SFU CMPT-212 2008-1 Topic: MFC Controls

Ján Maňuch

E-mail: jmanuch@sfu.ca

Monday 31st March, 2008

MFC Controls

Note. *Controls are windows with specialized behavior. A number of controls are provided by the Windows API and wrapped inside MFC classes:* `CStatic`, `CDialog`, `CEdit`, `CButton`, `CMenu`, *etc..*

Note. *MFC provides a mechanism for containment of windows (including controls) into other windows.*

- `CStatic` (display text, in a resource file: `LTEXT`):
  Example 1: static.cpp
  Example 2: resize.cpp
- `CMenu` (create menu, in a resource file: `MENU`):
  Example: firstmenu.cpp, menu.h

Resource files

- to simplify the creation of controls, MFC provides us with resource files

```
1  IDR_MENU MENU
2  BEGIN
3      POPUP "File"
4      BEGIN
5          MENUITEM "Some Operation", IDM_FILESOMEOPERATION
6          MENUITEM SEPARATOR
7          MENUITEM "Exit",           IDM_FILEEXIT
8      END
9      POPUP "Help"
10     BEGIN
11         MENUITEM "Contents",       IDM_HELPCONTENTS
12         MENUITEM "About",          IDM_HELPABOUT
13     END
14 END
```

Example: menu.cpp, menu.h, menu.rc

MFC controls (continued)

- `CDialog` (resource: `DIALOG`) — a window for getting information from the user (usually contains other controls); usually created using resource files (`GetDlgItem()` used to access those controls in the program)
- `CEdit` (resource: `EDITTEXT`) — edit text control: allows user to edit 1 line; `GetWindowText()` and `SetWindowText()` used to exchange information with this control
- `CButton` (resource: `PUSHBUTTON` or `DEFPUSHBUTTON`) — generates a message when clicked

Example of Dialog: addition.cpp addition.h addition.rc (modified [Deitel])

More complex example with MainWindow and Dialog: dialog.cpp dialog.h dialog.rc dialog.ico

Notes:

- `GetDlgItemText()` is a shortcut for `GetDlgItem()` and `GetWindowText()`
- `SetDlgItemText()` is a shortcut for `GetDlgItem()` and `SetWindowText()`

Example with a multiline CEdit: edittext.cpp, edittext.h, edittext.rc (modified [Deitel])

Remark:

- use `ES_MULTILINE` in the resource file in style description

Boxes

- group boxes:
  - groups of *check boxes* (resource: `AUTOCHECKBOX`)
    Example: checkbox.cpp, checkbox.rc, checkboxids.h (modified [Deitel])
    - check boxes are `CButton`s; to create a check box directly in your program, you have to create `CButton` object and when calling `Create()` method on it, specify `BS_CHECKBOX` in the style parameter
    - to get state of check box, use `GetCheck()` method, to set a state, use `SetCheck(int)` method
    - group box (resource: `GROUPBOX`)is `CButton` with style `BS_GROUPBOX`

- groups of *radio buttons* (resource: `AUTORADIOBUTTON`)
  Example: radiobutton.cpp, radiobutton.rc, radiobuttonids.h (modified [Deitel])
  - radio button is again a `CButton` with style `BS_AUTORADIOBUTTON`
  - at most one radio button in the group can be selected; groups are specified by style `WS_GROUP` (starts a new group)
  - to find which radio button is selected, use `GetCheckedRadioButton(int,int)` — returns id of selected radio button or `0` if none selected

- list boxes (resource: `LISTBOX`):
  Example: listbox.cpp, listbox.rc, listboxids.h (modified [Deitel])
  - list box is `CListBox`
  - to add item to listbox, use `AddString(char*)` method (returns `int` — index of the element) or `InsertString(int,char*)`; to delete item, use `DeleteString(int)` method
  - style `LBS_SORT` cause items in a list box to be sorted
  - `GetCurSel()` method returns index of selected element, or `LB_ERR` if none is selected
  - `GetText(int,char*)` — gets a string from a list box with specified index, stores it in the second parameter
  - `ResetContent()` — eliminates all items

- combo boxes (resource: `COMBOBOX`)
  Example: combobox.cpp, combobox.rc, comboboxids.h (modified [Deitel])
  - combo box is `CComboBox` — works as a combination of list box and edit control
  - `AddString(char*)`, `InsertString(int,char*)` and `DeleteString(int)` methods work the same way as for `CListBox`
  - style `CBS_SORT` cause items in a combo box to be sorted
  - `GetCurSel()` method returns index of selected element, or `CB_ERR` if none is selected (user might have typed a new item which is not on the list)
  - `GetLBText(int,char*)` — gets a string from a combo box with specified index, stores it in the second parameter; `GetWindowText(char*,int)` gets the text in edit control part of combo box
  - `ResetContent()` — eliminates all items

Combined Example: boxes.h, boxes.rc, boxes.cpp, boxes-mainwindow.h, boxes-mainwindow.cpp, boxes-mydialog.h, boxes-mydialog.cpp

- `ON_LBN_SELCHANGE( id, handler )` — add a message handler to message map:
  - `id` is id of a list box
  - if user changes the selected item in this list box, Windows will send a message to our application
  - MFC is mapping this message to handler `handler`
- `GetCount()` — returns the number of items in the list/combo box
- `SetItemData(int index,void*)` — can associate a pointer to data of any type with an item in the list/combo box
- `GetItemData(int index)` — returns the associated pointer with the item

Sample questions

- What is a control in MFC? List 3 examples of MFC controls.
- What is a resource file? Why is it useful?