How do I trigger an EN_UPDATE notification for all of my edit controls when the user’s locale information changes?

devblogs.microsoft.com/oldnewthing/20180723-00

July 23, 2018

Raymond Chen

A customer had a dialog box with a bunch of edit controls. Some of these edit controls contained floating-point values, and the code parses them by calling a locale-sensitive parser, so that users in the United States (say) can use the period as the decimal marker, whereas users in Germany can use the comma. The user might change the locale settings from US-style to Germany-style, and the customer wants to handle this change by updating the text in all of the edit controls to match the new locale settings.

The customer decided that the way to do this was to handle the `WM_ SETTINGCHANGE` message, and the customer concluded that they could get everything to work if only they could cajole every edit control into generating a `EN_ UPDATE` notification when this happened. The existing handling for this message would finish the work. (I don’t quite see how, but that’s what they said.)

The literal answer to the question is that you can just send the `EN_ UPDATE` notification yourself.

```
void SendFakeEnUpdateNotification(HWND hwndEdit)
{
  FORWARD_WM_COMMAND(
    GetParent(hwndEdit), GetDlgCtrlID(hwndEdit), hwndEdit,
    EN_UPDATE, SendMessage);
}
```

You can find all the edit controls by enumerating them.

```
void GenerateFakeEnUpdateNotificationsForChildWindows(HWND hdlg)
{
  EnumChildWindows(hdlg, [](HWND hwnd, LPARAM lParam)
  {
    auto hdlg = reinterpret_cast<HWND>(lParam);
    wchar_t className[10];
    if (GetClassName(hwnd, className, 10) == 4 &&
        CompareStringOrdinal(className, -1,
                             L"edit", -1, TRUE) == CSTR_EQUAL) {
      FORWARD_WM_COMMAND(
        hdlg, GetDlgCtrlID(hwndEdit), hwndEdit,
        EN_UPDATE, SendMessage);
    }
    return TRUE;
  }, reinterpret_cast<LPARAM>(hdlg));
}
```

This enumerates all the child windows, picks out the edit controls, and generates a fake `EN_ UPDATE` notification on their behalf.

But you can do better.

For example, instead of sending a fake `EN_ UPDATE` notification, you may as well just go straight to the code that handles the notification. If your dialog procedure says

```
  case WM_COMMAND:
    switch (GET_WM_COMMAND_ID(wParam, lParam)) {
    // These three edit controls are the ones that
    // contain decimal values.
    case IDC_VALUE1:
    case IDC_VALUE3:
    case IDC_VALUE5:
      switch (GET_WM_COMMAND_CMD(wParam, lParam)) {
      case EN_UPDATE:
        OnEditUpdate(GET_WM_COMMAND_HWND(wParam, lParam));
        break;
      ...
      }
      break;
      ...
    }
    ...
  }
```

then you can just call your `OnEditUpdate` method directly.

```
void TriggerEditUpdateChildWindows(HWND hdlg)
{
  EnumChildWindows(hdlg, [](HWND hwnd, LPARAM lParam)
  {
    auto hdlg = reinterpret_cast<HWND>(lParam);
    wchar_t className[10];
    if (GetClassName(hwnd, className, 10) == 4 &&
        CompareStringOrdinal(className, -1,
                             L"edit", -1, TRUE) == CSTR_EQUAL) {
      OnEditUpdate(hwnd);
    }
    return TRUE;
  }, reinterpret_cast<LPARAM>(hdlg));
}
```

In fact, you can do even better still. As written, we're overloading the `EN_UPDATE` notification, which means that the `OnEditUpdate` message needs to decide what got updated, and detect that this was just a fake update for the purpose of updating the decimal separator. Why not just call that function directly?

```
void TriggerEditUpdateChildWindows(HWND hdlg)
{
  EnumChildWindows(hdlg, [](HWND hwnd, LPARAM lParam)
  {
    auto hdlg = reinterpret_cast<HWND>(lParam);
    wchar_t className[10];
    if (GetClassName(hwnd, className, 10) == 4 &&
        CompareStringOrdinal(className, -1,
                             L"edit", -1, TRUE) == CSTR_EQUAL) {
      UpdateDecimalSeparator(hwnd);
    }
    return TRUE;
  }, reinterpret_cast<LPARAM>(hdlg));
}
```

Wait, you can do even better still.

You already know which three edit controls you need to update: They're `IDC_VALUE1`, `IDC_VALUE3`, and `IDC_VALUE5`. So you can just update them.

```
void UpdateDecimalSeparators(HWND hdlg)
{
  static const int editsWithDecimals[] = {
    IDC_VALUE1, IDC_VALUE3, IDC_VALUE5 };
  for (int id : editsWithDecimals) {
    UpdateDecimalSeparator(GetDlgItem(hdlg, id));
  }
}
```

In fact, it's highly likely that you already obtained these child window handles and have stashed them away in member variables.

```
void UpdateDecimalSeparators(HWND hdlg)
{
  UpdateDecimalSeparator(m_hwndValue1);
  UpdateDecimalSeparator(m_hwndValue3);
  UpdateDecimalSeparator(m_hwndValue5);
}
```

There you go. No need to generate a fake notification (that may confuse other parts of your program). If you want to update the decimal separator in three edit controls, then just update the decimal separator in three edit controls.

Raymond Chen

Follow