

# How do I customize the console properties for a shortcut to a console application?

 [devblogs.microsoft.com/oldnewthing/20130527-00](http://devblogs.microsoft.com/oldnewthing/20130527-00)

May 27, 2013



Raymond Chen

You already know how to create a shortcut:

```
#include <windows.h>
#include <tchar.h>
#include <shlobj.h>
#include <atlbase.h>

// class CCoInitialize incorporated here by reference

int __cdecl _tmain(int argc, TCHAR **argv)
{
    // error checking elided for expository purposes
    CCoInitialize init;
    CComPtr<IShellLink> spsl;
    spsl.CoCreateInstance(CLSID_ShellLink);
    spsl->SetPath(TEXT("C:\\Windows\\system32\\cmd.exe"));
    CComQIPtr<IPersistFile>(spsl)->Save(L"Here.lnk", TRUE);
    return 0;
}
```

If you double-click the resulting shortcut from Explorer, it will run the command processor in a default console window.

Today's Little Program customizes the other console properties, so you can control settings like the console buffer size and whether QuickEdit is enabled by default.

We use the `IShellDataList` interface to attach "bonus data" to the shell link. The data we are interested in here is the `NT_CONSOLE_PROPS`. Remember, Little Programs perform little to no error checking, use hard-coded paths, and all that other stuff that make them unsuitable for shipping-quality code.

```

int __cdecl _tmain(int argc, TCHAR **argv)
{
    CCoInitialize init;
    CComPtr<IShellLink> spsl;
    spsl.CoCreateInstance(CLSID_ShellLink);
    spsl->SetPath(TEXT("C:\\Windows\\system32\\cmd.exe"));

    NT_CONSOLE_PROPS props;
    ZeroMemory(&props, sizeof(props));
    props.dbh.cbSize = sizeof(props);
    props.dbh.dwSignature = NT_CONSOLE_PROPS_SIG;
    props.wFillAttribute = FOREGROUND_BLUE | FOREGROUND_GREEN |
        FOREGROUND_RED; // white on black
    props.wPopupFillAttribute = BACKGROUND_BLUE | BACKGROUND_GREEN |
        BACKGROUND_RED | BACKGROUND_INTENSITY |
        FOREGROUND_BLUE | FOREGROUND_RED;
        // purple on white
    props.dwWindowSize.X = 132; // 132 columns wide
    props.dwWindowSize.Y = 50; // 50 lines tall
    props.dwScreenBufferSize.X = 132; // 132 columns wide
    props.dwScreenBufferSize.Y = 1000; // large scrollbar
    props.uCursorSize = 25; // small cursor
    props.bQuickEdit = TRUE; // turn QuickEdit on
    props.bAutoPosition = TRUE;
    props.uHistoryBufferSize = 25;
    props.uNumberOfHistoryBuffers = 4;
    props.ColorTable[ 0] = RGB(0x00, 0x00, 0x00);
    props.ColorTable[ 1] = RGB(0x00, 0x00, 0x80);
    props.ColorTable[ 2] = RGB(0x00, 0x80, 0x00);
    props.ColorTable[ 3] = RGB(0x00, 0x80, 0x80);
    props.ColorTable[ 4] = RGB(0x80, 0x00, 0x00);
    props.ColorTable[ 5] = RGB(0x80, 0x00, 0x80);
    props.ColorTable[ 6] = RGB(0x80, 0x80, 0x00);
    props.ColorTable[ 7] = RGB(0xC0, 0xC0, 0xC0);
    props.ColorTable[ 8] = RGB(0x80, 0x80, 0x80);
    props.ColorTable[ 9] = RGB(0x00, 0x00, 0xFF);
    props.ColorTable[10] = RGB(0x00, 0xFF, 0x00);
    props.ColorTable[11] = RGB(0x00, 0xFF, 0xFF);
    props.ColorTable[12] = RGB(0xFF, 0x00, 0x00);
    props.ColorTable[13] = RGB(0xFF, 0x00, 0xFF);
    props.ColorTable[14] = RGB(0xFF, 0xFF, 0x00);
    props.ColorTable[15] = RGB(0xFF, 0xFF, 0xFF);
    CComQIPtr<IShellLinkDataList>(spsl)->AddDataBlock(&props);

    CComQIPtr<IPersistFile>(spsl)->Save(L"Here.lnk", TRUE);
    return 0;
}

```

Raymond Chen

**Follow**

