الفصل الخامس

استخدام أزرار التحكم

تعتبر أزرار التحكم **Button Controls** من أكثر أدوات التحكم استخداماً، ولا يكاد يخلو منها أى مربع حوارى. ويوجد منها ثلاثة أنواع، الأول هو زر الأمر **Button** (يطلق عليه أيضاً زر التحكم)، والثانى هو زر خانة الخيار **RadioButton**، والثالث هو زر مربع الاختيار **CheckBox**. ولكل زر من أزرار التحكم استخدام مميز عن الأنواع الأخرى. وفى هذا الفصل سنتناول الأنواع الثلاثة بشيءٍ من التفصيل.

بانتهاء هذا الفصل ستتعرف على:

- استخدام أزرار الأمر **Buttons**.
- استخدام أزرار الاختيار **Radio Buttons**.
- استخدام مربعات الاختيار **Check Boxes**.
- معالجة الأزرار المختلفة أثناء تنفيذ البرنامج.

obeikandl.com

استخدام أزرار الأمر Buttons

يعد زر الأمر من أبسط الأدوات المستخدمة فى برامج النوافذ، فكل زر يقوم بتمثيل أمر معين، وعند نقره يقوم باستدعاء الحدث الخاص بهذا الأمر. وتحتوى جميع المربعات الحوارية تقريباً على زرين على الأقل هما زر **Ok** وزر **Cancel** حيث يقوم المعالج تلقائياً بإنشاء هذين الزرين فى التطبيقات المبنية على المربعات الحوارية كما رأينا فى الفصول السابقة. رغم ذلك فإن الكثير من المربعات الحوارية يحتاج لإضافة أزرار أخرى للقيام بمهام معينة، كما تحتاج هذه الأزرار إلى تغيير خصائصها وأشكالها وكتابة الدوال التى تقوم بتنفيذها، وهذا ما سنتعرف عليه بالتفصيل فى هذا الجزء من الفصل.

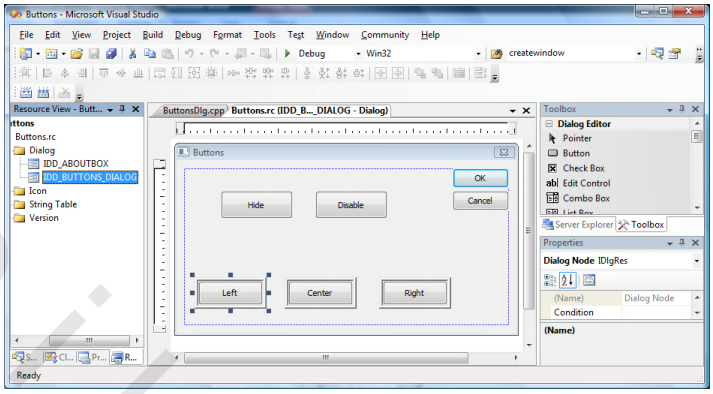
إضافة أزرار إلى المربع الحوارى

لمعرفة كيفية إضافة أزرار جديدة إلى المربع الحوارى، قم بإنشاء مشروع حوارى جديد كما تعلمت فى الفصول السابقة باسم **Buttons**، ثم تابع معنا الخطوات الآتية:

١. نشط التبويب **Resource View** ثم قم بتوسيع المجلد **Dialog** وانقر العنصر **IDC_BUTTONS_DIALOG** نقراً مزدوجاً، يظهر قالب المربع الحوارى داخل نافذة المحرر. انقر مربع النص الموجود بالمربع الحوارى ثم اضغط زر **Delete** من لوحة المفاتيح لحذف مربع النص نهائياً.

٢. اختر رمز زر الأمر Button من مربع الأدوات ثم قم بإضافة زرين بالركن الأيسر العلوى من المربع الحوارى (انظر شكل ٥-١).

٣. اختر الزر الأيسر ثم انقره بزر الفأرة الأيمن واختر **Properties** من القائمة الموضعية، يظهر مربع الخصائص الذى يعكس خصائص الزر الجديد فى هذه الحالة. انقر الزر الموجود بالركن الأيمن العلوى من مربع الخصائص إذا أردت أن يظل مربع الخصائص مفتوحاً مهما نقرت فى أى مكان آخر وحينئذٍ يتغير هذا الزر ليصبح هكذا .



شكل ٥-١ إضافة أزرار الأمر للمربع الحوارى

٤. من مربع الخصائص، قم بإدخال اسم الزر داخل خانة ID وليكن IDC_SHOW_HIDE ثم أدخل عنواناً للزر داخل خانةCaption وليكن Hide.

٥. انقر الزر الأيمن، وفى هذه الحالة سيعبر مربع الخصائص عن خصائص هذا الزر وليس الزر الأيسر. قم بإدخال اسم الزر داخل خانة ID وليكن IDC_ENABLE_DISABLE ثم أدخل عنواناً للزر داخل خانةCaption وليكن Disable.

٦. قم بإضافة ثلاثة أزرار أخرى بالجزء السفلى من المربع الحوارى (راجع شكل ٥-١).

٧. اختر الثلاثة أزرار الأخيرة وذلك بنقر كل زر على حده أثناء الضغط على زر Ctrl من لوحة المفاتيح.

٨. من مربع الخصائص، قم بتخصيص القيمة True للخاصيتين Client Edge وModal Frame لإضفاء أشكال مختلفة على الأزرار.

٩. قم بإدخال أسماء وعناوين الأزرار الثلاثة باختيار كل زر على حدة كما يلى:

	الزر الأيسر	*الزر الأوسط*	*الزر الأيمن*
الاسم	IDC_LEFT	IDC_CENTER	IDC_RIGHT
العنوان	Left	Center	Right

١٠. قم بإغلاق مربع الخصائص. يجب أن يظهر المربع الحوارى الآن مطابقاً للموجود بشكل ٥-١ السابق.

إضافة دوال حدث نقر الزر

عندما تقوم بنقر زر موجود داخل مربع حوارى، يقوم الزر بإرسال رسالة تسمى BN_CLICKED إلى المربع الحوارى.وحتى يستجيب البرنامج لهذه الرسالة بالحدث المناسب لابد من وجود دالة تحدد هذا الحدث والتى نطلق عليها دالة الاحتواء Handler Function أو دالة معالجة الحدث.

فى المثال الذى بين أيدينا فى هذا الفصل، يجب أن يحتوى كل زر من الأزرار الخمسة على دالة الاحتواء التى تحدد الحدث الذى سيقوم به الزر.وقبل أن نبدأ فى كتابة هذه الدوال، سنوضح بإيجاز المهام التى سيقوم بها كل زر من هذه الأزرار وذلك كما يلى:

- يقوم الزر IDC_SHOW_HIDE بالتبديل بين إظهار وإخفاء الأزرار الثلاثة الموجودة بالجزء السفلى من المربع الحوارى بالإضافة إلى تبديل عنوان الزر نفسه بين Hide وShow.
- يقوم الزر IDC_ENABLE_DISABLE بالتبديل بين تمكين أو تعطيل الأزرار الثلاثة الموجودة بالجزء السفلى من المربع الحوارى بالإضافة إلى تبديل عنوان الزر نفسه بين Enable وDisable. وحينما يكون الزر معطَّلاً Disabled، يكون عنوانه خافت اللون ولا يستجيب لنقر الفأرة أو ضغط أى مفتاح من لوحة المفاتيح.

لإضافة دالة حدث نقر الزر الأول Hide، من نافذة التصميم انقر زر Hide نقراً مزدوجاً، يظهر هيكل دالة الحدث باسم OnBnClickedShowHide() داخل نافذة تحرير الكود تمهيداً لإدخال الكود. وعلى هذا سيتم استدعاء الدالة OnBnClickedShowHide()

كلما قمت بنقر زر IDC_SHOW_HIDE، لكن ماذا سيحدث ؟ يعتمد ذلك على الكود الذى سيتم كتابته.

كتابة دوال أزرار التحكم

لتحديد ما يقوم به الزر IDC_SHOW_HIDE بمجرد نقره، قم بكتابة الكود التالى داخل الدالة OnBnClickedShowHide():

```
1.    void CButtonsDlg::OnBnClickedShowHide()
2.    {
3.      BOOL bVisible = GetDlgItem(IDC_LEFT)->
                                        IsWindowVisible();
4.      GetDlgItem(IDC_LEFT)->ShowWindow(bVisible ?
                                    SW_HIDE  : SW_SHOW);
5.      GetDlgItem(IDC_CENTER)->ShowWindow(bVisible ?
                                     SW_HIDE : SW_SHOW);
6.      GetDlgItem(IDC_RIGHT)->ShowWindow(bVisible ?
                                     SW_HIDE : SW_SHOW);
7.      GetDlgItem(IDC_SHOW_HIDE)->SetWindowText(bVisible
                                     ? "Show" : "Hide");
8.    }
```

وعن هذا الكود، نوضح ما يلى:

- فى السطر رقم ٣ قمنا بتمرير اسم الزر IDC_LEFT إلى الدالة GetDlgItem() التى تقوم بإرجاع مؤشر Pointer إلى تصنيف CWnd الخاص بالزر. يُستخدم هذا المؤشر فى استدعاء الدالة IsWindowVisible() (حيث يعامل الزر معاملة النافذة) وهى دالة عضو فى التصنيف CWnd تقوم بإرجاع القيمة SW_HIDE إذا كان الزر مرئى أو القيمة SW_SHOW إذا كان الزر مخفى، كما يتم تخزين هذه القيمة داخل المتغير المنطقى bvisible. وقد اكتفينا باختبار زر واحد، لأن الأزرار الثلاثة ستكون جميعا مرئية أو مخفية فى نفس الوقت.
- فى السطور ٤و٥و٦ قمنا بتمرير أسماء الأزرار بالترتيب إلى الدالة GetDlgItem() التى تقوم بإرجاع مؤشر إلى تصنيف CWnd المصاحب لكل زر، ومن ثمَّ يتم استخدام هذا المؤشر فى استدعاء الدالة العضو ShowWindow() التى تقوم

obeikandl.com

بإرجاع القيمة SW_HIDE التى تتسبب فى إخفاء الأزرار أو القيمة SW_SHOW التى تتسبب فى إظهار الأزرار، وهذا يعتمد بالطبع على حالة الأزرار المتمثلة فى المتغير bVisible الذى تم تمريره إلى الدالة ()ShowWindow وبالتالى سيتم إخفاء الأزرار الثلاثة مرةً واحدةً بالنقر على زر Hide ثم إظهارها بالنقر على نفس الزر مرةً أخرى.

- فى السطر رقم ٧، قمنا بتمرير اسم الزر IDC_SHOW_HIDE إلى الدالة ()GetDlgItem التى تقوم بإرجاع مؤشر إلى تصنيف CWnd الخاص بالزر، ومن ثمَّ يتم استخدام هذا المؤشر فى استدعاء الدالة العضو ()SetWindowText التى تقوم بتغيير العنوان من Hide إلى Show تبعاً لقيمة المتغير bvisible. فإذا كانت قيمته True يكون عنوان الزر هو Hide وإلا يكون عنوان الزر Show.
- بالنظر إلى الدالة الموجودة بالسطر ٧ ومن أول وهلة يظهر لك أن عنوان الزر سيكون Show عندما تكون الأزرار مرئية ويكون Hide حينما تكون الأزرار مخفية، إلا أننى أطلب منك إعادة التدقيق فى الدالة الموجودة بالسطر رقم ٤ حيث إذا كانت الأزرار مرئية، فسيتم إخفاؤها فى السطور ٥ و٦ وبالتالى يجب أن يكون عنوان الزر فى السطر ٧ هو Show. وعلى ذلك فإن الدالة صحيحة ولا تحتاج إلى تعديل كما توقعت من أول وهلة.

لتجربة عمل الزر بعد إدخال كود الدالة، تابع معنا الخطوات الآتية:

١. قم ببناء البرنامج وتنفيذه كما تعلمت فى الفصل السابق.
٢. انقر زر Hide، تلاحظ اختفاء الأزرار Left و Right و Center ويتغير عنوان الزر الذى قمت بنقره من Hide إلى Show.
٣. انقر زر Show، تلاحظ ظهور الأزرار Left و Right و Center مرةً أخرى ويتغير عنوان الزر الذى قمت بنقره من Show إلى Hide.

لكتابة دالة حدث الزر الثانى Disable، انقر زر Disable من نافذة التصميم نقراً مزدوجاً، يظهر هيكل دالة الحدث باسم ()OnBnClickedEnableDisable داخل

نافذة تحرير الكود تمهيداً لإدخال الكود. وعلى هذا سيتم استدعاء الدالة OnBnClickedEnableDisable() كلما قمت بنقر زر IDC_ENABLE_DISABLE. قم بكتابة الكود التالى داخل الدالة الجديدة:

```
1.    void CButtonsDlg::OnBnClickedEnableDisable()
2.    {
3.        BOOL bState = GetDlgItem(IDC_LEFT)->
                                IsWindowEnabled();
4.        GetDlgItem(IDC_LEFT)->EnableWindow(!bState);
5.        GetDlgItem(IDC_CENTER)->EnableWindow(!bState);
6.        GetDlgItem(IDC_RIGHT)->EnableWindow(!bState);
7.        GetDlgItem(IDC_ENABLE_DISABLE)->
              SetWindowText(bState ? "Enable" : "Disable");
8.    }
```

وعن هذا الكود، نوضح ما يلى:

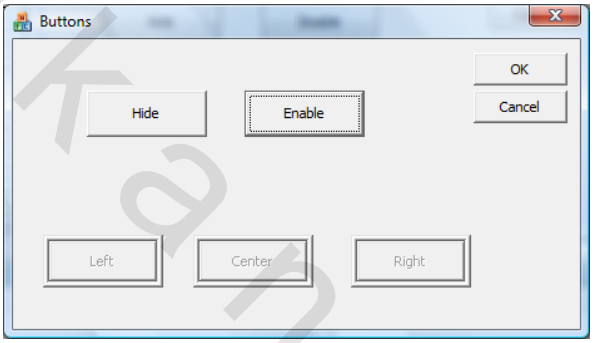
- فى السطر رقم ٣، قمنا بتمرير اسم الزر IDC_LEFT إلى الدالة GetDlgItem() التى تقوم بإرجاع مؤشر إلى تصنيف CWnd الخاص بالزر. يُستخدم هذا المؤشر فى استدعاء الدالة العضو IsWindowEnabled() التى تقوم بإرجاع القيمة TRUE إذا كان الزر ممكّناً أو القيمة FALSE إذا كان الزر معطَّلاً، كما يتم تخزين هذه القيمة داخل المتغير المنطقى bstate .وقد اكتفينا باختبار زر واحد لأن الأزرار الثلاثة ستكون جميعا ممكّنة أو معطَّلة فى نفس الوقت.
- فى السطور ٤و٥و٦ قمنا بتمرير أسماء الأزرار بالترتيب إلى الدالة GetDlgItem() حيث يقوم المؤشر باستدعاء الدالة العضوEnableWindow() بتمرير عكس حالة الأزرار إلى هذه الدالة وبالتالى تتسبب فى تمكين الأزرار الثلاثة فى حالة تعطيلها والعكس صحيح.
- فى السطر رقم ٧، قمنا بتمرير اسم الزر IDC_ENABLE_DISABLE إلى الدالة GetDlgItem() حيث يقوم المؤشر باستدعاء الدالة SetWindowText() التى تقوم بتغيير عنوان الزر من Disable إلى Enable أو العكس تبعاً لقيمة المتغير bState.

لتجربة عمل الزر **Disable** بعد إدخال كود الدالة، تابع معنا الخطوات الآتية:

١. قم ببناء البرنامج وتنفيذه.

٢. انقر زر **Disable**، تلاحظ تعطيل الأزرار **Left** و **Right** و **Center** ويتغير عنوان الزر الذى قمت بنقره من **Disable** إلى **Enable** (انظر شكل ٥-٢).

٣. انقر زر **Enable**، تلاحظ إعادة تمكين الأزرار **Left** و **Right** و **Center** مرةً أخرى ويتغير عنوان الزر الذى قمت بنقره من **Enable** إلى **Disable**.



شكل ٥-٢ تعطيل الأزرار الثلاثة بمجرد نقر زر **Disable**

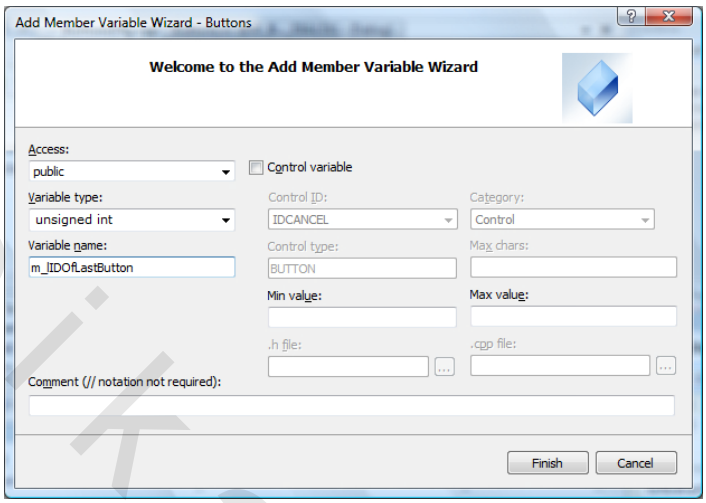
إضافة دوال الأزرار السفلية

قم بإضافة دوال أحداث الأزرار الثلاثة السفلية كما سبق حتى تحصل على الدوال **OnBnClickedLeft()** و**OnBnClickedRight()** و**OnBnClickedCenter()** للأزرار **Left** و **Right** و **Center** على الترتيب.

لكتابة كود الدوال الثلاثة، سنقوم بتعريف دالة جديدة لتغيير العنوان ومتغير يحتوى على آخر زر تم نقره وذلك باتباع الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.

٢. انقر التصنيف **CButtonsDlg** بزر الفأرة الأيمن ثم اختر **Add** ➢ **Add Variable** من القائمة الموضعية، يظهر معالج إضافة متغير جديد (انظر شكل ٥-٣).



شكل ٥-٣ إضافة متغير جديد للتصنيف

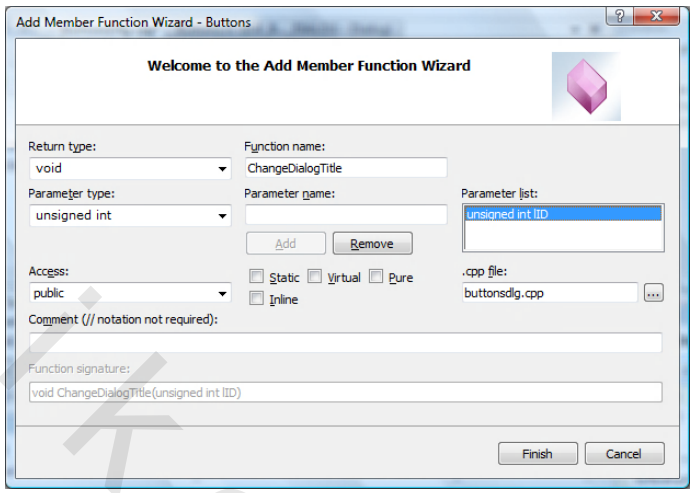
٣. اختر نوع المتغير من مربع السرد **Variable Type** وهو **unsigned int** فى هذه الحالة ثم اكتب اسم المتغير فى خانة **Variable name** وليكن **m_lIDOfLastButton** واختر درجة المتغير من مربع السرد **Access** وهو **Public** فى هذه الحالة.

٤. انقر زر **Finish** لإغلاق نافذة المعالج.

٥. انقر التصنيف **CButtonDlg** بزر الفأرة الأيمن مرةً أخرى ثم اختر **Add Add Function** من القائمة الموضعية، يظهر معالج إضافة دالة جديدة (انظر شكل ٥-٤).

٦. اختر نوع الدالة (أى نوع المتغير الذى ستعيده الدالة) من مربع السرد **Return Type** وهو **void** فى هذه الحالة وذلك لأن الدالة لن تعيد أية بيانات.

٧. اكتب اسم الدالة فى مربع النص **Function Name** وليكن **ChangeDialogTitle**.



شكل ٥-٤ إضافة دالة جديدة للتصنيف

٨. لإدخال معامل للدالة، قم باختيار نوع المعامل من مربع السرد **Parameter Type** وهو **unsigned int** ثم قم بتعيين اسم المعامل داخل مربع النص **Parameter Name** وليكن **IID** وأخيراً انقر زر **Add** لإضافة المعامل إلى قائمة معاملات الدالة بمربع السرد **Parameter List**.

٩. اختر مستوى وصول الدالة من مربع السرد **Access** وهو **Public** فى هذه الحالة.

١٠. انقر زر **Finish** لإغلاق نافذة المعالج وإضافة الدالة الجديدة إلى التصنيف.

١١. قم بإضافة كود الدوال الأربعة كما يلى (بالنقر المزدوج لكل زر ثم كتابة الكود التالى):

```
1.  void CButtonsDlg::OnBnClickedLeft()
2.  {
3.      ChangeDialogTitle(IDC_LEFT);
4.  }

5.  void CButtonsDlg::OnBnClickedCenter()
6.  {
7.      ChangeDialogTitle(IDC_CENTER);
8.  }

9.  void CButtonsDlg::OnBnClickedRight()
```

```
10.  {
11.      ChangeDialogTitle(IDC_RIGHT);
12.  }
13.  void CButtonsDlg::ChangeDialogTitle(unsigned int IID)
14.  {
15.      CString strCaption;
16.      GetDlgItem(IID)->GetWindowText(strCaption);
17.      strCaption += " was clicked";
18.      if(m_IIDOfLastButton == IID)
19.          strCaption += " again";
20.          SetWindowText(strCaption);
21.          m_IIDOfLastButton = IID;
22.  }
```

وعن هذا الكود، نوضح ما يلى:

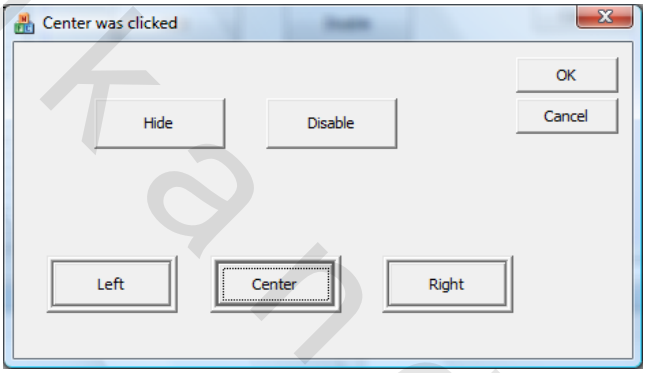
- فى السطور من ١إلى ١٢تقوم دوال الاحتواء **Handler Functions** باستدعاء الدالة **()ChangeDialogTitle** وذلك باستخدام اسم الزر كمعامل للدالة.
- تحتوى السطور من ١٣إلى ٢٢ على كود الدالة **()ChangeDialogTitle** حيث يتم استدعاء الدالة **()GetDlgItem** التى تقوم بإرجاع مؤشر إلى تصنيف **CWnd** الخاص بالأداة المناظرة، ومن ثمَّ يتم استخدام هذا المؤشر فى استدعاء الدالة **()GetWindowText** التى تقوم بتخزين عنوان الزر فى المتغير **strCaption**.
- فى السطر رقم ١٧استخدمنا المعامل =+ لإضافة العبارة **"was clicked"** إلى نهاية العنوان.
- فى السطر رقم ١٨تتم مقارنة اسم الزر الذى تم نقره بالمتغير العضو **m_IIDOfLastButton** الذى يحتوى على اسم آخر زر تم نقره.فإذا تساوى الاسمان ، يتم إضافة كلمة **again** إلى نهاية العنوان كما فى سطر ١٩.
- فى السطر٢٠، تم تمرير المتغير **strCaption** إلى الدالة **()SetWindowText** التى تقوم بتغيير عنوان المربع الحوارى.
- فى السطر ٢١، تم تخزين اسم الزر الذى تم نقره فى المتغير العضو **m_IIDOfLastButton**.

لتجربة عمل المربع الحوارى بعد التعديلات الأخيرة، تابع معنا الخطوات الآتية:

١. قم ببناء البرنامج وتنفيذه.

٢. انقر زر **Center**، تلاحظ تغير عنوان المربع الحوارى إلى **Center was clicked**.

٣. انقر زر **Center** مرةً أخرى، تلاحظ تغير عنوان المربع الحوارى إلى **Center was clicked again** (انظر شكل ٥-٥).



شكل ٥-٥ تأثير نقر أيٍ من الأزرار السفلية

استخدام أزرار الاختيار Radio Buttons

يوجد الكثير من المواقف التى يكون أمامك فيها خيار واحد فقط من عدة خيارات.فمثلاً عند إنشاء تطبيق قاعدة بيانات يحتوى على أسماء العملاء التى تتعامل معها الشركة، يوجد حقل من الحقول لتحديد نوع العميل هل هو رجل أم امرأة، وفى هذه الحالة لديك خيار واحد ولا يصح بأى حال من الأحوال أن تختار الخيارين فى نفس الوقت، وهذا هو الغرض الذى من أجله تم تصميم أزرار الاختيار **RadioButtons**.

سنقوم بإنشاء مشروع جديد كى نتعرف فيه على كيفية إضافة أزرار الاختيار إلى المربع الحوارى وكيفية تخصيص الكود اللازم لعملها. قم بإنشاء مشروع حوارى جديد باسم **Sports** كما تعلمت فى هذا الفصل والفصول السابقة.

إضافة أزرار الاختيار إلى المربع الحوارى

يتم غالبا وضع أزرار الاختيار داخل مربع مجموعة **Group Box**، كما يحتوى كل زر اختيار داخل المجموعة على خاصية تسمى **Autostyle** والتى يجب تنشيطها حتى يتم تعطيل زر الاختيار تلقائياً بمجرد تنشيط زر اختيار آخر داخل المجموعة، حيث يتم تنشيط هذه الخاصية تلقائياً بمجرد إضافة زر الاختيار إلى المربع الحوارى.

كما يحتوى كل زر اختيار على خاصية أخرى تسمى **Group**، والتى يتم تنشيطها مع زر الاختيار الأول فقط داخل المجموعة تمهيداً لإنشائها، لذا يجب تعطيل هذه الخاصية مع أزرار الاختيار الأخرى الموجودة فى نفس المجموعة.

فى المثال **Sports** الذى بين أيدينا، سنقوم بإضافة مجموعتين من أزرار الاختيار، الأولى يتم فيها اختيار النادى والثانية يتم فيها اختيار نوع اللعبة المفضَّلة. لأداء ذلك، تابع معنا الخطوات الآتية:

١. انقر المربع الحوارى **IDD_SPORTS_DIALOG** نقراً مزدوجاً، يتم فتح قالب المربع الحوارى داخل محرر الموارد. انقر مربع النص الموجود بالمربع الحوارى ثم اضغط مفتاح **Delete** من لوحة المفاتيح لحذف النص نهائياً.

٢. اختر رمز مربع المجموعة

من مربع أدوات التحكم ثم قم بإضافة مربع مجموعة جديد بالجانب الأيسر المربع الحوارى (انظر شكل ٥-٦).

٣. تأكد من اختيار مربع المجموعة الذى قمت بإضافته، ثم انقر المربع بزر الفأرة الأيمن واختر **Properties** من القائمة الموضعية.

٤. من مربع الخصائص، أدخل عنوان مربع المجموعة فى خانة **Caption** وليكن **Team** ثم قم بتخصيص القيمة **True** للخاصية **Tab Stop**.

٥. اختر رمز زر الاختيار

من مربع الأدوات ثم قم بإضافة أربعة أزرار داخل مربع المجموعة.

obeikandl.com

٦. قم بإدخال أسماء وعناوين أزرار الاختيار الأربعة باختيار كل زر على حدة مع تخصيص القيمة True للخاصية Group المصاحبة للزر الأول وطبقاً للبيانات التالية:

	الأول	الثانى	الثالث	الرابع
الاسم	IDC_ZAMALEK	IDC_AHLY	IDC_ISMAELY	IDC_MASRY
العنوان	Zamalek	Ahly	Ismaely	Masry

٧. اختر رمز مربع المجموعة من مربع الأدوات ثم قم بإضافة مربع جديد يمين مربع المجموعة Team (انظر شكل ٥-٦).

٨. أدخل عنوان مربع المجموعة فى خانة Caption وليكن Game ثم قم بتخصيص القيمة True للخاصية Tab stop.

٩. اختر رمز زر الاختيار من مربع الأدوات ثم قم بإضافة ثلاث أزرار داخل مربع المجموعة Game.

١٠. اختر أزرار الاختيار الثلاثة الموجودة بالمجموعة Game وذلك بنقر كل زر على حدة أثناء الضغط على زر Ctrl من لوحة المفاتيح.

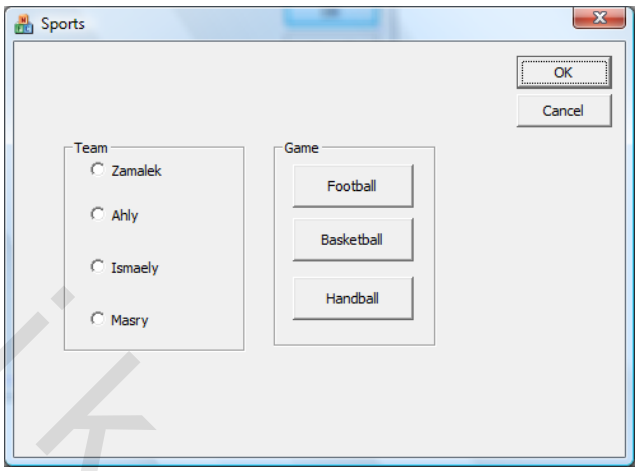
١١. من مربع الخصائص، قم بتخصيص القيمة True للخاصية Push-Like كى تظهر أزرار الاختيار كما لو كانت أزرار أمر.

١٢. قم بإدخال أسماء وعناوين أزرار الاختيار الثلاثة باختيار كل زر على حدة مع تخصيص القيمة True للخاصية Group المصاحبة للزر الأول وطبقاً للبيانات التالية:

	العلوية	الوسطى	السفلية
الاسم	IDC_FOOTBALL	IDC_BASKETBALL	IDC_HANDBALL
العنوان	Football	Basketball	Handball

يجب أن يظهر المربع الحوارى الآن كما فى شكل ٥-٦.

obeikandl.com

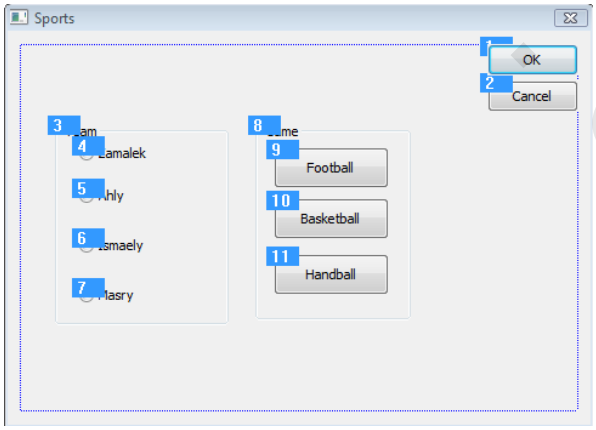


شكل ٥–٦ إضافة أزرار الاختيار للمربع الحوارى

للتأكد من ترتيب موارد المربع الحوارى ترتيباً صحيحاً حتى يسهل التحرك بينها باستخدام الأسهم أو مفتاح Tab، تابع معنا الخطوات التالية:

١. افتح قائمة Format من شريط القوائم ثم اختر Tab Order من القائمة المنسدلة (أو اضغط الاختصار Ctrl+D من لوحة المفاتيح)، يظهر المربع الحوارى Sports وبه ترتيب عناصر التحكم (الموارد) الموجودة بالمربع الحوارى (انظر شكل ٥–٧).

٢. قم بمقارنة الترتيب الموجود أمامك مع الترتيب الموجود بالشكل.



شكل ٥–٧ ترتيب جدولة المربع الحوارى

٣. لاختبار المربع الحوارى، اتبع أى من الطرق التالية:

- اضغط الاختصار **Ctrl+T** من لوحة المفاتيح
- افتح قائمة **Format** ثم اختر **Test Dialog** من القائمة المنسدلة
- انقر زر الموجود أقصى يسار شريط الأدوات **Dialog Editor** (لإظهار هذا الشريط، انقر أى شريط أدوات بزر الفأرة الأيمن ثم اختر **Dialog Editor** من القائمة الموضعية الناتجة)

٤. تأكد من أنك قادر على اختيار نادى واحد من الأندية ولعبة واحدة من الألعاب.

٥. اضغط مفتاح **Esc** من لوحة المفاتيح لإنهاء الاختبار.

التعرف على زر الاختيار النشط

الغرض من استخدام مربع مجموعة يحتوى على عدد من أزرار الاختيار هو اختيار عنصر واحد من عناصر متعددة. وأثناء البرنامج يتم اتخاذ قرار معين بناءً على العنصر المختار، وذلك بتعيين متغير يحتوى على العنصر المختار لكل مربع مجموعة، ويحتوى هذا المتغير افتراضيا على العنصر الأول فى المجموعة والذى تكون قيمة الخاصية **Group** فيه هى **True**. لتعيين متغير لكل مجموعة من المجموعتين، تابع معنا الخطوات الآتية:

١. نشط التبويب **Class View** من نافذة عمل المشروع.

٢. انقر التصنيف **CSportsDlg** بزر الفأرة الأيمن ثم اختر **Add** ⮞ **Add Variable** من القائمة الموضعية، يظهر معالج إضافة متغير جديد (انظر شكل ٥-٨).

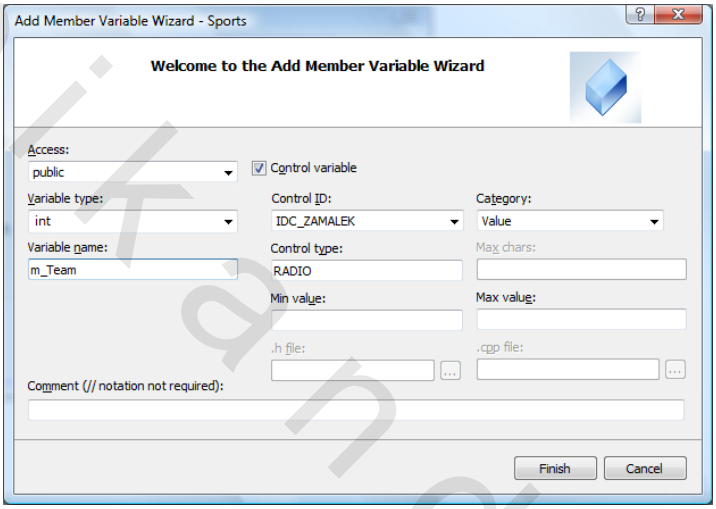
٣. قم بتنشيط مربع الاختيار **Control variable** لأننا نرغب فى إنشاء متغير مرتبط بإحدى أدوات التحكم الموجودة بالمربع الحوارى.

٤. اختر زر الاختيار الأول داخل مربع المجموعة **Team** من مربع السرد **Control ID** وهو **IDC_ZAMALEK** فى هذه الحالة.

obeikandi.com

٥. اختر **Value** من مربع السرد **Category** لأننا نريد قيمة أداة التحكم فقط ولا نحتاج إلى التعامل مع الخصائص الأخرى المصاحبة لهذه الأداة.

٦. قم بإدخال نوع المتغير داخل مربع السرد والتحرير **Variable Type** وهو **int** فى هذه الحالة حيث تظهر القيمة **bool** كقيمة افتراضية وحيدة داخل المربع.



شكل ٥-٨ إضافة متغير جديد للتصنيف

٧. اكتب اسم المتغير فى خانة **Variable name** وليكن **m_Team** واختر درجة المتغير من مربع السرد **Access** وهو **Public** فى هذه الحالة.

٨. انقر زر **Finish** لإغلاق نافذة المعالج.

٩. قم بتكرار الخطوات السابقة لإنشاء المتغير **m_Game** المرتبط بزر الاختيار **IDC_FOOTBALL** داخل مربع المجموعة **Game**.

١٠. يمكنك تعيين التعليق الذى يظهر بجوار عبارة الإعلان عن المتغير داخل ملف التعريف وذلك بإدخال نص التعليق داخل مربع السرد **Comment** (راجع شكل ٥-٨).

وبهذا نكون قد قمنا بتعريف متغيرين جديدين يحتويان على القيمة صفر كقيمة ابتدائية.

الخطوة الأخيرة هى إدخال الكود الذى يحدد العنصر المختار فى كل مجموعة أثناء سير البرنامج. من نافذة التصميم، انقر زر Ok نقراً مزدوجاً ثم قم بإضافة كود الدالة الافتراضية OnBnClickedOk() كما يلى:

```
1.    void CSportsDlg::OnBnClickedOk()
2.    {
3.       CString strMessage;
4.       CString strGame;
5.       CString strTeam;
6.       UpdateData();
7.       GetDlgItem(IDC_FOOTBALL + m_Game)->
                                    GetWindowText(strGame);
8.       GetDlgItem(IDC_ZAMALEK + m_Team)->
                                    GetWindowText(strTeam);
9.       strMessage = "My Favorite team is " + strTeam + " and
                              my favorite game is " + strGame;
10.      AfxMessageBox(strMessage);
11.      OnOK();
12.   }
```

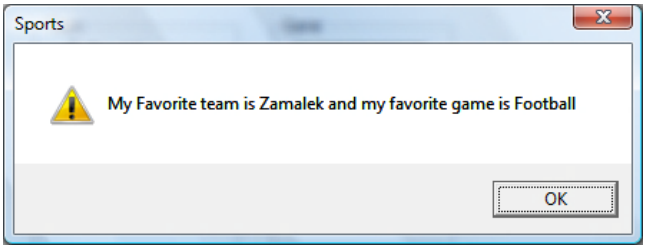
فى السطر رقم ٦، استخدمنا الدالة UpdateData() التى تقوم بنقل قيمة العنصر إلى المتغير المناظر، لذلك بعد السطر ٨ تصبح قيمة كل من m_Team وm_Game رقم تابع لأول عنصر بالمجموعة. فمثلاً إذا تم اختيار Zamalek، سيحتوى المتغير m_Team على القيمة صفر، وإذا تم اختيار Ahly ، يحتوى المتغير على القيمة ١ وهكذا. بعد ذلك يتم تخزين هذه القيم فى المتغيرين strGame وstrTeam كما فى السطرين ٧و٨.

لتجربة عمل المربع الحوارى، تابع الخطوات الآتية:

١. قم ببناء البرنامج وتنفيذه.

٢. اختر Zamalek مثلاً من المجموعة الأولى وFootball من المجموعة الثانية.

٣. انقر زر Ok، تظهر الرسالة الموضحة بالشكل ٥-٩.



شكل ٥-٩ الرسالة التى تظهر عند نقر زر Ok

استخدام مربعات الاختيار Check Boxes

يعتبر مربع الاختيار أحد أنواع أزرار التحكم وهو عبارة عن مربع صغير بجواره تعليق. وقد استخدمنا من قبل الكثير من مربعات الاختيار مثل مربع الاختيار **Control variable** بالمربع الحوارى **Add New Variable** (راجع شكل ٥-٨). يمكنك عند العمل مع مربعات الاختيار تعطيل هذه المربعات أو تنشيط أحد المربعات بنقره بزر الفأرة الأيسر كما يمكنك تنشيط أكثر من مربع اختيار. وعلى ذلك فهناك ثلاث حالات لمربع الاختيار، فقد يكون مربع الاختيار منشَّطاً ودلالة ذلك ظهور علامة ✓ داخل المربع، أو يكون معطَّلاً ودلالة ذلك عدم ظهور أى علامات داخل المربع، وقد يكون غير قادر على تحديد حالته حينما يتم اختيار أكثر من أداة مختلفة فى خاصية معينة ودلالة ذلك ظهور علامة ✓ باهتة داخل مربع الاختيار.

إضافة مربعات الاختيار إلى المربع الحوارى

سنقوم فيما يلى بإضافة مربعى اختيار إلى مشروع **Sports** لتحديد ما إذا كان الشخص يزور النادى أم لا وإذا ما كان يتناول العشاء فيه أم لا.

لإضافة مربعى الاختيار إلى المربع الحوارى، تابع معنا الخطوات الآتية:

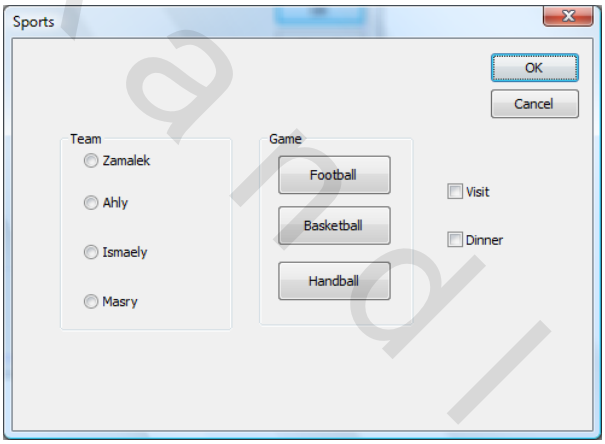
١. قم بفتح المربع الحوارى **IDD_SPORTS_DIALOG** داخل نافذة محرر الموارد.
٢. اختر رمز مربع الاختيار Check Box من مربع الأدوات ثم قم بإضافة مربعى الاختيار كما بالشكل ٥-١٠.
٣. اختر مربع الاختيار العلوى ثم انقره بزر الفأرة الأيمن، واختر **Properties** من

القائمة الموضعية لإظهار مربع الخصائص.

٤. قم بإدخال الاسم داخل الخاصية ID وليكن IDC_VISIT والعنوان فى الخاصية Caption وليكن Visit.

٥. نشط خاصية Group حتى يتم إنهاء مجموعة Game السابق إنشاؤها.

٦. اختر مربع الاختيار السفلى ثم قم بإدخال اسم مربع الاختيار داخل الخاصية ID وليكن IDC_DINNER ثم قم بإدخال عنوان لمربع الاختيار داخل الخاصية Caption وليكن Dinner. يجب أن يظهر الآن المربع الحوارى كما فى شكل ٥-١٠.



شكل ٥-١٠إضافة مربعى الاختيار إلى المربع الحوارى

إعداد مربعات الاختيار

كما ذكرنا فى أزرار الاختيار، أحياناً نريد معرفة حالة مربع الاختيار حتى يتم اتخاذ قرار معين بناءً على هذه الحالة. لمعرفة حالة مربع الاختيار، يتم استخدام التصنيف CButton والدوال الموجودة بالتصنيف مثل دالة GetCheck() التى تعكس حالة مربع الاختيار، والدالة SetCheck() المستخدمة لتنشيط أو تعطيل مربع الاختيار.

فيما يلى سنقوم بإدخال الكود الذى يحدد كيفية ربط مربعى اختيار ببعضهما البعض، حيث يمكن فى هذا المثال أن يدخل الشخص إلى النادى فقط أو أن يدخل النادى ويتناول فيه

وجبة العشاء، بينما لا يمكنه تناول وجبة العشاء فى النادى إذا لم يحضر إلى النـــادى مـــن الأساس. وهذا يعنى عدم القدرة على تنشيط مربع الاختيار **Dinner** فى حالة عدم تنشيط مربع الاختيار **Visit**. وكذلك عدم القدرة على تعطيل مربع الاختيار **Visit** إلا بعد تعطيل مربع الاختيار **Dinner**. قم بإنشاء دالتى احتواء لكلٍ من مربعى الاختيار بنقرهمـــا نقـــراً مزدوجاً واحدةً تلو الأخرى ثم قم بإدخال الكود التالى:

```
1.    void CSportsDlg::OnBnClickedVisit()
2.    {
3.      CButton* pVisit = (CButton*)GetDlgItem(IDC_VISIT);
4.      CButton* pDinner = (CButton*)GetDlgItem(IDC_DINNER);
5.      if (pDinner->GetCheck())
6.            pVisit->SetCheck (1);
7.    }

8.    void CSportsDlg::OnBnClickedDinner()
9.    {
10.     CButton* pVisit = (CButton*)GetDlgItem(IDC_VISIT);
11.     CButton* pDinner = (CButton*)GetDlgItem(IDC_DINNER);
12.     if (!pVisit->GetCheck())
13.           pDinner->SetCheck (0);
14.   }
```

وعن هذا الكود نوضح ما يلى:

- فى السطور ٣ و٤ و١٠ و١١ تم تحويل المؤشر الآتى من الدالة **()GetDlgItem** إلى مؤشر فى التصنيف **CButton** الذى يحتوى على دوال الوظائف المستخدمة مع جميع أنواع الأزرار. ولعلك تتساءل عن سبب عدم استخدام هذا التصنيف عنـــد العمل مع أزرار التحكم ؟؟ والإجابة ببساطة أن أزرار التحكم ليس لها حالة، لذلك استخدمنا مؤشر التصنيف **CWnd** مباشرة، أما فى حالة مربعات الاختيـــار فقـــد استخدمنا المؤشر لاستدعاء الدوال **()GetCheck** و**()Setcheck** الأعضـــاء فى التصنيف **CButton** للتعرف على حالة مربع الاختيار أو تعيينها على الترتيب.
- فى السطر رقم ٥ تم استعمال عبارة **If** التى تقوم باختبار مربع الاختيار **Dinner**. فإذا كان نشطاً، يتم تنشيط مربع الاختيار **Visit** باستخدام دالـــة **()SetCheck**

وتمرير القيمة ١ والتى تعنى تنشيط مربع الاختيار، وإلا لا يستجيب مربع الاختيار نهائياً.

- فى السطر رقم ١٢ تم استعمال عبارة **If** كى تؤدى عكس العملية السابقة، حيث تختبر مربع الاختيار **Visit**، فإذا لم يكن منشطاً، يتم تعطيل مربع الاختيار **Dinner**.

لتجربة عمل المربع الحوارى بعد إدخال التعديلات السابقة، تابع معنا الخطوات الآتية:

١. قم ببناء البرنامج وتنفيذه.

٢. قم بتنشيط وتعطيل مربعى الاختيار ولاحظ عدم تنشيط مربع الاختيار **Dinner** إلا إذا كان مربع الاختيار **Visit** نشطاً. وعدم تعطيل مربع الاختيار **Visit** إلا إذا كان مربع الاختيار **Dinner** معطَّلاً.

لإكمال المثال الذى بين أيدينا فى هذا الفصل، قم بإدخال الكود الموجود فى السطور من ١٠ إلى ١٧ إلى دالة حدث نقر الزر **Ok** وذلك كما يلى:

```
1.    void CSportsDlg::OnBnClickedOk()
2.    {
3.      CString strMessage;
4.      CString strGame;
5.      CString strTeam;
6.      UpdateData();
7.      GetDlgItem(IDC_FOOTBALL + m_Game)->
                                    GetWindowText(strGame);
8.      GetDlgItem(IDC_ZAMALEK + m_Team)->
                                    GetWindowText(strTeam);
9.      strMessage = "My Favorite Team is " + strTeam + " and
                              my favorite game is " + strGame;
10.     CButton* pVisit = (CButton*)GetDlgItem(IDC_VISIT);
11.     CButton* pDinner = (CButton*)GetDlgItem(IDC_DINNER);
12.     if (pVisit->GetCheck())
13.       {
14.           strMessage += " and I visit the club";
15.              if (pDinner ->GetCheck())
16.                 strMessage += " to have dinner";
17.       }
```

```
18.  AfxMessageBox(strMessage);
19.  OnOK();
20.  }
```

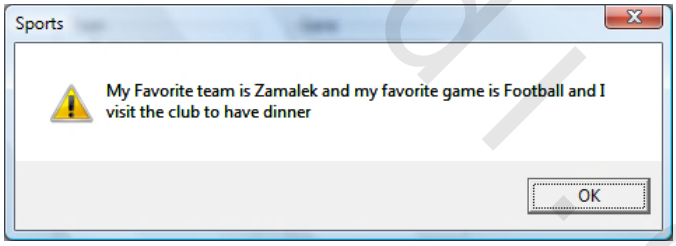
نلاحظ من الكود السابق للدالة **OnBnClickedOk()** أنها أخـــذت حـــالات مربعـــى الاختيار فى الحسبان عند إظهار الرسالة وذلك كما يلى:

- فى السطور ١٠ و١١ تم تحويل المؤشر الآتى من الدالة **GetDlgItem()** إلى مؤشر فى التصنيف بحيث يشير المؤشر **pVisit** إلى مربع الاختيار **IDC_VISIT** بينما يشير المؤشر **pDinner** إلى مربع الاختيار **IDC_DINNER**.
- فى السطور من ١٢ إلى ١٧ تم استخدام المؤشرات السابقة لاختبار حالة كل مربع اختيار على حدة.

لتجربة عمل المربع الحوارى بعد إدخال التعديلات، تابع الخطوات الآتية:

١. قم ببناء البرنامج وتنفيذه.

٢. نشط كلاً من مربعى الاختيار ثم انقر زر **Ok**، تظهر الرسالة وبها معلومات إضافية تعبر عن حالة مربعى الاختيار (انظر شكل ٥-١١).



شكل ٥-١١رسالة تعبر عن اختيارات المربع الحوارى