

SRP_Registry

Reads and modifies the Windows registry.

Syntax

```
Result = SRP_Registry(Service, Key, ValueName, Value, Error)
```

Returns

The meaning of the return value depends on the service.

Parameters

Parameter	Description
Service	The action to be performed against the registry. Required.
Key	Registry key to which the service is applied. Required.
ValueName	The name of the target value within the key. Optional. Blank value names can be used to refer to the Key's default value.
Value	Additional data based on the service's needs. Optional. See Remarks for details.
Error	Readable error message if the service fails.

Remarks

The SRP_Registry function simplifies registry handling in OpenInsight. Normally, it takes a handful of Window API calls to do this. With SRP_Registry, you only need this one function.

Service

Much like the Utility function in OpenInsight, the SRP_Registry function provides multiple services which are identified in the Service parameter. The services are:

Service	Description
EXISTS	Determines if a registry key or value exists.
LISTVALUES	Lists a key's values.
LISTKEYS	Lists a key's sub keys.
READ	Reads a value from the registry.
READDEF	Reads a value from the registry returning a default value if not found.
WRITE	Writes a value to the registry.
DELETE	Deletes a key or value from the registry.

Click on the service names above to read more details. Continue below for an explanation of the Key parameter.

Key

The Key parameter identifies the registry key. Think of the registry key as a folder. A registry key can contain values and other keys, just like a folder can contain files and other folders. In fact, the syntax for identifying keys looks just like a file path.

The first part of every key identifies a registry hive. Much like a system drive, such as C:, a registry hive identifies a predefined volume, designed to hold certain kinds of registry keys. There are currently five hives:

Hive	Abbr.	Description
------	-------	-------------

HKEY_CLASSES_ROOT	HKCR	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. 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. This handle should not be used in a service or an application that impersonates different users.
HKEY_CURRENT_USER	HKCU	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 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.
HKEY_LOCAL_MACHINE	HKLM	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.
HKEY_USER	HKU	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.
HKEY_CURRENT_CONFIG	HKCC	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 .

So, every key begins with a hive, which you can identify using either the long name or the abbreviation. From there, each key is separated by a backslash. Here are a couple sample keys:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Help
- HKCU\Software\SRP\SRPEditor

The second key uses the HKCU abbreviation for HKEY_CURRENT_USER.

Keys are not case sensitive, so don't worry if your case doesn't match what you see in RegEdit. Now that we understand how to identify keys, we need to understand values.

ValueName

Values are identified using the ValueName parameter. As we explained earlier, keys are like folders. So, values are like files. A key always has a single unnamed value referred to simply as the default value, yet keys may contain any number of other values.

To identify the default value of a key, set the ValueName parameter to "". To identify a name value, pass that value's name. For instance, the HKCU\Software\SRP\SRPEditor key contains a value named Visible. So, to access that value, set Key to "HKCU\Software\SRP\SRPEditor" and ValueName to "Visible".

Like keys, value names are not case sensitive.