

# A brief history of the `GetEnvironmentStrings` functions

 [devblogs.microsoft.com/oldnewthing/20130117-00](http://devblogs.microsoft.com/oldnewthing/20130117-00)

January 17, 2013



Raymond Chen

The `GetEnvironmentStrings` function has a long and troubled history.

The first bit of confusion is that the day it was introduced in Windows NT 3.1, it was exported funny. The UNICODE version was exported under the name `GetEnvironmentStringsW`, but the ANSI version was exported under the name `GetEnvironmentStrings` without the usual `A` suffix.

A mistake we have been living with for over two decades.

This is why the `winbase.h` header file contains these confusing lines:

```
WINBASEAPI
LPCH
WINAPI
GetEnvironmentStrings(
    VOID
);
```

```
WINBASEAPI
LPWCH
WINAPI
GetEnvironmentStringsW(
    VOID
);
```

```
#ifdef UNICODE
#define GetEnvironmentStrings GetEnvironmentStringsW
#else
#define GetEnvironmentStringsA GetEnvironmentStrings
#endif // !UNICODE
```

It's trying to clean up a mess that was created long ago, and it only partly succeeds. This is why your IDE may get confused when you try to call the `GetEnvironmentStrings` function and send you to the wrong definition. It's having trouble untangling the macros whose job is

to try to untangle the original mistake.

The kernel folks tried to clean this up as quickly as they could, by exporting new functions with the names `GetEnvironmentStringsW` and `GetEnvironmentStringsA`, like they should have been in the first place, but for compatibility purposes, they still have to export the weird unsuffixed `GetEnvironmentStrings` function. And then to avoid all the “gotcha!”s from people looking for proof of nefarious intent, they kept the mistake in the public header files to make their actions visible to all.

Though personally, I would have tidied things up differently:

```
WINBASEAPI
LPCH
WINAPI
GetEnvironmentStrings(
    VOID
);
```

```
WINBASEAPI
LPCH
WINAPI
GetEnvironmentStringsA(
    VOID
);
```

```
WINBASEAPI
LPWCH
WINAPI
GetEnvironmentStringsW(
    VOID
);
```

```
#ifdef UNICODE
#define GetEnvironmentStrings GetEnvironmentStringsW
#else
#define GetEnvironmentStrings GetEnvironmentStringsA
#endif // !UNICODE
```

I would have left the declaration of the mistaken `GetEnvironmentStrings` function in the header file, but redirected the symbolic name to the preferred suffixed version.

But then again, maybe my version would have confused IDEs even more than the current mechanism does.

The other unfortunate note in the history of the `GetEnvironmentStrings` function is the odd way it handled the Unicode environment. Back in the old days, the `GetEnvironmentStrings` function returned a raw pointer to the environment block. The result was that if some other code modified the environment, your pointer became invalid, and there was nothing you could do about it. As I noted, the function was subsequently changed so that both the ANSI and Unicode versions return *snapshots* of the environment strings, so that the environment strings you received wouldn't get spontaneously corrupted by another thread.

Raymond Chen

**Follow**

