



The following example shows how to handle edwin event primarily a text update. There are lot of times when we want to restrict what goes into the edwin such as only numbers, and nothing else.

Header file required:

```
#include <eikedwin.h>
```

Add following lines in .rssi file.

```
RESOURCE EDWIN r_your_edwin
{
    max_length
    default_text_editor_text_case;
    allowed_case_modes_editor_all_case_modes;
    numeric_keymap_editor_standard_number_mode_keymap;
    default_input_text_editor_text_input_mode;
    allowed_input_modes_editor_all_input_modes;
    special_characters_RAVKON_SPECIAL_CHARACTER_TABLE_DIALOG;
}
```

Add following lines in .h file.

```
class CYourContainer : public CCoeControl, public MEikEdwinObserver
{
public:
    void HandleEdwinEventL(CEikEdwin *aEdwin, TEdwinEvent aEventType);
};
```

Add following lines in .cpp file.

```
//In your container class

CEikEdwin* yourEdwin = new (ELeave)CEikEdwin;
yourEdwin->SetEdwinObserver(this); //to ensure you get callbacks when anything happens in the edwin

// Create a resource reader that we'll use to get the settings
TResourceReader reader;
iEikonEnv->CreateResourceReaderLC(reader, R_YOUR_EDWIN);
yourEdwin->ConstructFromResourceL(reader);
CleanupStack::PopAndDestroy(); // reader

// Implement the HandleEdwinEventL in your container class
void CYourContainer::HandleEdwinEventL(CEikEdwin *aEdwin, TEdwinEvent aEventType)
{
    if (aEventType == EEventTextUpdate)
    {
        <32>TBuf16;
        ->GetText(buf16);
        TBool needRedraw;
        for (TInt i = buf16.Length()-1; i > -1; i--)
        {
            if (!TChar(buf16[i]).IsDigit())
            {
                Delete(i, 1);      buf16.
                = ETrue;          needReDraw
            }
        }
    }
}
```

Handle_edwin_events

```
}
}
if (needReDraw )
{
    ->SetTextL(aEdwin, buf16);
    ->HandleTextChangedL();
    ->SetCursorPositionL(buf16.Length(), EFalse);
}
}
}
```

In case of a `CEikRichTextEditor`, calling the `SetTextL` directly would result in a `ETEXT 12` or a `EIKCOCTL 70` panic, to avoid that modify the code a little bit in the `HandleEdwinEventL` above, in case of a text change needed (`needReDraw` is `ETrue`) like this.

```
CEikRichTextEditor* editor = static_cast< CEikRichTextEditor* >(aEdwin);
CRichText* richText = editor->RichText();
richText->Reset();
richText->InsertL(0, buf16); // insert starting from 0 pos
//aEdwin->SetTextL(&buf16); // This causes the EText12 or EIKCOCTL 70 panics
aEdwin->HandleTextChangedL();
aEdwin->SetCursorPositionL(buf16.Length(), EFalse);
```

Now when the `HandleEdwinEvent` function gets called the `aEdwin` would point to your `Edwin` created earlier, and if there is a text inputted by the user, it would check if what's entered is not a digit, it would keep removing them. Thereby ensuring that you have only digits in the `edwin`.

Added by - Mayank on 12/05/2009