:mod:`winreg` --- Windows registry access

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 4)

Unknown directive type "module".

```
.. module:: winreg
   :platform: Windows
   :synopsis: Routines and objects for manipulating the Windows registry.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 8)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Mark Hammond <MarkH@ActiveState.com>
```

These functions expose the Windows registry API to Python. Instead of using an integer as the registry handle, a :ref:`handle object <handle-object>` is used to ensure that the handles are closed correctly, even if the programmer neglects to explicitly close them.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 12); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 19)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   Several functions in this module used to raise a
   :exc:`WindowsError`, which is now an alias of :exc:`OSError`.
```

Functions

This module offers the following functions:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 31)

Unknown directive type "function".

```
.. function:: CloseKey(hkey)

   Closes a previously opened registry key.  The *hkey* argument specifies a
   previously opened key.

   .. note::

      If *hkey* is not closed using this method (or via :meth:`hkey.Close()
      <PyHKEY.Close>`), it is closed when the *hkey* object is destroyed by
      Python.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 43)

Unknown directive type "function".

```
.. function:: ConnectRegistry(computer_name, key)

   Establishes a connection to a predefined registry handle on another computer,
   and returns a :ref:`handle object <handle-object>`.

   *computer_name* is the name of the remote computer, of the form
   ``r"\\computername"``.  If ``None``, the local computer is used.

   *key* is the predefined handle to connect to.

   The return value is the handle of the opened key. If the function fails, an
   :exc:`OSError` exception is raised.

   .. audit-event:: winreg.ConnectRegistry computer_name,key winreg.ConnectRegistry

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 62)

Unknown directive type "function".

```
.. function:: CreateKey(key, sub_key)

   Creates or opens the specified key, returning a
   :ref:`handle object <handle-object>`.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *sub_key* is a string that names the key this method opens or creates.

   If *key* is one of the predefined keys, *sub_key* may be ``None``. In that
   case, the handle returned is the same key handle passed in to the function.

   If the key already exists, this function opens the existing key.

   The return value is the handle of the opened key. If the function fails, an
   :exc:`OSError` exception is raised.

   .. audit-event:: winreg.CreateKey key,sub_key,access winreg.CreateKey

   .. audit-event:: winreg.OpenKey/result key winreg.CreateKey

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 88)

Unknown directive type "function".

```
.. function:: CreateKeyEx(key, sub_key, reserved=0, access=KEY_WRITE)

   Creates or opens the specified key, returning a
   :ref:`handle object <handle-object>`.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *sub_key* is a string that names the key this method opens or creates.

   *reserved* is a reserved integer, and must be zero. The default is zero.

   *access* is an integer that specifies an access mask that describes the desired
   security access for the key.  Default is :const:`KEY_WRITE`.  See
   :ref:`Access Rights <access-rights>` for other allowed values.

   If *key* is one of the predefined keys, *sub_key* may be ``None``. In that
   case, the handle returned is the same key handle passed in to the function.

   If the key already exists, this function opens the existing key.
```

```
   The return value is the handle of the opened key. If the function fails, an
   :exc:`OSError` exception is raised.

   .. audit-event:: winreg.CreateKey key,sub_key,access winreg.CreateKeyEx

   .. audit-event:: winreg.OpenKey/result key winreg.CreateKeyEx

   .. versionadded:: 3.2

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 122)

Unknown directive type "function".

```
.. function:: DeleteKey(key, sub_key)

   Deletes the specified key.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *sub_key* is a string that must be a subkey of the key identified by the *key*
   parameter.  This value must not be ``None``, and the key may not have subkeys.

   *This method can not delete keys with subkeys.*

   If the method succeeds, the entire key, including all of its values, is removed.
   If the method fails, an :exc:`OSError` exception is raised.

   .. audit-event:: winreg.DeleteKey key,sub_key,access winreg.DeleteKey

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 143)

Unknown directive type "function".

```
.. function:: DeleteKeyEx(key, sub_key, access=KEY_WOW64_64KEY, reserved=0)

   Deletes the specified key.

   .. note::
      The :func:`DeleteKeyEx` function is implemented with the RegDeleteKeyEx
      Windows API function, which is specific to 64-bit versions of Windows.
      See the `RegDeleteKeyEx documentation
      <https://msdn.microsoft.com/en-us/library/ms724847%28VS.85%29.aspx>`__.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *sub_key* is a string that must be a subkey of the key identified by the
   *key* parameter. This value must not be ``None``, and the key may not have
   subkeys.

   *reserved* is a reserved integer, and must be zero. The default is zero.

   *access* is an integer that specifies an access mask that describes the desired
   security access for the key.  Default is :const:`KEY_WOW64_64KEY`.  See
   :ref:`Access Rights <access-rights>` for other allowed values.

   *This method can not delete keys with subkeys.*

   If the method succeeds, the entire key, including all of its values, is
   removed. If the method fails, an :exc:`OSError` exception is raised.

   On unsupported Windows versions, :exc:`NotImplementedError` is raised.

   .. audit-event:: winreg.DeleteKey key,sub_key,access winreg.DeleteKeyEx
```

```
   .. versionadded:: 3.2

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 181)

Unknown directive type "function".

```
.. function:: DeleteValue(key, value)

   Removes a named value from a registry key.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *value* is a string that identifies the value to remove.

   .. audit-event:: winreg.DeleteValue key,value winreg.DeleteValue
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 193)

Unknown directive type "function".

```
.. function:: EnumKey(key, index)

   Enumerates subkeys of an open registry key, returning a string.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *index* is an integer that identifies the index of the key to retrieve.

   The function retrieves the name of one subkey each time it is called.  It is
   typically called repeatedly until an :exc:`OSError` exception is
   raised, indicating, no more values are available.

   .. audit-event:: winreg.EnumKey key,index winreg.EnumKey

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 212)

Unknown directive type "function".

```
.. function:: EnumValue(key, index)

   Enumerates values of an open registry key, returning a tuple.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *index* is an integer that identifies the index of the value to retrieve.

   The function retrieves the name of one subkey each time it is called. It is
   typically called repeatedly, until an :exc:`OSError` exception is
   raised, indicating no more values.

   The result is a tuple of 3 items:

   +-------+----------------------------------------+
   | Index | Meaning                                |
   +=======+========================================+
   | ``0`` | A string that identifies the value name    |
   +-------+----------------------------------------+
   | ``1`` | An object that holds the value data, and   |
   |       | whose type depends on the underlying       |
   |       | registry type                              |
```

```
   +-------+-------------------------------------------+
   | ``2`` | An integer that identifies the type of the |
   |       | value data (see table in docs for          |
   |       | :meth:`SetValueEx`)                        |
   +-------+-------------------------------------------+

   .. audit-event:: winreg.EnumValue key,index winreg.EnumValue

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 247)

Unknown directive type "index".

```
.. index::
   single: % (percent); environment variables expansion (Windows)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 250)

Unknown directive type "function".

```
.. function:: ExpandEnvironmentStrings(str)

   Expands environment variable placeholders ``%NAME%`` in strings like
   :const:`REG_EXPAND_SZ`::

      >>> ExpandEnvironmentStrings('%windir%')
      'C:\\Windows'

   .. audit-event:: winreg.ExpandEnvironmentStrings str winreg.ExpandEnvironmentStrings
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 261)

Unknown directive type "function".

```
.. function:: FlushKey(key)

   Writes all the attributes of a key to the registry.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   It is not necessary to call :func:`FlushKey` to change a key. Registry changes are
   flushed to disk by the registry using its lazy flusher.  Registry changes are
   also flushed to disk at system shutdown.  Unlike :func:`CloseKey`, the
   :func:`FlushKey` method returns only when all the data has been written to the
   registry. An application should only call :func:`FlushKey` if it requires
   absolute certainty that registry changes are on disk.

   .. note::

      If you don't know whether a :func:`FlushKey` call is required, it probably
      isn't.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 281)

Unknown directive type "function".

```
.. function:: LoadKey(key, sub_key, file_name)

   Creates a subkey under the specified key and stores registration information
   from a specified file into that subkey.

   *key* is a handle returned by :func:`ConnectRegistry` or one of the constants
   :const:`HKEY_USERS` or :const:`HKEY_LOCAL_MACHINE`.
```

```
   *sub_key* is a string that identifies the subkey to load.

   *file_name* is the name of the file to load registry data from. This file must
   have been created with the :func:`SaveKey` function. Under the file allocation
   table (FAT) file system, the filename may not have an extension.

   A call to :func:`LoadKey` fails if the calling process does not have the
   :const:`SE_RESTORE_PRIVILEGE` privilege.  Note that privileges are different
   from permissions -- see the `RegLoadKey documentation
   <https://msdn.microsoft.com/en-us/library/ms724889%28v=VS.85%29.aspx>`__ for
   more details.

   If *key* is a handle returned by :func:`ConnectRegistry`, then the path
   specified in *file_name* is relative to the remote computer.

   .. audit-event:: winreg.LoadKey key,sub_key,file_name winreg.LoadKey
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 307)

Unknown directive type "function".

```
.. function:: OpenKey(key, sub_key, reserved=0, access=KEY_READ)
              OpenKeyEx(key, sub_key, reserved=0, access=KEY_READ)

   Opens the specified key, returning a :ref:`handle object <handle-object>`.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *sub_key* is a string that identifies the sub_key to open.

   *reserved* is a reserved integer, and must be zero.  The default is zero.

   *access* is an integer that specifies an access mask that describes the desired
   security access for the key.  Default is :const:`KEY_READ`.  See :ref:`Access
   Rights <access-rights>` for other allowed values.

   The result is a new handle to the specified key.

   If the function fails, :exc:`OSError` is raised.

   .. audit-event:: winreg.OpenKey key,sub_key,access winreg.OpenKey

   .. audit-event:: winreg.OpenKey/result key winreg.OpenKey

   .. versionchanged:: 3.2
      Allow the use of named arguments.

   .. versionchanged:: 3.3
      See :ref:`above <exception-changed>`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 338)

Unknown directive type "function".

```
.. function:: QueryInfoKey(key)

   Returns information about a key, as a tuple.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   The result is a tuple of 3 items:

   +-------+---------------------------------------------+
   | Index | Meaning                                     |
   +=======+=============================================+
   | ``0`` | An integer giving the number of sub keys    |
   |       | this key has.                               |
   +-------+---------------------------------------------+
   | ``1`` | An integer giving the number of values this |
   |       | key has.                                    |
```

```
+-------+-----------------------------------------+
| ``2`` | An integer giving when the key was last |
|       | modified (if available) as 100's of     |
|       | nanoseconds since Jan 1, 1601.          |
+-------+-----------------------------------------+

.. audit-event:: winreg.QueryInfoKey key winreg.QueryInfoKey
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 364)

Unknown directive type "function".

```
.. function:: QueryValue(key, sub_key)

   Retrieves the unnamed value for a key, as a string.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *sub_key* is a string that holds the name of the subkey with which the value is
   associated.  If this parameter is ``None`` or empty, the function retrieves the
   value set by the :func:`SetValue` method for the key identified by *key*.

   Values in the registry have name, type, and data components. This method
   retrieves the data for a key's first value that has a ``NULL`` name. But the
   underlying API call doesn't return the type, so always use
   :func:`QueryValueEx` if possible.

   .. audit-event:: winreg.QueryValue key,sub_key,value_name winreg.QueryValue
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 383)

Unknown directive type "function".

```
.. function:: QueryValueEx(key, value_name)

   Retrieves the type and data for a specified value name associated with
   an open registry key.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *value_name* is a string indicating the value to query.

   The result is a tuple of 2 items:

   +-------+-----------------------------------------+
   | Index | Meaning                                 |
   +=======+=========================================+
   | ``0`` | The value of the registry item.         |
   +-------+-----------------------------------------+
   | ``1`` | An integer giving the registry type for |
   |       | this value (see table in docs for       |
   |       | :meth:`SetValueEx`)                     |
   +-------+-----------------------------------------+

   .. audit-event:: winreg.QueryValue key,sub_key,value_name winreg.QueryValueEx
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 408)

Unknown directive type "function".

```
.. function:: SaveKey(key, file_name)

   Saves the specified key, and all its subkeys to the specified file.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.
```

```
   *file_name* is the name of the file to save registry data to.  This file
   cannot already exist. If this filename includes an extension, it cannot be
   used on file allocation table (FAT) file systems by the :meth:`LoadKey`
   method.

   If *key* represents a key on a remote computer, the path described by
   *file_name* is relative to the remote computer. The caller of this method must
   possess the :const:`SeBackupPrivilege` security privilege.  Note that
   privileges are different than permissions -- see the
   `Conflicts Between User Rights and Permissions documentation
   <https://msdn.microsoft.com/en-us/library/ms724878%28v=VS.85%29.aspx>`__
   for more details.

   This function passes ``NULL`` for *security_attributes* to the API.

   .. audit-event:: winreg.SaveKey key,file_name winreg.SaveKey
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 433)

Unknown directive type "function".

```
.. function:: SetValue(key, sub_key, type, value)

   Associates a value with a specified key.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *sub_key* is a string that names the subkey with which the value is associated.

   *type* is an integer that specifies the type of the data. Currently this must be
   :const:`REG_SZ`, meaning only strings are supported.  Use the :func:`SetValueEx`
   function for support for other data types.

   *value* is a string that specifies the new value.

   If the key specified by the *sub_key* parameter does not exist, the SetValue
   function creates it.

   Value lengths are limited by available memory. Long values (more than 2048
   bytes) should be stored as files with the filenames stored in the configuration
   registry.  This helps the registry perform efficiently.

   The key identified by the *key* parameter must have been opened with
   :const:`KEY_SET_VALUE` access.

   .. audit-event:: winreg.SetValue key,sub_key,type,value winreg.SetValue
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 461)

Unknown directive type "function".

```
.. function:: SetValueEx(key, value_name, reserved, type, value)

   Stores data in the value field of an open registry key.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   *value_name* is a string that names the subkey with which the value is
   associated.

   *reserved* can be anything -- zero is always passed to the API.

   *type* is an integer that specifies the type of the data. See
   :ref:`Value Types <value-types>` for the available types.

   *value* is a string that specifies the new value.

   This method can also set additional value and type information for the specified
   key.  The key identified by the key parameter must have been opened with
   :const:`KEY_SET_VALUE` access.
```

```
   To open the key, use the :func:`CreateKey` or :func:`OpenKey` methods.

   Value lengths are limited by available memory. Long values (more than 2048
   bytes) should be stored as files with the filenames stored in the configuration
   registry.  This helps the registry perform efficiently.

   .. audit-event:: winreg.SetValue key,sub_key,type,value winreg.SetValueEx
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 491)

Unknown directive type "function".

```
.. function:: DisableReflectionKey(key)

   Disables registry reflection for 32-bit processes running on a 64-bit
   operating system.

   *key* is an already open key, or one of the predefined :ref:`HKEY_* constants
   <hkey-constants>`.

   Will generally raise :exc:`NotImplementedError` if executed on a 32-bit operating
   system.

   If the key is not on the reflection list, the function succeeds but has no
   effect.  Disabling reflection for a key does not affect reflection of any
   subkeys.

   .. audit-event:: winreg.DisableReflectionKey key winreg.DisableReflectionKey
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 509)

Unknown directive type "function".

```
.. function:: EnableReflectionKey(key)

   Restores registry reflection for the specified disabled key.

   *key* is an already open key, or one of the predefined :ref:`HKEY_* constants
   <hkey-constants>`.

   Will generally raise :exc:`NotImplementedError` if executed on a 32-bit operating
   system.

   Restoring reflection for a key does not affect reflection of any subkeys.

   .. audit-event:: winreg.EnableReflectionKey key winreg.EnableReflectionKey
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 524)

Unknown directive type "function".

```
.. function:: QueryReflectionKey(key)

   Determines the reflection state for the specified key.

   *key* is an already open key, or one of the predefined
   :ref:`HKEY_* constants <hkey-constants>`.

   Returns ``True`` if reflection is disabled.

   Will generally raise :exc:`NotImplementedError` if executed on a 32-bit
   operating system.

   .. audit-event:: winreg.QueryReflectionKey key winreg.QueryReflectionKey
```

Constants

The following constants are defined for use in many :mod:`_winreg` functions.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 544); *backlink*

Unknown interpreted text role "mod".

HKEY_* Constants

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 551)

Unknown directive type "data".

```
.. data:: HKEY_CLASSES_ROOT

   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.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 558)

Unknown directive type "data".

```
.. data:: HKEY_CURRENT_USER

   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.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 565)

Unknown directive type "data".

```
.. data:: HKEY_LOCAL_MACHINE

   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.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 571)

Unknown directive type "data".

```
.. data:: HKEY_USERS

   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.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 577)

Unknown directive type "data".

```
.. data:: HKEY_PERFORMANCE_DATA

   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.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 585)

Unknown directive type "data".

```
.. data:: HKEY_CURRENT_CONFIG

   Contains information about the current hardware profile of the
   local computer system.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 590)

Unknown directive type "data".

```
.. data:: HKEY_DYN_DATA

   This key is not used in versions of Windows after 98.
```

Access Rights

For more information, see Registry Key Security and Access.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 603)

Unknown directive type "data".

```
.. data:: KEY_ALL_ACCESS

   Combines the STANDARD_RIGHTS_REQUIRED, :const:`KEY_QUERY_VALUE`,
   :const:`KEY_SET_VALUE`, :const:`KEY_CREATE_SUB_KEY`,
   :const:`KEY_ENUMERATE_SUB_KEYS`, :const:`KEY_NOTIFY`,
   and :const:`KEY_CREATE_LINK` access rights.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 610)

Unknown directive type "data".

```
.. data:: KEY_WRITE

   Combines the STANDARD_RIGHTS_WRITE, :const:`KEY_SET_VALUE`, and
   :const:`KEY_CREATE_SUB_KEY` access rights.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 615)

Unknown directive type "data".

```
.. data:: KEY_READ

   Combines the STANDARD_RIGHTS_READ, :const:`KEY_QUERY_VALUE`,
   :const:`KEY_ENUMERATE_SUB_KEYS`, and :const:`KEY_NOTIFY` values.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 620)

Unknown directive type "data".

```
.. data:: KEY_EXECUTE

   Equivalent to :const:`KEY_READ`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 624)

Unknown directive type "data".

```
.. data:: KEY_QUERY_VALUE

   Required to query the values of a registry key.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 628)

Unknown directive type "data".

```
.. data:: KEY_SET_VALUE

   Required to create, delete, or set a registry value.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 632)

Unknown directive type "data".

```
.. data:: KEY_CREATE_SUB_KEY

   Required to create a subkey of a registry key.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 636)

Unknown directive type "data".

```
.. data:: KEY_ENUMERATE_SUB_KEYS

   Required to enumerate the subkeys of a registry key.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 640)

Unknown directive type "data".

```
.. data:: KEY_NOTIFY

   Required to request change notifications for a registry key or for
   subkeys of a registry key.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 645)

Unknown directive type "data".

```
.. data:: KEY_CREATE_LINK

   Reserved for system use.
```

64-bit Specific

For more information, see Accessing an Alternate Registry View.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 658)

Unknown directive type "data".

```
.. data:: KEY_WOW64_64KEY

   Indicates that an application on 64-bit Windows should operate on
   the 64-bit registry view.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 663)

Unknown directive type "data".

```
.. data:: KEY_WOW64_32KEY

   Indicates that an application on 64-bit Windows should operate on
   the 32-bit registry view.
```

Value Types

For more information, see Registry Value Types.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 677)

Unknown directive type "data".

```
.. data:: REG_BINARY

   Binary data in any form.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 681)

Unknown directive type "data".

```
.. data:: REG_DWORD

   32-bit number.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 685)

Unknown directive type "data".

```
.. data:: REG_DWORD_LITTLE_ENDIAN

   A 32-bit number in little-endian format. Equivalent to :const:`REG_DWORD`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 689)

Unknown directive type "data".

```
.. data:: REG_DWORD_BIG_ENDIAN

   A 32-bit number in big-endian format.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 693)

Unknown directive type "data".

```
.. data:: REG_EXPAND_SZ

   Null-terminated string containing references to environment
   variables (``%PATH%``).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 698)

Unknown directive type "data".

```
.. data:: REG_LINK
```

```
   A Unicode symbolic link.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 702)

Unknown directive type "data".

```
.. data:: REG_MULTI_SZ

   A sequence of null-terminated strings, terminated by two null characters.
   (Python handles this termination automatically.)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 707)

Unknown directive type "data".

```
.. data:: REG_NONE

   No defined value type.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 711)

Unknown directive type "data".

```
.. data:: REG_QWORD

   A 64-bit number.

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 717)

Unknown directive type "data".

```
.. data:: REG_QWORD_LITTLE_ENDIAN

   A 64-bit number in little-endian format. Equivalent to :const:`REG_QWORD`.

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 723)

Unknown directive type "data".

```
.. data:: REG_RESOURCE_LIST

   A device-driver resource list.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 727)

Unknown directive type "data".

```
.. data:: REG_FULL_RESOURCE_DESCRIPTOR

   A hardware setting.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 731)

Unknown directive type "data".

```
.. data:: REG_RESOURCE_REQUIREMENTS_LIST
```

```
   A hardware resource list.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 735)

Unknown directive type "data".

```
.. data:: REG_SZ

   A null-terminated string.
```

Registry Handle Objects

This object wraps a Windows HKEY object, automatically closing it when the object is destroyed. To guarantee cleanup, you can call either the :meth:`~PyHKEY.Close` method on the object, or the :func:`CloseKey` function.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 745); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 745); *backlink*

Unknown interpreted text role "func".

All registry functions in this module return one of these objects.

All registry functions in this module which accept a handle object also accept an integer, however, use of the handle object is encouraged.

Handle objects provide semantics for :meth:`__bool__` -- thus

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 754); *backlink*

Unknown interpreted text role "meth".

```
if handle:
    print("Yes")
```

will print `Yes` if the handle is currently valid (has not been closed or detached).

The object also support comparison semantics, so handle objects will compare true if they both reference the same underlying Windows handle value.

Handle objects can be converted to an integer (e.g., using the built-in :func:`int` function), in which case the underlying Windows handle value is returned. You can also use the :meth:`~PyHKEY.Detach` method to return the integer handle, and also disconnect the Windows handle from the handle object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 765); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 765); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 771)

Unknown directive type "method".

```
.. method:: PyHKEY.Close()
```

```
   Closes the underlying Windows handle.

   If the handle is already closed, no error is raised.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 778)

Unknown directive type "method".

```
.. method:: PyHKEY.Detach()

   Detaches the Windows handle from the handle object.

   The result is an integer that holds the value of the handle before it is
   detached.  If the handle is already detached or closed, this will return
   zero.

   After calling this function, the handle is effectively invalidated, but the
   handle is not closed.  You would call this function when you need the
   underlying Win32 handle to exist beyond the lifetime of the handle object.

   .. audit-event:: winreg.PyHKEY.Detach key winreg.PyHKEY.Detach
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]winreg.rst, line 793)

Unknown directive type "method".

```
.. method:: PyHKEY.__enter__()
            PyHKEY.__exit__(*exc_info)

   The HKEY object implements :meth:`~object.__enter__` and
   :meth:`~object.__exit__` and thus supports the context protocol for the
   :keyword:`with` statement::

      with OpenKey(HKEY_LOCAL_MACHINE, "foo") as key:
          ...  # work with key

   will automatically close *key* when control leaves the :keyword:`with` block.
```