Understanding the Registry on Windows

Windows Registry is a hierarchical database that stores configuration settings and options on Microsoft Windows operating systems. It contains settings for low-level operating system components as well as the applications running on the platform: the kernel, device drivers, services, SAM, user interface and third party applications all make use of the registry. The registry also provides a means to access counters for profiling system performance.

The registry contains two basic elements: keys and values.

Keys

Registry keys are similar to folders — in addition to values, each key can contain subkeys, which may contain further subkeys, and so on. Keys are referenced with a syntax similar to Windows' path names, using backslashes to indicate levels of hierarchy. Each subkey has a mandatory name, which is a non-empty string that cannot contain any backslash or null character, and whose letter case is insignificant.

The hierarchy of registry keys can only be accessed from a known root key handle (which is anonymous but whose effective value is a constant numeric handle) that is mapped to the content of a registry key preloaded by the kernel from a stored "hive", or to the content of a subkey within another root key, or mapped to a registered service or DLL that provide access to its contained subkeys and values.

Example

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows refers to the subkey "Windows" of the subkey "Microsoft" of the subkey "Software" of the HKEY_LOCAL_MACHINE root key.

There are seven predefined root keys:

- HKEY_LOCAL_MACHINE or HKLM
- HKEY_CURRENT_CONFIG or HKCC (only in Windows 9x/ME and NT-based versions of Windows)
- HKEY_CLASSES_ROOT or HKCR
- HKEY_CURRENT_USER or HKCU
- HKEY_USERS or HKU
- HKEY_PERFORMANCE_DATA (only in NT-based versions of Windows, but invisible in the Windows Registry Editor)
- HKEY_DYN_DATA (only in Windows 9x/ME, and visible in the Windows Registry Editor)

Like other files and services in Windows, all registry keys may be restricted by access control lists (ACLs), depending on user privileges, or on security tokens acquired by applications, or on system security policies enforced by the system (these restrictions may be predefined by the system itself, and configured by local system administrators or by domain administrators). Different users, programs, services or remote systems may only see some parts of the hierarchy or distinct hierarchies from the same root keys.

- HKEY_LOCAL_MACHINE (HKLM)—The key located by HKLM is actually not stored on disk, but maintained in memory by the system kernel in order to map there all other subkeys. Applications cannot create any additional subkeys. On NT-based versions of Windows, this key contains four subkeys, "SAM", "SECURITY", "SYSTEM", and "SOFTWARE", that are loaded at boot time within their respective files located in the %SystemRoot%\System32\config folder. A fifth subkey, "HARDWARE", is volatile and is created dynamically, and as such is not stored in a file (it exposes a view of all the currently detected Plug-and-Play devices). On Windows Vista, Windows Server 2008, Windows Server 2008 R2, and Windows 7, a sixth subkey is mapped in memory by the kernel and populated from boot configuration data (BCD).
- "HKLMSAM" key—usually appears as empty for most users (unless they are granted access by administrators of the local system or administrators of domains managing the local system). It is used to reference all "Security and Accounts Management" (SAM) databases for all domains into which the local system has been administratively authorized or configured (including the local domain of the running system, whose SAM database is stored a subkey also named "SAM": other subkeys will be created as needed, one for each supplementary domain).
- Each SAM database contains all built-in accounts (mostly group aliases) and configured accounts (users, groups and their aliases, including guest accounts and administrator accounts) created and configured on the respective domain, for each account in that domain, it notably contains the user name which can be used to log on that domain, the internal unique user identifier in the domain, their cryptographically hashed password on that domain, the location of storage of their user registry hive, various status flags (for example if the account can be enumerated and be visible in the logon prompt screen), and the list of domains (including the local domain) into which the account was configured.
- "HKLMSecurity" key—usually appears empty for most users (unless they are granted access by users with administrative privileges) and is linked to the Security database of the domain into which the current user is logged on (if the user is logged on the local system domain, this key will be linked to the registry hive stored by the local machine and managed by local system administrators or by the built-in "System" account and Windows installers). The kernel will access it to read and enforce the security policy applicable to the current user and all applications or operations executed by this user. It also contains a "SAM" subkey which is dynamically linked to the SAM database of the domain onto which the current user is logged on.
- "HKLMsystem" key—normally only writable by users with administrative privileges on the local system. It contains information about the Windows system setup, data for the secure random number generator (RNG), the list of currently mounted devices containing a file system, several numbered "HKLMsystem\Control Sets" containing alternative configurations for system hardware drivers and services running on the local system (including the currently used one and one backup), a "HKLMsystem\Select" subkey containing the status of these Control Sets, and a "HKLMsystem\CurrentControlSet" which is dynamically linked at boot time to the Control Set which is currently used on the local system. Each configured Control Set contains:
  - "Enum" subkey enumerating all known Plug-and-Play devices and associating them with installed system drivers (and storing the device-specific configurations of these drivers).
  - "Services" subkey listing all installed system drivers (with non device-specific configuration, and the enumeration of devices for which they are instantiated) and all programs running as services (how and when they can be automatically started).
  - "Control" subkey organizing the various hardware drivers and programs running as services and all other system-wide configuration.
Registry values are name/data pairs stored within keys. Registry values are referenced separately from registry keys. Each registry value stored in a registry key has a unique name whose letter case is not significant. The Windows API functions that query and manipulate registry values take variable-length names separately from the key path and/or handle that identifies the parent key. Registry values may contain backslashes in their name but doing so makes them difficult to distinguish from their key paths when using some legacy Windows Registry API functions (whose usage is deprecated in Win32).

The terminology is somewhat misleading, as each registry key is similar to an associative array, where standard terminology would refer to the name part of each registry value as a “key”. The terms are a holdover from the 16-bit registry in Windows 3, in which registry keys could not contain arbitrary name/data pairs, but rather contained only one unnamed value (which had to be a string). In this sense, the entire registry was like a single associative array where the registry keys (in both the registry sense and dictionary sense) formed a hierarchy, and the registry values were all strings. When the 32-bit registry was created, so was the additional capability of creating multiple named values per key, and the meanings of the names were somewhat distorted.

Compatibility with the previous behavior, each registry key may have a “default” value, whose name is the empty string.

Each value can store arbitrary data with variable length and encoding, but which is associated with a symbolic type (defined as a numeric constant) defining how to parse this data. The standard types are:

<table>
<thead>
<tr>
<th>Type ID</th>
<th>Symbolic type name</th>
<th>Meaning and encoding of the data stored in the registry value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>REG_NONE</td>
<td>No type (the stored value, if any)</td>
</tr>
<tr>
<td>1</td>
<td>REG_SZ</td>
<td>A string value, normally stored and exposed in UTF-16LE (when using the Unicode version of Win32 API functions), usually terminated by a null character</td>
</tr>
<tr>
<td>2</td>
<td>REG_EXPAND_SZ</td>
<td>An “expandable” string value that can contain environment variables, normally stored and exposed in UTF-16LE, usually terminated by a null character</td>
</tr>
<tr>
<td>3</td>
<td>REG_BINARY</td>
<td>Binary data (any arbitrary data)</td>
</tr>
<tr>
<td>4</td>
<td>REG_DWORD / REG_DWORD_LITTLE_ENDIAN</td>
<td>A DWORD value, a 32-bit unsigned integer (numbers between 0 and 4,294,967,295 [2^32 – 1]) (little-endian)</td>
</tr>
<tr>
<td>5</td>
<td>REG_DWORD_BIG_ENDIAN</td>
<td>A DWORD value, a 32-bit unsigned integer (numbers between 0 and 4,294,967,295 [2^32 – 1]) (big-endian)</td>
</tr>
<tr>
<td>6</td>
<td>REG_LINK</td>
<td>A symbolic link (UNICODE) to another registry key, specifying a root key and the path to the target key</td>
</tr>
<tr>
<td>7</td>
<td>REG_MULTI_SZ</td>
<td>A multi-string value, which is an ordered list of non-empty strings, normally stored and exposed in UTF-16LE, each one terminated by a null character, the list being normally terminated by a second null character.</td>
</tr>
</tbody>
</table>
Hives

The Registry comprises a number of logical sections, or “hives” (the word hive constitutes an in-joke). Hives are generally named by their Windows API definitions, which all begin “HKEY”. They are frequently abbreviated to a three- or four-letter short name starting with “HK” (e.g. HKCU and HKLM). Technically, they are predefined handles (with known constant values) to specific keys that are either maintained in memory, or stored in hive files stored in the local file system and loaded by the system kernel at boot time and then shared (with various access rights) between all processes running on the local system, or loaded and mapped in all processes started in a user session when the user logs on the system.

The HKEY_LOCAL_MACHINE (local machine-specific configuration data) and HKEY_CURRENT_USER (user-specific configuration data) nodes have a similar structure to each other; user applications typically look up their settings by first checking for them in “HKEY_CURRENT_USERSoftwareVendor’s nameApplication’s nameVersionSetting name”, and if the setting is not found, look instead in the same location under the HKEY_LOCAL_MACHINE key. However, the converse may apply for administrator-enforced policy settings where HKLM may take precedence over HKCU. The Windows Logo Program has specific requirements for where different types of user data may be stored, and that the concept of least privilege be followed so that administrator-level access is not required to use an application.

Related Articles

- Understanding the Registry on Windows (IT Knowledge Base)
  - informational
  - windows
  - registry
  - keys

- Granting Editing Privileges for Kaltura Videos (IT Knowledge Base)
  - kaltura
  - edit
  - editing
  - permission
  - granting
  - permission-granting
  - media
  - video
  - collaboration
  - kb-how-to-article

- Mapping the Q: Drive on Windows (IT Knowledge Base)
  - how_to
  - q_drive
  - shared_resource
  - windows
  - kb-how-to-article
  - drive
  - drive_mapping

- Configuring Remote Desktop on Windows (IT Knowledge Base)
  - remote_desktop
  - windows
  - installing
  - kb-how-to-article

- Configuring Remote Desktop on iOS (IT Knowledge Base)
  - remote
  - ios
  - windows
  - kb-how-to-article

- Windows 7 Support (IT Knowledge Base)
  - windows_7
  - windows
  - microsoft

- Using SafeAssign in HuskyCT (IT Knowledge Base)
  - safeassign
  - huskyct
  - blackboard
• Setting Up Google Calendar in Outlook 2016 for Windows (IT Knowledge Base)
  • calendar
  • google_calendar
  • google
  • g_suite
  • outlook
  • windows
  • kb-how-to-article

• Setting up and Using Skype for Business (IT Knowledge Base)
  • skype
  • skype_for_business
  • windows
  • mac
  • faculty
  • staff
  • kb-how-to-article

• Using Calendar in Outlook for Windows (IT Knowledge Base)
  • calendar
  • outlook
  • windows
  • meeting
  • appointment
  • shared_calendar
  • kb-how-to-article

• Connecting Windows to a Projector (IT Knowledge Base)
  • projector
  • windows
  • technology
  • classroom
  • kb-how-to-article

• Finding the IP Address on Windows 10 (IT Knowledge Base)
  • ip_address
  • ip
  • windows_10
  • windows
  • network
  • internet
  • kb-how-to-article

• Accessing a Shared Mailbox in Outlook on Windows (IT Knowledge Base)
  • windows
  • email
  • mailbox
  • outlook
  • shared
  • kb-how-to-article

• Taking a Screenshot on Mac (IT Knowledge Base)
  • mac
  • windows
  • screenshot
  • android
  • iphone
  • kb-how-to-article

• Repairing UCONN-SECURE Wireless on Windows (IT Knowledge Base)
  • internet
  • wireless
  • connection
  • windows
  • repair
  • connect
  • kb-troubleshooting-article