

How can I tell whether a change to a control was due to the user or due to my program?

 devblogs.microsoft.com/oldnewthing/20250106-00

January 6, 2025



In many UI frameworks, controls raise an event when the state of the control changes. The Win32 edit control raises the `EN_CHANGED` notification, for example. This is sometimes desirable. For example, maybe you want the “Author” field to match the current Book ID.

```
void OnBookIdChanged()  
{  
    auto id = ReadBookId();  
    auto author = LookUpAuthor(id);  
    SetAuthorText(author);  
}
```

Even if the book ID is set programmatically, you still want the author to be updated.

But often, you want your program to respond to changes in a control if they were initiated by the user, but not if they were triggered by the program itself. For example, you want to update a table with the new author as the user types it, but you don’t want to trigger an update when the code initializes the author field. How can you tell whether a change notification was due to the user or due to your program?

The simple solution is to set a flag when you are changing the value from your program, and then check that flag in the notification to decide whether to respond to it.

```

bool m_authorSetProgrammatically = false;

void SetAuthorProgrammatically(PCWSTR author)
{
    m_authorSetProgrammatically = true;
    SetWindowText(m_authorText, author);
    m_authorSetProgrammatically = false;
}

void OnAuthorChanged()
{
    if (m_authorSetProgrammatically)
    {
        // Ignore programmatic setting
        return;
    }

    [[ update the author in the table ]]
}

```

This also means that if an external component programmatically changes the author, your code will treat it as if the user had changed it. This is a good thing, because that external component might be an assistive technology that is updating the edit box on behalf of the user.

Note that if there are multiple “author changed” handlers, and one of the other ones triggers additional changes that in turn change the author again (say, by applying an autocorrection to the new author), the second change will still be ignored because `m_authorSetProgrammatically` is still `true`. To avoid that, you could reset the flag on the first notification.

```

void OnAuthorChanged()
{
    if (std::exchange(m_authorSetProgrammatically, false))
    {
        // Ignore programmatic setting
        return;
    }

    [[ update the author in the table ]]
}

```