

EN 此内容没有您的语言版本，但有英语版本。

Predefined Keys

An application must open a key before it can add data to the registry. To open a key, an application must supply a handle to another key in the registry that is already open. The system defines predefined keys that are always open. Predefined keys help an application navigate in the registry and make it possible to develop tools that allow a system administrator to manipulate categories of data. Applications that add data to the registry should always work within the framework of predefined keys, so administrative tools can find and use the new data.

An application can use handles to these keys as entry points to the registry. These handles are valid for all implementations of the registry, although the use of the handles may vary from platform to platform. In addition, other predefined handles have been defined for specific platforms. The following are handles to the predefined keys.

Handle	Description
HKEY_CLASSES_ROOT	<p>Registry entries subordinate to this key define types (or classes) of documents and the properties associated with those types. Shell and COM applications use the information stored under this key.</p> <p>This key also provides backward compatibility with the Windows 3.1 registration database by storing information for DDE and OLE support. File viewers and user interface extensions store their OLE class identifiers in HKEY_CLASSES_ROOT, and in-process servers are registered in this key.</p> <p>This handle should not be used in a service or an application that impersonates different users.</p> <p>For more information, see HKEY_CLASSES_ROOT.</p>
HKEY_CURRENT_CONFIG	<p>Contains information about the current hardware profile of the local computer system. The information under HKEY_CURRENT_CONFIG describes only the differences between the current hardware configuration and the standard configuration. Information about the standard hardware configuration is stored under the Software and System keys of HKEY_LOCAL_MACHINE.</p> <p>HKEY_CURRENT_CONFIG is an alias for HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\Current.</p> <p>For more information, see HKEY_CURRENT_CONFIG.</p>
HKEY_CURRENT_USER	<p>Registry entries subordinate to this key define the preferences of the current user. These preferences include the settings of environment variables, data about program groups, colors, printers, network connections, and application preferences. This key makes it easier to establish the current user's settings; the key maps to the current user's branch in</p>

	<p>HKEY_USERS. In HKEY_CURRENT_USER, software vendors store the current user-specific preferences to be used within their applications. Microsoft, for example, creates the HKEY_CURRENT_USER\Software\Microsoft key for its applications to use, with each application creating its own subkey under the Microsoft key.</p> <p>The mapping between HKEY_CURRENT_USER and HKEY_USERS is per process and is established the first time the process references HKEY_CURRENT_USER. The mapping is based on the security context of the first thread to reference HKEY_CURRENT_USER. If this security context does not have a registry hive loaded in HKEY_USERS, the mapping is established with HKEY_USERS\Default. After this mapping is established it persists, even if the security context of the thread changes.</p> <p>All registry entries in HKEY_CURRENT_USER except those under HKEY_CURRENT_USER\Software\Classes are included in the per-user registry portion of a roaming user profile. To exclude other entries from a roaming user profile, store them in HKEY_CURRENT_USER_LOCAL_SETTINGS.</p> <p>This handle should not be used in a service or an application that impersonates different users. Instead, call the RegOpenCurrentUser function.</p> <p>For more information, see HKEY_CURRENT_USER.</p>
HKEY_CURRENT_USER_LOCAL_SETTINGS	<p>Registry entries subordinate to this key define preferences of the current user that are local to the machine. These entries are not included in the per-user registry portion of a roaming user profile.</p> <p>Windows Server 2008, Windows Vista, Windows Server 2003 and Windows XP/2000: This key is supported starting with Windows 7 and Windows Server 2008 R2.</p>
HKEY_LOCAL_MACHINE	<p>Registry entries subordinate to this key define the physical state of the computer, including data about the bus type, system memory, and installed hardware and software. It contains subkeys that hold current configuration data, including Plug and Play information (the Enum branch, which includes a complete list of all hardware that has ever been on the system), network logon preferences, network security information, software-related information (such as server names and the location of the server), and other system information.</p> <p>For more information, see HKEY_LOCAL_MACHINE.</p>
HKEY_PERFORMANCE_DATA	<p>Registry entries subordinate to this key allow you to access performance data. The data is not actually stored in the registry; the registry functions cause the system to collect the data from its source.</p>
HKEY_PERFORMANCE_NLSTEXT	<p>Registry entries subordinate to this key reference the text strings that describe counters in the local language of the area in which the computer system is running. These entries are not available to Regedit.exe and Regedt32.exe.</p> <p>Windows 2000: This key is not supported.</p>
HKEY_PERFORMANCE_TEXT	<p>Registry entries subordinate to this key reference the text strings that describe counters in US English. These entries are not available to Regedit.exe and Regedt32.exe.</p> <p>Windows 2000: This key is not supported.</p>

HKEY_USER RS	Registry entries subordinate to this key define the default user configuration for new users on the local computer and the user configuration for the current user.
-------------------------	---

The **RegOverridePredefKey** function enables you to map a predefined registry key to a specified key in the registry. For instance, a software installation program could remap a predefined key before installing a DLL component. This enables the installation program to easily examine the information that the DLL's installation procedure writes to the predefined key.

The **RegDisablePredefinedCache** and **RegDisablePredefinedCacheEx** functions disable handle caching for predefined registry handles. Services that use impersonation should call **RegDisablePredefinedCacheEx** before using predefined registry handles.

The predefined handles are not thread safe. Closing a predefined handle in one thread affects any other threads that are using the handle.