# How do I programmatically obtain the user’s selected accent color in Windows 10?

devblogs.microsoft.com/oldnewthing/20170405-00

April 5, 2017

Raymond Chen

If you want to obtain the user’s accent color, you can ask the `UISettings` object for the current color value of `UIColorType.Accent`.

How you get this information depends on what kind of program you’re writing.

If you’re writing a Store app or a classic desktop program in C++/CX, then you do this:

```
namespace vm = Windows::UI::ViewManagement;

void GetAccentColor()
{
    auto settings = ref new vm::UISettings();
    auto color = settings->GetColorValue(vm::UIColorType::Accent);
    // color.A, color.R, color.G, and color.B are the color channels.
}
```

If you’re writing a Store app or a classic desktop program in C#, then you do this:

```
using vm = Windows.UI.ViewManagement;

void GetAccentColor()
{
    var settings = new vm.UISettings();
    var color = settings.GetColorValue(vm.UIColorType.Accent);
    // color.A, color.R, color.G, and color.B are the color channels.
}
```

If you’re writing a Store app in JavaScript, then you do this:

```
var vm = Windows.UI.ViewManagement;

function getAccentColor() {
    var settings = new vm.UISettings();
    var color = settings.getColorValue(vm.UIColorType.accent);
    // color.a, color.r, color.g, and color.b are the color channels.
}
```

If you’re writing a Store app or a classic desktop program in raw C++ (you crazy person you), then you get to do this:

```
namespace abi_vm = ABI::Windows::UI::ViewManagement;
namespace wrl = Microsoft::WRL;
namespace wf = Windows::Foundation;

void GetAccentColor()
{
    // Error checking has been elided for expository purposes.
    wrl::ComPtr<abi_vm::IUISettings> settings;
    wf::ActivateInstance(wrl::Wrappers::HStringReference(
     RuntimeClass_Windows_UI_ViewManagement_UISettings).Get(), &settings);
    ABI::Windows::UI::Color color;
    settings->GetColorValue(abi_vm::UIColorType::Accent, &color);
    // color.A, color.R, color.G, and color.B are the color channels.
}
```

And if you’re writing a Store app or a classic desktop program in C++/WinRT, you write this:

```
namespace winrt_vm = winrt::Windows::UI::ViewManagement;

void GetAccentColor()
{
    winrt_vm::UISettings settings;
    auto color = settings.GetColorValue(winrt_vm::UIColorType::Accent);
    // color.A, color.R, color.G, and color.B are the color channels.
}
```

Note that I used namespace aliases instead of importing the entire namespace. This makes the code a little bit uglier, but I think it’s useful for expository purposes because it makes it clearer which namespace each identifier comes from.

Raymond Chen

**Follow**