

How is the `CommandLineToArgvW` function intended to be used?

devblogs.microsoft.com/oldnewthing/20100916-00

September 16, 2010



Raymond Chen

The `CommandLineToArgvW` function does some basic command line parsing. A customer reported that it was producing strange results when you passed an empty string as the first parameter:

```
LPWSTR argv = CommandLineToArgvW(L"", &argc);
```

Well, okay, yeah, but huh?

The first parameter to `CommandLineToArgvW` is supposed to be the value returned by `GetCommandLineW`. That's the command line, and that's what `CommandLineToArgvW` was designed to parse. If you pass something else, then `CommandLineToArgvW` will try to cope, but it's not really doing what it was designed for.

It turns out that the customer was mistakenly passing the `lpCmdLine` parameter that was passed to the `wWinMain` function:

```
int WINAPI wWinMain(  
    HINSTANCE hInstance,  
    HINSTANCE hPrevInstance,  
    LPWSTR lpCmdLine,  
    int nCmdShow)  
{  
    int argc;  
    LPWSTR argv = CommandLineToArgvW(lpCmdLine, &argc);  
    ...  
}
```

That command line is not in the format that `CommandLineToArgvW` expects. The `CommandLineToArgvW` function wants the full, unexpurgated command line as returned by the `GetCommandLineW` function, and it breaks it up on the assumption that the first word on the command line is the program name. If you hand it an empty string, the `CommandLineToArgvW` function says, "Whoa, whoever generated this command line totally screwed up. I'll try to muddle through as best I can."

Next time, we'll look at the strange status of quotation marks and backslashes in `CommandLineToArgvW` .

Raymond Chen

Follow

