

Reading the Environment PATH Variable

The Code

The `GetEnvironmentVariable()` Windows API function is used to read an environment variable. The code to read the PATH variable is below. Paste the code in the [CLICK event](#) of a button in a window:

```
declare function GetEnvironmentVariable
VarName = "PATH"
VarLength = GetEnvironmentVariable(VarName, "", 0) + 1
VarValue = space(VarLength+1)
VarLength = GetEnvironmentVariable(VarName, VarValue, VarLength)
VarValue = VarValue[1, VarLength]
call msg(@window , 'Path is ' : VarValue)
```

The code is a bit tricky. Note that there are **two** calls to `GetEnvironmentVariable()`. The reason is that the **PATH** variable is passed as a pointer to a buffer whose size must be determined. The purpose of the first call is to determine the buffer size. The buffer size must be one byte larger than the actual value returned, because the string is terminated by a **Char(0)**. Passing a null string in the second argument, and zero in the third argument, returns the necessary buffer size.

The second call to `GetEnvironmentVariable()` returns the value, including the terminating **Char(0)**. The terminating **Char(0)** is stripped off. **VarValue** will contain the value of the **PATH** environment variable.

The Windows API Declaration

The code above will not run until the declaration for `GetEnvironmentVariable()` has 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_KERNEL32**), with the first line as **KERNEL32** and containing the declarations as shown below.

```
KERNEL32
LONG STDCALL GetEnvironmentVariableA(LPCHAR, LPCHAR, ULONG) As GetEnvironmentVariable
//....add any other declarations in KERNEL32 here.....
```

The function is aliased to its the ANSI version **GetEnvironmentVariableA()**.

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_KERNEL32'
```

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