WINDOWS NT REGISTRY

By Lance Jensen

The only value entries are the package name, the mail-to protocol, the path to the icon, and the shell open command strings.

## **\Description**

This is where Windows NT stores the names and versions of your software. It can be useful for lookup, but should never be changed manually.

## **\Program Groups**

Descriptions of any program groups, as maintained with Program Manager, are stored here.

## **\Secure**

I'm afraid I have found very little about this key, just that it's a storage location for keys that require more than the usual amount of security. If anyone can point me to a data source, I'd appreciate it.

## **\Software Packages**

The data stored for each software package will vary widely. As an example, here is what our registry entries might look like:

### **\Executive Software**

#### **\Diskeeper**

**\2.0**

#### **\Analyze**

**\Fat**

**\Ntfs**

#### **\CurrentVersion**

#### **\Defragment**

**\Fat**

**\Ntfs**

#### **\ServerEntries**

#### **\UserSettings**

#### **\EmergencyUndelete**

#### **\Undelete**

You, of course, want to know what you can do with this data. By scanning through the value entries in these sub-keys, I find that Diskeeper is installed at D:\ExecSoft\Diskeeper (from \Diskeeper), that it is version 3.0 build 172 (from \CurrentVersion) , it was upgraded from version 2.0 (from \2.0) , and that it is set to run at the lowest priority (from \UserSettings).

Much of the data may not be understandable, but at the least, you can find where the files are. When an Uninstall fails, this is where you find the information to manually uninstall a package. But there are many tips & tricks for these keys. For example, under **\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall** you will find all of the applications listed in Control Panel \Add/Remove Programs. If you have something listed which really isn't on your system, you can delete it here. Of course, you can also delete any other entry, which means you would not be able to run Add/Remove on that package, so be careful!

## **\HKEY\_LOCAL\_MACHINE\SECURITY**

# WINDOWS NT REGISTRY

By Lance Jensen

This hive contains the security information for the local machine. This information includes all group names, all user names and passwords, what rights each user has and what groups each user belongs to. It is maintained via User Manager.

The information is encrypted and is stored in binary format, so you can't edit it with REGEDT32 or REGEDIT. About the only thing you can do is view the user and group names.

## **\HKEY\_LOCAL\_MACHINE\SAM**

In addition to the **\SECURITY** key, **\HKEY\_LOCAL\_MACHINE** also has a **\SAM** (Security Accounts Manager) key. It contains only one sub-key, **\SAM**, which is mapped to the sub-key **\SAM** under **\SECURITY**. Thus any change made to one sub-key also changes the other. However, as in **\SECURITY**, most of the information is encrypted and stored in binary format.

The first sub-key of **\SAM** is **\Domains**. It has two sub-keys, **\Account** and **\Builtin**, and they each have three sub-keys, **\Aliases**, **\Groups** and **\Users**. Each of these has a code-number sub-key for each member (if any), plus **\Names**, which contains as sub-keys the actual names of the members (such as **\Administrators** or **\Users**). **\Account\Users\Names** will contain the names of user accounts, as maintained in the User Manager program. **\Builtin\Aliases\Names** will contain the built-in groups **\Administrators**, **\Backup Operators**, **\Guests**, **\Power Users**, **\Replicator** and **\Users**.

The other sub-key is **\RXACT**, which stands for Registry Transaction. It's usually empty.

## **\HKEY\_USERS**

This hive has two keys. One is a code string which identifies you, the current user. It contains many Control Panel values. Since **\HKEY\_CURRENT\_USER** points to this key, I won't detail the sub-keys here.

The other key, **\DEFAULT**, is very similar to the first key, but does not contain all the same data, and has some different sub-keys & values. These are the default settings for most of the Control Panel options.

## **\HKEY\_CURRENT\_USER**

This is almost entirely Control Panel data. Basically, these data define how Windows NT looks and runs when you (the "current user") are logged in.

### **\AppEvents**

**\AppEvents** contains two sub-keys, **\EventLabels** and **\Schemes**. Under **\EventLabels** are subkeys which are Windows events such as minimizing or maximizing windows. These are the events which you can assign commands to in the Control Panel **\Sounds** window. Each sub-key has a REG\_SZ value entry whose value is the label of that event. For example, the sub-key **\MailBeep** has the label "New Mail Notification".

The other sub-key, **\Schemes**, contains two sub-keys, **\Apps** and **\Names**. Under **\Apps** will be the sub-key **\Default**, plus sub-keys for specific applications such as **\Explorer** and **\Office97**. Under **\Default** will be a series of sub-keys corresponding to those under **\EventLabels**. These sub-keys do not have value entries; instead, they have further sub-keys for each sound scheme that has been defined in Control Panel **\Sounds**, plus **\current**. It is under these sub-keys that

# WINDOWS NT REGISTRY

By Lance Jensen

you find a REG\_SZ value entry whose value is the name of the sound file associated with the event. You could change the file name from within the Registry, but why bother when you can do it so much more easily in "Sounds" in Control Panel? About the only thing it makes sense to do here is to review a Sounds scheme more swiftly than you could in "Sounds".

The other **\Apps** sub-keys for specific applications hold sub-keys for application-specific events, and have the same structure as the sub-keys under **\Apps \Default**.

**\Schemes \Names** has the same sub-keys as you find under any **\Schemes \Apps \Default** sub-key. They contain the actual names of the various Sounds schemes.

## **\Console**

The console is a Windows NT function that emulates MS-DOS functionality, allowing you to run MS-DOS programs and issue DOS level commands. It can be quite useful in troubleshooting a system. The sub-keys of **\Console** define the console screen, font, layout, colors, etc. The values are controlled through Control Panel **\Console**. Instructions on what you can change and how to do it can be found in Help by clicking the Index tab and typing "command prompt windows". Then click Display and select the subject you want.

## **\Control Panel**

This is where most of your Control Panel settings are stored. There are a couple of dozen sub-keys, some with sub-keys of their own. Since they are all maintained through Control Panel, you should not change anything here.

## **\Environment**

User variables maintained through Control Panel **\System \Environments** are stored here. You should have at least the definitions for Temp and Tmp, associating them with the **\Temp** folder.

## **\Keyboard Layout**

There are two subkeys here, **\Preload** and **\Substitutes**, whose value entries contain codes for the keyboard layouts defined for the current user. The keyboard codes are defined in subkeys under **\HKEY\_LOCAL\_MACHINE \SYSTEM \CurrentControlSet \Control \Keyboard Layouts**. This is maintained through Control Panel **\Keyboard**.

## **\Network**

If you are connected to a network, you will have this sub-key. The sub-keys of **\Network** specify the shared directories and devices to which File Manager will connect your system when you log on. Each key will have some or all of the value entries ConnectionType, ProviderName, ProviderType, RemotePath and UserName.

## **\Printers**

Here we have three sub-keys, **\Connections**, **\DevModes2** and **\Settings**, which store the data you enter in Control Panel **\Printers**.

## **\Software**

As in **\HKEY\_LOCAL\_MACHINE \SOFTWARE**, the sub-keys and value entries here contain data for some of the software installed on your system. Each software package may appear as a sub-key of **\SOFTWARE**, but there will also be sub-keys which are manufacturers (such as Microsoft

# WINDOWS NT REGISTRY

## By Lance Jensen

or Executive Software) with software packages listed as sub-keys below the company sub-key. The data under the software sub-keys can anything the software manufacturer wants. Browsing through the data may or may not be useful, but you certainly don't want to change anything!

### **UNICODE Program Groups**

The sub-keys here contain data regarding program groups such as you see on clicking the Start button. The data is all in binary format, so there is nothing worth viewing.

### **The Menu Bar**

We really ought to cover the Menu Bar options, just for completeness. Registry starts with Open Local and Close, which simply open and close the registry files. Load Hive and Restore are used to load data from a disk file; you get the usual Windows view of the directory tree and contents, and select the file you want. Likewise, Unload Hive and Save Key are used to write data to disk files. Select Computer is used to choose any computer on the network (assuming you have the permissions to do so) and access the Registry on that machine. The rest of the options are obvious: Print Subtree, Printer Setup, Save Subtree as, and Exit.

Edit gives you the options Add Key, Add Value and Delete. If you have a value entry highlighted, you can also use the options Binary, String, DWORD and Multi String; these display the value of the value entry in the format selected.

Tree can be used to expand and collapse the viewed tree. Personally, I have no use for it, finding it much more convenient to just double-click on a key within the tree.

View allows you to display the tree alone, or the data alone, or both together. It has a Split option, used to position the "split" between tree and data, though you can also move the division line by doing a click-and-hold. There are also options to display the value of any value entry as binary data, to refresh the active hive, or to refresh all hives. Most valuable, though, is Find Key, which allows a search of the entire hive for a specific key.

Security allows you to set permissions on any hive, and to audit access and changes to the Registry. In addition to the obvious use of finding out who is doing what with the registry, you can use this as a log of whatever changes you make.

Options allows you to change the registry font, and to toggle Auto-Refresh, Confirm on Delete, and Save on Exit, but the most important toggle is Read-Only Mode. You should keep this one set, and only change it when you are going to make a change. Setting Read-Only will not prevent you from making changes outside of the editor. Window gives you the usual display options (Cascade, Tile and Arrange), and lets you pick a particular hive. Help has the standard Windows Help options.