

Re: limiting edit box characters

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2004-12/1219.html>

From: Bill Thompson (billt61_at_rgv.rr.com)

Date: 12/24/04

Date: Fri, 24 Dec 2004 01:20:41 -0600

"Jayw710" <jayw710@aol.com> wrote in message
news:20041224005303.08274.00002469@mb-m05.aol.com...

> *Hi,*

>

> *Is there a way to limit what characters an edit box will accept? I could do*

> *keyboard intercept but that seems like overkill since there are a million*

> *controls on the page and they all have different characters they can accept. I*

> *looked for an event for edit box that tells when a new character is entered,*

> *but couldn't find that either. Is there a standard way to do this?*

>

> *E.g. I just want to limit my chars to the set to all digits 0-9 plus the period*

> *and comma. I want to add commas automatically if the user doesn't.*

>

> *Ideas?*

>

> *Thank you:)*

>

> *Jay*

>

Derive a class from CEdit. Add a WM_CHAR event handler. Use the derived class for your edit controls. Note: when you define a variable for a edit control from the dialog definition screen, create a Control type, NOT a String type. Then, replace the CEdit type definition for the controls with your cedit-derived class name.

You can use GetWindowText to get the string data out of your cedit-derived class object. This implies you don't need to use UpdateData, which is kind of a pain in the neck anyway.

You can create your new cedit-derived class using the class wizard, choose New Class..., and select CEdit as the base class.

As a further enhancement, you could add a member function to your Cedit-derived class to specify different character masking behaviors; call the member function from your dialog's constructor for each instance of the new edit controls. Alternatively, you could derive different classes with different behaviours.

If you find that the WM_CHAR event handler isn't sufficient to cover your needs, consider working with WM_KEYDOWN as well.

Here's an example:

```
// CLASS DEFINITION

class CMyEdit : public CEdit
{
// Construction
public:
    CMyEdit();

// Attributes
public:

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CMyEdit)
    //}AFX_VIRTUAL

// Implementation
public:
    virtual ~CMyEdit();

    // Generated message map functions
protected:
    //{AFX_MSG(CMyEdit)
    afx_msg void OnChar(UINT nChar, UINT nRepCnt, UINT nFlags);
    //}AFX_MSG

    DECLARE_MESSAGE_MAP()
};

// CLASS IMPLEMENTATION

CMyEdit::CMyEdit()
{
}

CMyEdit::~~CMyEdit()
{
}
```

```
}  
  
BEGIN_MESSAGE_MAP(CMyEdit, CEdit)  
    //{AFX_MSG_MAP(CMyEdit)  
    ON_WM_CHAR()  
    //}AFX_MSG_MAP  
END_MESSAGE_MAP()  
  
// note that we only pass numeric characters through to the cedit base class  
// effectively ignoring any character that is not '0', '1'...'9'  
  
void CMyEdit::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)  
{  
    if (nChar >= '0' && nChar <= '9')  
        CEdit::OnChar(nChar, nRepCnt, nFlags);  
}
```