

# Reading a Registry Value

The Windows Registry is a centralized database of settings, for Windows hardware, software, and applications. It was introduced in Windows 95 to replace the jumble of .INI files, environment variables, and other settings that were previously required. By centralizing these settings, and providing functions in the Windows API for reading and writing registry values, maintenance of these settings has become considerably simplified.

The code below will read the OpenEngine queue name, which is stored in the Windows Registry in the following area:

Key: **HKEY\_LOCAL\_MACHINE\Software\RevSoft\OpenInsight**

SubKey: **LastServerName**

## The Code

A registry key is "opened" with [RegOpenKeyEx\(\)](#). After the key is opened, a value is read using [RegQueryValueEx\(\)](#). After the key has been read, it should be closed using [RegCloseKey\(\)](#). Below is the code to read the queue name, displaying a message when complete. To test, copy this code to the [C LINK event](#) of a button:

```
declare function RegOpenKeyEx , RegCloseKey
declare subroutine RegQueryValueEx

/* equates for the base registry keys */
equ HKEY_CLASSES_ROOT$      to 0x80000000
equ HKEY_CURRENT_USER$     to 0x80000001
equ HKEY_LOCAL_MACHINE$    to 0x80000002
equ HKEY_USERS$            to 0x80000003
equ HKEY_PERFORMANCE_DATA$ to 0x80000004
equ HKEY_CURRENT_CONFIG$   to 0x80000005
equ HKEY_DYN_DATA$         to 0x80000006

equ KEY_QUERY_VALUE$ to 0x0001
equ ERROR_SUCCESS to 0x0000
options = 0
samDesired = KEY_QUERY_VALUE$
KeyHandle = 0
Hkey = HKEY_LOCAL_MACHINE$
SubKey = "SOFTWARE\RevSoft\OpenInsight":\00\
stat = 0
null = ''
LockVariable KeyHandle as Long
stat = RegOpenKeyEx(Hkey, SubKey, options, samDesired, KeyHandle)
If Stat = ERROR_SUCCESS Then
  QueueName = str(\00\, 512)
  Reg_SZ = 1
  cbBuf = 512
  Key = "LastServerName":\00\
  RegQueryValueEx(KeyHandle, Key, 0, Reg_SZ, QueueName, cbBuf)
  call msg(@window, 'Queue Name = ': QueueName[1, cbBuf - 1])
end
rv = RegCloseKey( KeyHandle)
```

The queue name is returned in **QueueName**. Note how the buffer (**cbBuf**) is initialized to 512 bytes of char(0). When [RegQueryValueEx\(\)](#) returns, **cbBuf** contains the length of the return value, including the char(0) which is the delimiter. The actual queue name is everything except the char(0). This is a common technique in calling Windows APIs that return a variable length string.

## The Windows API Declarations

The code above will not run until the declarations for [RegOpenKeyEx\(\)](#), [RegQueryValueEx\(\)](#), and [RegCloseKey\(\)](#) have been added. To add the declarations, do the following:

1. Log out of the application.
2. Log into the **SYSPROG** application.
3. Add a row, (call it **DLL\_APICALLS\_ADVAPI32**), with the first line as **ADVAPI32** and containing the declarations as shown below.

```
ADVAPI32
LONG STDCALL RegOpenKeyExA(HANDLE, LPCHAR, ULONG, ULONG, LPLONG) AS RegOpenKeyEx
LONG STDCALL RegQueryValueExA(HANDLE, LPCHAR, LPVOID, LPULONG, LPCHAR, LPULONG) AS RegQueryValueEx
LONG STDCALL RegCloseKey(HANDLE)
```

4. Save the row.
5. Run **Declare\_FCNS** at the System Editor Exec Line to create the declaration header, as shown below:

```
RUN DECLARE_FCNS 'DLL_APICALLS_ADVAPI32'
```

6. Exit the editor.
7. Log out of SYSPROG.
8. Log into your application.
9. Run the window.