الفصل السادس عشر
العمل مع الصور والرسوم

للصور الرسومية والرموز فوائد كثيرة داخل تطبيقاتك، فقد تعبِّر صورة عما تعجز عنه مقالة كاملة. وهناك العديد من الأسباب التى تدعو المبرمجين إلى استخدام الصور والرسوم داخل تطبيقاتهم، إلا أن السبب الرئيسى هو مساعدة المستخدم فى التعرف على برنامج معين أو وظيفة معينة بسهولة تامة بدلاً من الإبحار فى نـــصٍ طويـــل يحتوى على غابة من الكلمات.

بانتهاء هذا الفصل ستتعرف على:

- *إنشاء وإحضار وتحرير الرموز والصور النقطية.*
- *عرض الصور والرسوم داخل المربعات الحوارية.*
- *استخدام الصور مع أزرار التحكم.*
- *استخدام الصور مع أدوات التحكم الأخرى.*

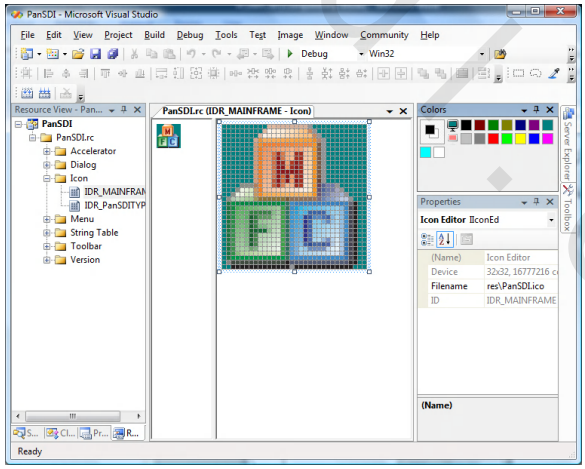
obeikandl.com

تحتوى تطبيقات Windows على عدة أنواع من الصور الرسومية، لكن أشهرها على الإطلاق ما يلى:

- صور الرمز Image Icon وتستخدم كرمز للتطبيق نفسه وبالتالى تظهر داخل مستكشف Windows أو كاختصار على سطح المكتب.
- الصورة النقطية Bitmap Image وتستخدم لتزيين الشاشات وأزرار أشرطة الأدوات، كما يمكن وضعها فوق المربعات الحوارية والنوافذ.
- صورة المؤشر Cursor Image وتستخدم لتغيير شكل مؤشر الفأرة.

استخدام محرر الصور

يمكنك باستخدام محرر الصور إنشاء وتطوير الأنواع المختلفة من الصور والرسوم وذلك باستخدام العديد من الخيارات المتاحة بقائمة Image والتى منها يمكنك عكس ألوان الجزء المختار من الصورة أو تدوير الصورة أو حفظها أو حتى إنشاء صورة أخرى باسم جديد. يوجد أيضاً شريطان للأدوات، الأول شريط أدوات الصور Image Editor والآخر شريط الألوان Colors (انظر شكل ١٦-١).



شكل ١٦-١ تحرير الرمز داخل محرر الموارد

إذا لم يظهر شريط أدوات الصور أمامك على الشاشة، تابع معنا الخطوات الآتية:

١. افتح قائمة **Tools** من شريط القوائم ثم اختر **Customize** من القائمة المنسدلة، يظهر المربع الحوارى **Customize**.
٢. نشط التبويب **Toolbars** إذا لم يكن هو التبويب النشط.
٣. من قائمة أشرطة الأدوات **Toolbars**، نشَّط مربع الاختيار **Image Editor** ثم انقر زر **Close** لإغلاق المربع الحوارى **Customize**.

كبديل للخطوات السابقة، قم بتوجيه مؤشر الفأرة إلى أى مكان خالى فى شريط القوائم.انقر بزر الفأرة الأيمن ثم نشِّط الاختيار **Image Editor** من القائمة الموضعية.

أما إذا لم يظهر شريط الألوان، فافتح قائمة **Image** من شريط القوائم ثم اختر **Show Colors Window** من القائمة المنسدلة.

إنشاء وتحرير الرموز

على الرغم من صعوبة التفرقة بين الرموز (**Icons**) والصور النقطية **Bitmaps** بمجرد النظر، إلا أن هناك بعض السمات الأساسية التى تميز كلٍ منهما على الأخرى وذلك كما يلى:

- تتكون الصورة النقطية من مصفوفة من البيانات لتمثيل صورة ملونة من النقاط الضوئية، أما الرمز فيتكون من صورتين نقطيتين، الأولى خاصة بالألوان وتسمى **Color image bitmap** والأخرى تسمى **Mask bitmap**.
- للرموز أحجام بسيطة (غالباً **16X15** أو **32X32**) أما الصور النقطية فأحجامها كبيرة قد تصل إلى **2048X2048**.
- تحتوى الرموز على خاصية الصورة الشفافة **Transparent** وخاصية عكس الألوان **Inverted Color** أما الصور النقطية فلا تحتوى على أيٍ من الخاصتين.

تعديل الرمز الافتراضى للتطبيق

يقوم معالج التطبيقات **Application Wizard** بإضافة رمز افتراضى للتطبيقات

obeikandl.com

المنشأة من خلاله، يسمى هذا الرمز **IDR_MAINFRAME** والذى يمكنك مشاهدته داخل المربع الحوارى **About**. لإجراء تعديلات على الرمز الافتراضى للتطبيق، تابع معنا الخطوات الآتية:

١. نشط التبويب **Resource View** من نافذة عمل المشروع إذا لم يكن هو التبويب النشط.

٢. قم بتوسيع المجلد **Icon** ثم انقر اسم رمز التطبيق **IDR_MAINFRAME** نقراً مزدوجاً، تظهر صورة الرمز داخل نافذة محرر الموارد (راجع شكل ١٦-١).

٣. قم بإجراء التعديلات المطلوبة على الصورة الأساسية **Standard (32X32)** باستخدام شريطى **Colors** و **Image** كما لو كنت تعمل داخل برنامج **Paint** الموجود بنظام التشغيل **Windows**.

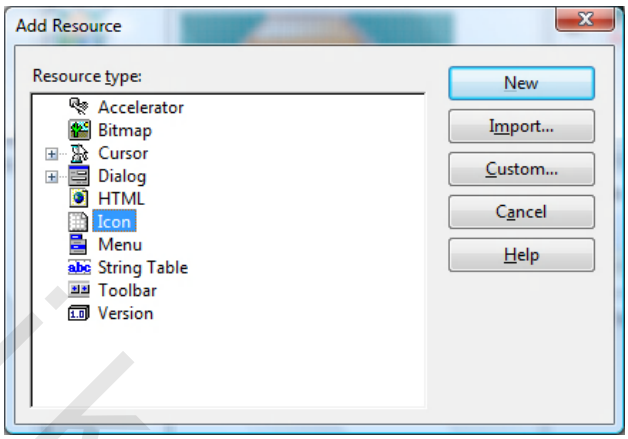
٤. اختر الصورة الأخرى للرمز **Small(16X16)** من مربع السرد والتحرير **Device** ثم قم بتعديلها إن أردت كما سبق.

إضافة رمز جديد للتطبيق

قد يحتوى التطبيق على عدة رموز بدلاً من رمز واحد وذلك بإنشاء رموز جديدة من البداية أو إحضار رموز موجودة مسبقاً منفردة أو داخل تطبيقات أخرى. لإنشاء رموز جديدة، تابع معنا الخطوات الآتية:

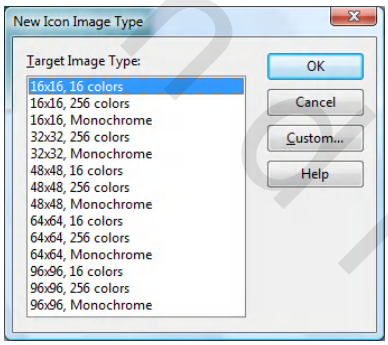
١. من نافذة التبويب **Resource View**، انقر أى مجلد بزر الفأرة الأيمن ثم اختر **Add>>Resource** من القائمة الموضعية، يظهر المربع الحوارى **Add Resource** (انظر شكل ١٦-٢).

٢. اختر **Icon** من قائمة **Resource Type** ثم انقر زر **New**، يتم إنشاء رمز فارغ داخل محرر الموارد.



شكل ١٦-٢ إضافة رموز جديدة

٣. انقر زر من شريط الأدوات **Image Editor**، يظهر المربع الحوارى **New Icon Image Type** (انظر شكل ١٦-٣).



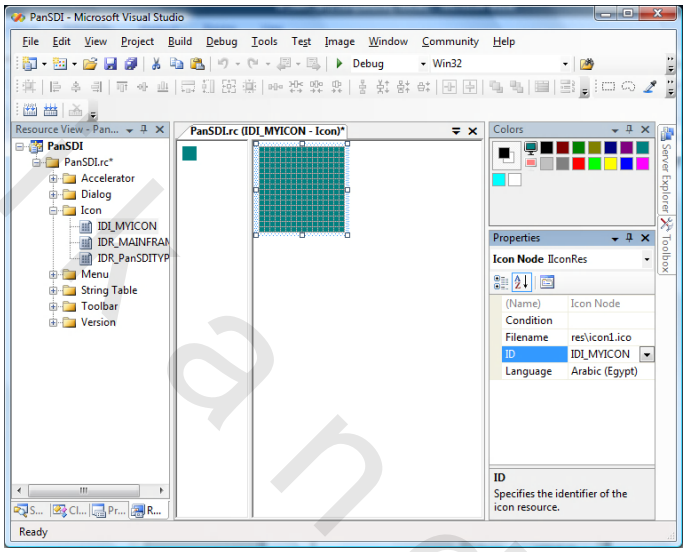
شكل ١٦-٣ اختيار حجم الرمز الجديد

٤. اختر الحجم المطلوب للصورة من قائمة **Target Image Type** أو انقر زر **Custom** لتعيين عرض وارتفاع وعدد ألوان الصورة يدوياً.

٥. انقر زر **Ok** لإغلاق المربع الحوارى والعودة مرةً أخرى إلى نافذة التحرير.

٦. من نافذة الخصائص، قم بإدخال اسم مناسب للرمز الجديد داخل الخاصية **ID** وليكن **IDI_MYICON**. لاحظ أن أسماء الرموز تصدر بالحروف **IDI_**.

٧. قم بإدخال اسم ملف الرمز داخل مربع **File name**. هذه الخطوة اختيارية حيث تقوم بيئة التطوير بإنشاء اسم الملف بنفس اسم الرمز تحت المجلد **res** المتفرع من المجلد الأساسى للمشروع (انظر شكل ١٦-٤).



شكل ١٦-٤ مربع خصائص الرمز

إذا أردت حذف رمز من رموز التطبيق، استخدم أى من الطريقتين الآتيتين:

- اختر اسم الرمز من التبويب **Resource View** ثم اضغط زر **Del** من لوحة المفاتيح، يتم حذف الرمز من التطبيق دون حذف ملف الرمز نفسه. فإذا أردت حذف ملف الرمز، قم بعمل ذلك يدوياً من مستكشف **Windows** أو جهاز الكمبيوتر.
- اختر الرمز داخل نافذة محرر الموارد ثم افتح قائمة **Image** واختر **Delete Image** من القائمة المنسدلة.

إضافة صورة نقطية للتطبيق

للصور النقطية العديد من الاستخدامات داخل التطبيقات. فهى تستخدم لإضفاء اللقطات الجمالية على النافذة التى تظهر على الشاشة، كما تستخدم أيضاً كواجهة لأزرار

التحكم كبديل جذاب عن النص المعتاد. قد يحتوى التطبيق على عدة صور نقطية وليس صورة واحدة وذلك بإنشاء صور نقطية جديدة من البداية أو إحضار صور موجودة مسبقاً منفردة أو داخل تطبيقات أخرى. لإنشاء صورة نقطية جديدة، تابع معنا الخطوات الآتية:

١. من نافذة التبويب **Resource View**، انقر أى مجلد بزر الفأرة الأيمن ثم اختر **Add>>Resource** من القائمة الموضعية، يظهر المربع الحوارى **Add Resource** (راجع شكل ١٦-٢).

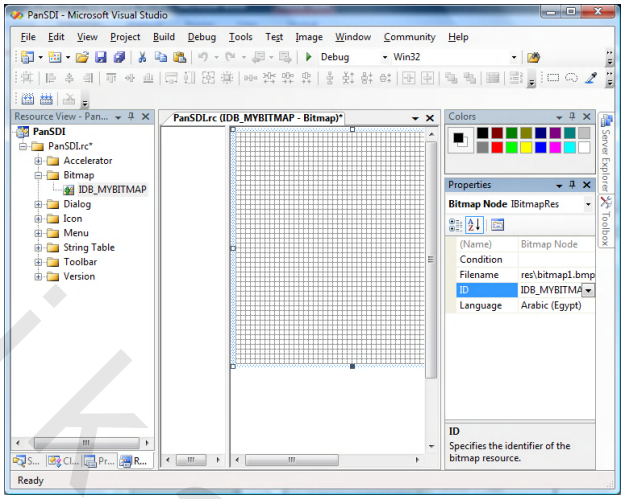
٢. اختر **Bitmap** من قائمة **Resource Type** ثم انقر زر **New**، يتم إنشاء صورة نقطية فارغة داخل محرر الموارد.

٣. تأكد من ظهور مربع الخصائص ثم قم بإدخال اسم الرمز داخل الخاصية **ID** وليكن **IDB_MYBITMAP**. لاحظ أن أسماء الرموز تصدر بالحروف **_IDB**.

٤. قم بتعيين حجم الصورة وعدد ألوانها داخل الخصائص **Width** و**Height** و**Colors**.

٥. قم بإدخال اسم ملف الصورة النقطية داخل الخاصية **File name**. وهذه الخطوة اختيارية كما ذكرنا حيث تقوم بيئة التطوير بإنشاء اسم الملف بنفس اسم الصورة النقطية تحت المجلد**res** المتفرع من المجلد الأساسى للمشروع (انظر شكل ١٦-٥).

obeikandl.com



شكل ١٦–٥ تعيين خصائص الصورة النقطية

ضبط حجم وألوان الصورة النقطية

تحتوى الصورة النقطية الجديدة افتراضياً على حجم **48X48** نقطة ضوئية كما تحتوى افتراضياً أيضاً على ١٦ لون. فإذا قمت بتقليل هذه الأبعاد، يتم حذف النقاط الضوئية من يمين وأسفل الشاشة. أما إذا قمت بزيادة الأبعاد، فيتم إضافة نقاط ضوئية يمين وأسفل الشاشة وتأخذ هذه النقاط نفس لون خلفية الصورة. لإعادة تعيين وضبط حجم وألوان الصورة النقطية، تابع معنا الخطوات الآتية:

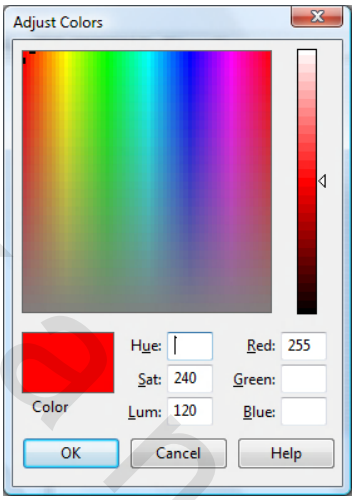
١. نشط التبويب **Resource View** من نافذة عمل المشروع إذا لم يكن هو التبويب النشط.

٢. قم بتوسيع المجلد **Bitmap** ثم انقر اسم الصورة النقطية نقراً مزدوجاً، تظهر الصورة داخل محرر الموارد.

٣. تأكد من ظهور مربع الخصائص ثم قم بإدخال أبعاد طول وعرض الصورة فى كلٍ من الخاصيتين **Height** و**Width**. كما يمكنك تعيين الأبعاد بنقر إحدى حواف الصورة والسحب فى الاتجاه المطلوب.

٤. اختر عدد الألوان المطلوب من مربع السرد والتحرير **Colors**.

٥. للتحكم فى الألوان الموجودة داخل مجموعة الألوان، انقر اللون الذى تريد تخصيصه نقراً مزدوجاً، يظهر المربع الحوارى **Adjust Colors** (انظر شكل ١٦-٦). اختر اللون المطلوب ثم انقر زر **Ok**.



شكل ١٦-٦ تخصيص الألوان

إحضار الصور إلى التطبيق

تعرضنا فيما سبق إلى كيفية إنشاء رمز أو صورة نقطية من البداية. هذا بالطبع يتطلب أن يكون لديك ميول فنية عالية، ولأن هذا غير متوفر دائماً، فنلجأ غالباً إلى إحضار الصور من ملفاتها الخارجية أو من تطبيقات أخرى لإضافتها إلى التطبيق الحالى. لإحضار صورة أو رمز من ملف، تابع معنا الخطوات الآتية:

١. انقر أى مجلد داخل التبويب **Resource View** بزر الفأرة الأيمن ثم اختر **Add>>Resource** من القائمة الموضعية، يظهر المربع الحوارى **Add Resource**.

٢. انقر زر **Import**، يظهر المربع الحوارى **Import** وهو يشبه إلى حد كبير المربع الحوارى **File Open**.

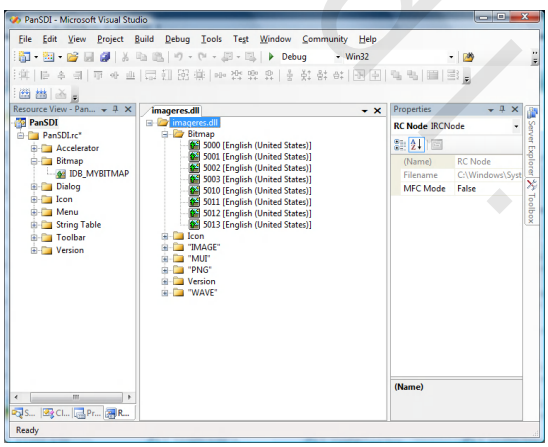
٣. اختر المجلد الذى يحتوى على الصورة (الصور) المراد إحضارها ثم اختـــر الصـــورة (الصور) المطلوبة وانقر زر **Open**، يتم إضافة الصورة إلى التطبيق إذا كان ملفها من الملفات المتعارف عليها كما يتم إعطاء اسم مميز للصورة ويوضع منها نســـخة داخل المجلد **res**.

إذا أردت إحضار صورة من ملف تنفيذى، تابع معنا الخطوات الآتية:

١. افتح قائمة **File** من شريط القوائم ثم اختر **File⮞Open** من القائمة المنســـدلة، يظهر المربع الحوارى **Open File** المعتاد.

٢. اختر **All Files** من مربع السرد والتحرير **Files of type**.

٣. اختر المجلد المناسب ثم اختر الملف التنفيذى الذى يحتوى على الصورة (الصـــور) المطلوبة حيث يكون الملف غالباً بأحد الامتدادات الآتيـــة **exe.** أو **dll.** أو **ocx.** وليكن ملف **Imageres.dll** الموجود أســـفل مجلـــد **System32** ثم انقـــر زر **Open**، تلاحظ ظهور موارد الملف التنفيذى داخل نافذة المحرر (انظر شكل ١٦-٧).

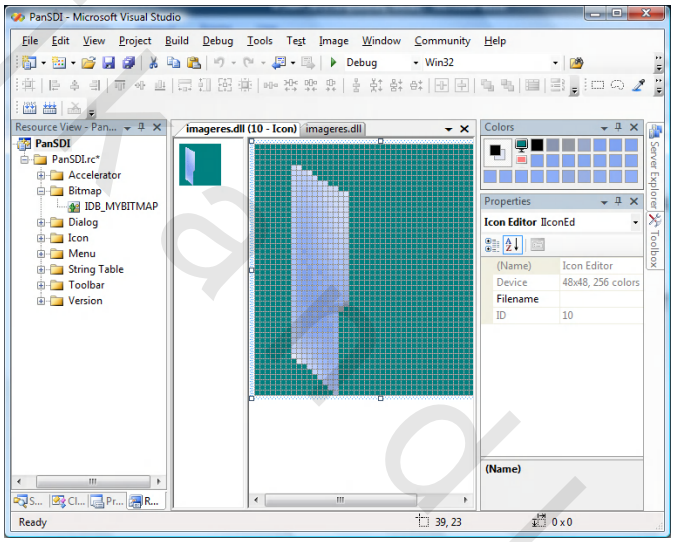


شكل ١٦-٧ إحضار الموارد من الملفات التنفيذية

٤. كى تتمكن من مشاهدة معاينة للصور، انقر أى من الصور نقراً مزدوجاً، تظهـــر الصورة داخل نافذة محرر الموارد (انظر شكل ١٦-٨).

٥. لإضافة صورة من الصور الموجودة بالقائمة إلى المشروع الحالى، قم بنقر الصورة المطلوبة بزر الفأرة الأيسر واستمر ضاغطاً ثم اضغط زر **Ctrl** من لوحة المفاتيح ووجه مؤشر الفأرة إلى التبويب **Resource View**، وعندما يتحول مؤشر الفأرة إلى علامة +، قم بتحرير زر الفأرة.

٦. افتح قائمة **File** من شريط القوائم ثم اختر **Close**.



شكل ١٦-٨ معاينة الصورة داخل نافذة محرر الموارد

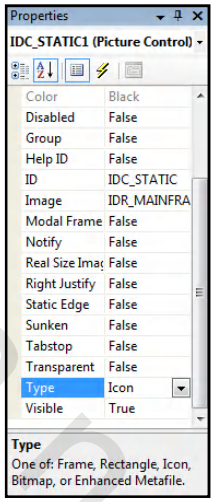
استخدام الصور داخل المربعات الحوارية

يوجد العديد من الطرق التى من خلالها يمكنك إظهار الصور داخل المربعات الحوارية، إلا أنه يفضل استخدام أداة الصور **Picture Control** لتمثيل كلٍ من الرموز والصور النقطية.

التحكم فى خصائص أداة الصور

كما ذكرنا منذ قليل، يقوم معالج التطبيقات **Application Wizard** تلقائياً بإضافة الرمز **IDR_MAINFRAME** لمشروعك. ولإظهار هذا الرمز، يقوم المعالج أيضاً بإضافة أداة

الصور بالمربع الحوارى **About**. يوضح شكل ١٦-٩ خصائص أداة الصور المستخدمة لاحتواء الرمز ومنه نلاحظ استخدام مربع السرد والتحرير **Type** لتحديد نوع الصورة، فإذا كانت صورة نقطية أو رمز، تم تعيين اسمها داخل مربع السرد والتحرير **Image**.



شكل ١٦-٩ خصائص أداة الصور المستخدمة لاحتواء الرموز

يوضح جدول ١٦-١ التالى أنواع الصور التى يمكن عرضها من خلال أداة الصور.

جدول ١٦-١ أنواع الصور التى يمكن عرضها داخل أداة الصور

النوع	*الاستخدام*
Frame	إظهار إطار أسود أو أبيض أو رمادى أو منقط، ويستخدم غالباً لتجميع عدة عناصر مع بعضها البعض
Rectangle	إظهار مستطيل مصمت أسود أو أبيض أو رمادى أو منقط
Icon	إظهار رمز
Bitmap	إظهار صورة نقطية
Enhanced Metafile	إظهار صورة محتواة داخل **Enhanced Metafile**

عرض الصور أثناء التشغيل

لتحميل رمز أو صورة نقطية أثناء التشغيل، يجب أن نقوم أولاً بتعيين متغير من نوع التصنيف **CStatic** لأداة الصور المستخدمة ثم بعد ذلك نستخدم إحدى الدوال الموضحة بجدول ١٦-٢ التالى تبعاً لنوع الصورة المستخدمة.

جدول ١٦-٢ الدوال المستخدمة لتحميل الصور أثناء التشغيل

الدالة	*الاستخدام*
SetIcon	تعريف الرمز الذى سيتم إظهاره
SetBitmap	تعريف الصورة النقطية التى سيتم إظهارها
SetCursor	تعريف المؤشر الذى سيتم إظهاره
SetEnhMetafile	تعريف صورة **Enhanced Metafile** التى سيتم إظهارها

للتعرف على كيفية استخدام الدوال السابقة لتحميل وعرض الصور أثناء التشغيل، استخدم معالج التطبيقات لإنشاء تطبيق حوارى جديد باسم **Images** ثم تابع معنا الخطوات التالية.

إضافة أداة الصور للمربع الحوارى

لإضافة أداة الصور للمربع الحوارى، تابع معنا الخطوات الآتية:

١. من نافذة تبويب عرض الموارد **Resource View**، قم بتوسيع المجلد **Dialog** ثم انقر اسم المربع الحوارى الذى سنقوم بإضافة أدوات الصور إليه **IDD_IMAGES_DIALOG** نقراً مزدوجاً، يظهر قالب المربع الحوارى داخل نافذة محرر الموارد.

٢. اختر مربع النص الافتراضى الموجود بالمربع الحوارى ثم اضغط مفتاح **Del** من لوحة المفاتيح لحذفه نهائياً لتوفير أقصى مساحة ممكنة.

٣. اختر رمز أداة الصور Picture Control من مربع الأدوات ثم انقر فى الـــركن الأيسر العلوى من المربع الحوارى، تلاحظ إضافة الأداة فى المكان الذى قمت بالنقر فيه.

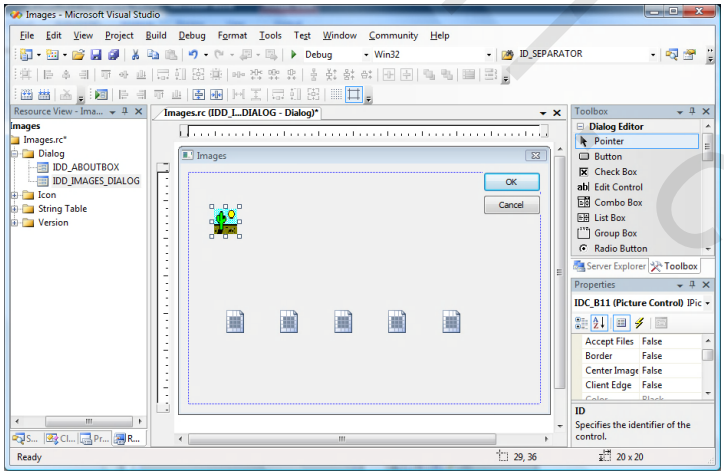
٤. من مربع الخصائص، قم بتغيير اسم الأداة من الخاصية ID لتصبح IDC_B1 مثلاً بدلاً من الاسم الافتراضى IDC_STATIC.

٥. اختر نوع الصورة التى ستقوم الأداة بعرضها من مربع السرد والتحريـــر Type وهو Bitmap فى هذه الحالة.

٦. بنفس الطريقة قم بإضافة خمسة أدوات صور إلى المربع الحوارى أســـفل الصـــورة الأولى (انظر شكل ١٦-١٠).

٧. اضغط مفتاح Ctrl من لوحة المفاتيح ثم انقر كل أداة من الأدوات الخمسة علـــى حده لاختيارها ثم قم باختيار نوع الصورة من مربع Type داخل مربع الخصائص ليصبح Icon (انظر شكل ١٦-١٠).

٨. قم باختيار كل أداة على حده ثم قم بإعطائها اســـم مناســـب ولـــيكن IDC_I1 وIDC_I2 وهكذا.



شكل ١٦-١٠ المربع الحوارى بعد إضافة الصور

ربط أداة الصور بمتغير

لربط أدوات الصور التى أضفناها بمتغيرات تعبر عن محتواها، تابع معنا الخطوات الآتية:

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

٢. انقر التصنيف **CImagesDlg** بزر الفأرة الأيمن ثم اختر **Add** ⮞ **Add Variable** من القائمة الموضعية، يظهر معالج إضافة متغير جديد.

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

٤. اختر أداة الصورة الأولى **IDC_B1** من مربع السرد **Control ID**.

٥. اختر **Control** من مربع السرد **Category** لأننا نحتاج إلى التعامل مع الخصائص الأخرى المصاحبة لهذه الأداة.

٦. تأكد من اختيار نوع البيانات **CStatic** بمربع السرد والتحرير **Variable Type**.

٧. اكتب اسم المتغير فى خانة **Variable name** وليكن **m_b1** واختر درجة المتغير من مربع السرد **Access** وهو **Public** فى هذه الحالة.

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

٩. قم بتكرار الخطوات السابقة للأدوات الأخرى من **IDC_I1** إلى **IDC_I5** وامنح كل متغير اسم فريد وليكن **m_i1** و **m_i2** وهكذا.

كتابة الكود اللازم لإظهار الصور أثناء التشغيل

لكتابة الكود اللازم لإظهار الصور أثناء التشغيل، تابع معنا الخطوات الآتية:

١. لأن أداة الصور الأولى ستحتوى على صورة نقطية، قم بإحضار صورة نقطية مع تغيير اسمها لاسم مميز وليكن **IDB_LOGO**.

٢. قم بإنشاء متغير عضو فى التصنيف **CImageDlg** من النوع **CBitmap** باسم **m_bmp**.

٣. قم بكتابة الكود التالى داخل الدالة **()OnInitDialog** بعد التعليقات **TODO**.

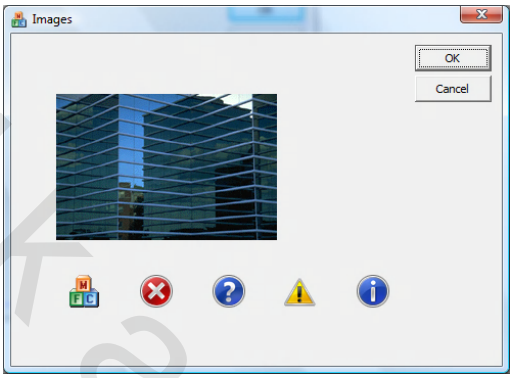
```
1.  // ** Load the Bitmap resource and
2.  // ** set the picture control to the loaded bitmap
3.  VERIFY(m_bmp.LoadBitmap(IDB_LOGO));
4.  m_b1.SetBitmap(m_bmp);

5.  CWinApp* pApp = AfxGetApp();
6.  HICON hIcon;

7.  // ** Load the application's icon and
8.  // ** set the picture control to the loaded bitmap
9.  hIcon = pApp->LoadIcon(IDR_MAINFRAME);
10. m_i1.SetIcon(hIcon);

11. // ** Load some of the standard icons and
12. // ** set each picture control
13. hIcon = pApp->LoadStandardIcon(IDI_HAND);
14. m_i2.SetIcon(hIcon);

15. hIcon = pApp->LoadStandardIcon(IDI_QUESTION);
16. m_i3.SetIcon(hIcon);
17. hIcon = pApp->LoadStandardIcon(IDI_EXCLAMATION);
18. m_i4.SetIcon(hIcon);

19. hIcon = pApp->LoadStandardIcon(IDI_ASTERISK);
20. m_i5.SetIcon(hIcon);
```

وعن هذا الكود نوضح ما يلى:

- فى السطر رقم ٣ استخدمنا الدالة **()LoadBitmap** لتحميل الصورة **IDB_LOGO** داخل المتغير **m_bmp**.
- فى السطر رقم ٤ استخدمنا الدالة **()SetBitmap** لعرض الصورة التى تم تخزينها داخل المتغير **m_bmp**.
- فى السطر رقم ٩ قمنا بتخزين رمز التطبيق داخل المتغير **hIcon** ثم عرضها داخل أداة الصور باستخدام الدالة **()SetIcon** فى السطر التالى.

- فى السطور ١٣ و ١٥ و ١٧ و ١٩ استخدمنا الدالة **LoadStandardIcon()** لتحميل بعض الرموز القياسية الموجودة داخل **Windows**.

قم ببناء البرنامج وتشغيله، تلاحظ ظهور المربع الحوارى كما فى الشكل ١٦-١١.



شكل ١٦-١١ عرض الصور أثناء التشغيل

استخدام الصور مع أزرار التحكم

غالباً ما تكون الصور والرسوم أكثر تعبيراً من الكلمات، لذلك ففى كثير من الأحيان يكون من الأفضل وضع صورة فوق الزر بدلاً من استخدام النصوص. ومن حسن الطالع، وجود التصنيف **CBitmapButton** داخل مكتبة **MFC** والذى صُمم خصيصاً لهذا الغرض، إلا أنه يجب تذكر شيئين هامين عند العمل مع الصور المصممة للوضع على أزرار التحكم، الأول أن يكون اسم الصورة عبارة عن سلسلة من الحروف والثانى تمكين الخاصية **Owner Draw** الموجودة بزر التحكم .

لتوضيح كيفية إنشاء واستخدام مثل هذه الأزرار، قم بإنشاء تطبيق حوارى جديد باسم **Smile** ثم أضف أداة زر إلى منتصف المربع الحوارى باسم **IDC_SMILE** وعنوان **Smile** ولا تنسى تخصيص القيمة **True** للخاصية **Owner Draw** بمربع الخصائص المصاحب للزر واحذف النص الافتراضى.فى هذا المثال سنقوم بإنشاء أربعة صور بحيث يتم استخدام صورة واحدة فقط تبعاً لحالة زر التحكم الموجود بالمربع الحوارى. لزر التحكم أربعة حالات أعلى **Up** وأسفل **Down** وممكّن **Enabled** أو **Focused** وغير ممكّن

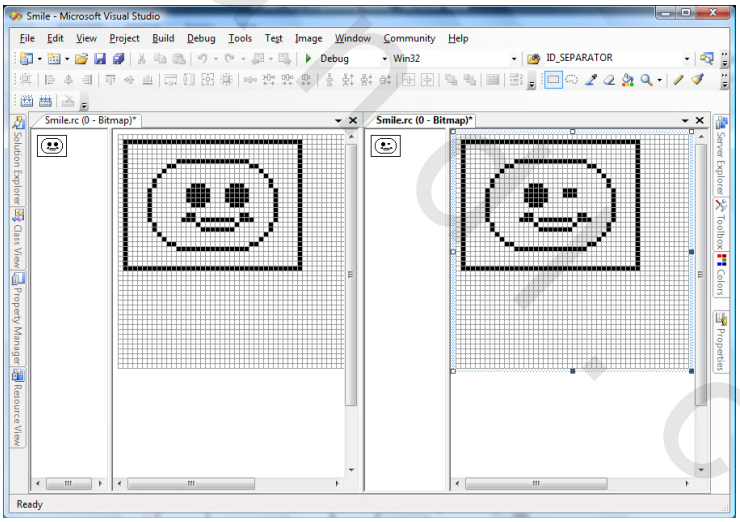
Disabled، تابع معنا الخطوات الآتية:

١. قم بإضافة صورة نقطية خالية إلى المشروع. لأداء ذلك، انقر المجلد الرئيسى بزر الفأرة الأيمن ثم اختر **Add>>Resource** من القائمة الموضعية واختر **Bitmap** بالمربع الحوارى الناتج **Add Resource** وأخيراً انقر زر **New**.

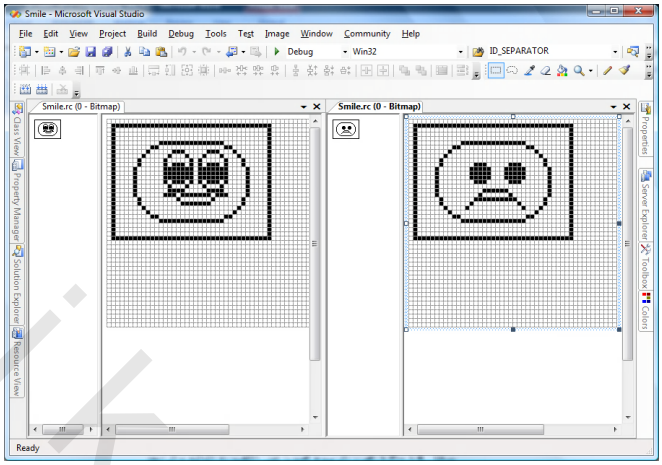
٢. تأكد من ظهور مربع الخصائص ثم قم بتغيير اسم الصورة الجديدة داخل الخاصية **ID** إلى اسم معبر وليكن **"SMILEU"** (بما فى ذلك الأقواس).

٣. قم بتحرير الصورة كما بالجانب الأيسر من شكل ١٦-١٢.

٤. كرر الخطوات من ١ إلى ٣ لإنشاء ثلاثة صور أخرى على أن تأخذ الأسماء **"SMILED"** و **"SMILEF"** أو **"SMILEX"** كما بالأشكال ١٦-١٢ و ١٦-١٣.



شكل ١٦-١٢ محرر الرسوم محتوياً على الصورة الأولى والثانية



شكل ١٦-١٣ محرر الرسوم محتوياً على الصورة الثالثة والرابعة

كما ذكرنا من قبل، تحتوى مكتبة MFC على التصنيف CBitmapButton المنبثق من التصنيف CButton والذى يستخدم لإجراء العمليات المختلفة عند وضع صورة نقطية على زر من أزرار التحكم. يحتوى هذا التصنيف على عدد قليل من الدوال، منها الدالة AutoLoad() المستخدمة لتحميل الصورة على الزر أثناء التشغيل.لتوضيح ذلك، قم بإضافة متغير عضو إلى التصنيف CSmileDlg من النوع CBitmapButton باسم مناسب وليكن m_bSmile ثم قم بإضافة السطر التالى للدالة OnInitDialog() بعد التعليقات TODO:

```
VERIFY(m_bSmile.AutoLoad(IDC_SMILE,this));
```

حيث تم تمرير اسم الزر للدالة AutoLoad() والذى تحصل منه على عنوان الزر الذى على أساسه يتم إظهار الصورة المناسبة (لذا قلنا أن اسم الصورة يجب أن يكون سلسلة من الحروف تنتهى بأحد الحروف U ، D ، X و F).

قبل أن تقوم ببناء التطبيق وتشغيله، نريد طريقة لوضع الزر فى حالة عدم التمكين. أبسط الحلول هو استخدام زر Cancel لأداء هذه المهمة. تابع معنا الخطوات الآتية:

١. قم بتغيير عنوان الزر Cancel إلى Dis/Enable.
٢. قم بتغيير اسم الزر إلى ID_DISENABLE.

٣. قم بإضافة دالة احتواء لنقر الزر باسم **OnBnClickedDisenable()**.

٤. قم بإضافة السطر التالى للدالة **OnBnClickedDisenable()** بعد التعليقات **TODO**:

```
m_bSmile.EnableWindow(!m_bSmile.IsWindowEnabled ());
```

والآن قم ببناء البرنامج وتنفيذه واضغط مفتاح **Tab** من لوحة المفاتيح ولاحظ تغير صورة الزر كلما فقد التركيز والعكس. حاول تمكين أو عدم تمكين الزر بنقر الزر **Dis/Enable**. حاول أيضاً نقر الزر، تلاحظ تغيير صورة الزر (انظرشكل ١٦-١٤).

شكل ١٦-١٤ تغيير الصورة بنقر الزر

استخدام الصور مع أدوات التحكم الأخرى

كما استخدمنا الصور مع أزرار التحكم، يمكننا أيضاً استخدامها مع أدوات التحكم الأخرى التى تعتمد على السرد مثل أداة السرد، شجرة البيانات أو مربع السرد والتحرير بحيث يكون لكل عنصر داخل المجموعة نص مميز بالإضافة إلى صورتين إحداها للتعبير عن اختيار العنصر والأخرى للتعبير عن عدم اختياره. هذه الصور قد تكون رموز أو صور نقطية داخل قائمة من الصور **Image List** .

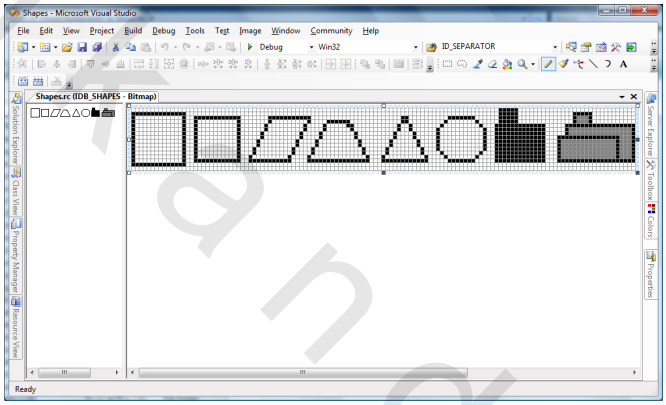
للتعرف على كيفية إنشاء قائمة الصور وكيفية استخدامها مع أداة التحكم، تابع معنا الخطوات الآتية:

obeikandl.com

١. استخدم معالج التطبيقات **Class Wizard** لإنشاء تطبيق حوارى جديد باسم **Shapes**.

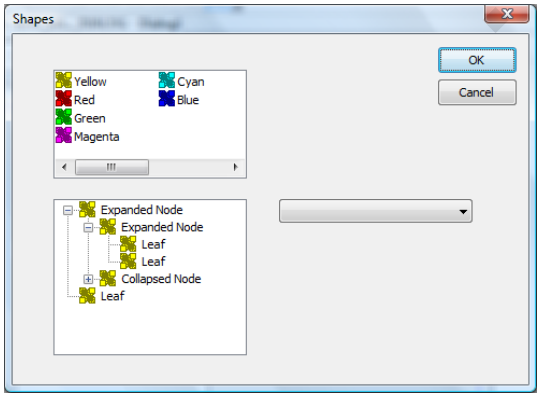
٢. لأننا نحتاج إلى مجموعة من الصور، فأبسط طريقة هى إنشاء صورة نقطية واحدة تحتوى على جميع الصور المطلوبة. قم بإضافة صورة نقطية للتطبيق باسم **IDB_SHAPES** بعرض ١٢٩وارتفاع ١٦.

٣. قم بتحرير الصورة النقطية كى تبدو كما فى شكل ١٦-١٥التالى.



شكل ١٦-١٥ الصورة النقطية بها مجموعة من الصور

٤. قم بإضافة أداة سرد وشجرة بيانات ومربع سرد وتحرير ممتد للمربع الحوارى **IDC_SHAPES_DIALOG** (انظر شكل ١٦-١٦).



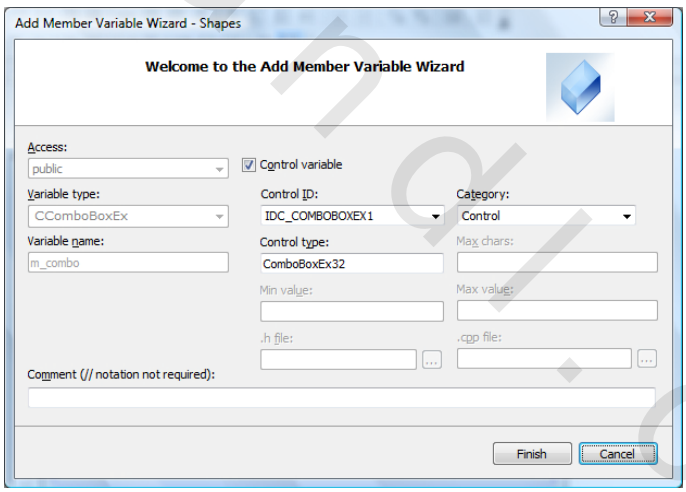
شكل ١٦-١٦تصميم المربع الحوارى **Shapes**

٥. من مربع الخصائص المصاحب لأداة السرد، اختر List من القائمة المصاحبة للخاصية View.

٦. من مربع الخصائص المصاحب لشجرة البيانات، قم بتخصيص القيمة True للخاصية Has Buttons وHas Lines وLines at Root.

٧. من مربع الخصائص المصاحب لمربع السرد والتحرير، اختر Drop List من القائمة المصاحبة للخاصية Type.

٨. قم بإضافة متغير للتصنيف CShapesDlg لكل أداة من الأدوات على أن تأخذ الأسماء m_list وm_tree وm_combo والأنواع CListCtrl وCListTree وCComboBoxEx على الترتيب، وفى كل مرة اختر Control من مربع السرد والتحرير Category (انظر شكل ١٦-١٧).



شكل ١٦-١٧إضافة المتغيرات إلى التصنيف CShapesDlg

إنشاء واستخدام قائمة الصور

تحتوى قائمة الصور على مجموعة من الصور النقطية التى سيتم استخدامها بواسطة أدوات التحكم الموجودة بالتطبيق. ولأن قائمة الصور مستقلة تماماً عن الأداة، فقد تستخدم نفس القائمة مع أكثر من أداة من أدوات التحكم. لإنشاء قائمة الصور لوضعها

داخل كائن من كائنات التصنيف CImageList، قم بإضافة متغير من النوع CImageList باسم m_imagelist إلى التصنيف CShapesDlg ثم قم بإضافة الكود التالى بعد السطر التعليقات TODO داخل الدالة OnInitDialog:

```
1.  static char* Shape[] = {
    "Square","Rectangle","Parallegelogram","Trapezoid","Trai
    ngle","Octagon"};
2.  int nShapes = 6;
3.  CBitmap bitmap;
4.  VERIFY(bitmap.LoadBitmap(IDB_SHAPES));
5.  m_imagelist.Create(IDB_SHAPES,16,1,0);
6.  m_imagelist.Add(&bitmap,(COLORREF)0xFFFFFF);
7.  m_list.SetImageList(&m_imagelist,LVSIL_SMALL);
8.  for(int i = 0;i<nShapes;i++)
9.      m_list.InsertItem(i,Shape[i],i);
10. m_tree.SetImageList(&m_imagelist,TVSIL_NORMAL);
11. HTREEITEM hTetragons,hOther;
12. hTetragons = m_tree.InsertItem("Tetragons",6,7);
13. hOther = m_tree.InsertItem("Other",6,7);
14. for(int i=0;i<nShapes;i++)
15.   m_tree.InsertItem(Shape[i],i,i,i<3 ? hTetragons : hOther);
16. m_combo.SetImageList(&m_imagelist);
17. COMBOBOXEXITEM cbItem;
18. cbItem.mask = CBEIF_TEXT
    |CBEIF_IMAGE|CBEIF_SELECTEDIMAGE;
19. for(int i=0;i<nShapes;i++)
20. {
21.   cbItem.pszText=Shape[i];
22.   cbItem.iItem = i;
23.   cbItem.iImage = cbItem.iSelectedImage = i;
24.   m_combo.InsertItem(&cbItem);
25. }
```

وعن هذا الكود، نوضح ما يلى:

- فى السطر رقم ١، قمنا بتعريف مصفوفة من الأسماء Shape. هذه الأسماء تتشابه مع الستة أشكال الأولى فى الصورة النقطية IDB_SHAPES.

- فى السطور ٣ و٤، قمنا باستخدام المتغير **bitmap** لتحميل الصورة النقطية **IDB_SHAPES**.
- فى السطر رقم ٥ تم استدعاء الدالة **()Create** والتى تحتوى على ثلاث معاملات، الأول هو اسم الصورة النقطية والثانى عبارة عن عرض النقطة فى كل صورة والثالث عبارة عن رقم لزيادة كفاءة الأداء، أما الرابع والأخير فيعبر عن اللون الذى سيتم استخدامه فى حالة الصور اللامعة.
- فى السطر رقم ٦ تم استدعاء الدالة **()Add** لإضافة محتويات الصورة النقطية للمتغير **m_imagelist**.
- فى السطر رقم ٧ تم استدعاء الدالة **()SetImageList** لملأ مربع السرد بالأشكال، حيث تحتوى الدالة على معاملين، الأول عنوان الصورة النقطية، والثانى لتحديد كيفية عرض الصور كما فى جدول ١٦-٣ التالى.

جدول ١٦-٣ طرق عرض الصور داخل مربع السرد

النوع	*المدلول*
LVSIL_NORMAL	تظهر قائمة الصور كرموز كبيرة
LVSIL_SMALL	تظهر قائمة الصور كرموز صغيرة
LVSIL_STATE	تظهر قائمة الصور حسب حالة كل صورة

- فى السطر رقم ٩ تم استدعاء الدالة **()InsertItem** داخل الدوارة **for** لإضافة العناصر الستة الموجودة بالمصفوفة **Shape** إلى مربع السرد، حيث تحتوى هذه الدالة على ثلاثة معاملات. الأول مكان العنصر داخل الأداة، والثانى نص العنصر نفسه، أما الثالث والأخير فيعبر عن مكان الصورة داخل قائمة الصور المعرفة بالدالة **()SetImageList**.

- فى السطر رقم ١٠ تم استدعاء الدالة **SetImageList()** لملأ شجرة البيانات بالأشكال، حيث تحتوى الدالة على معاملين، الأول عنوان الصورة النقطية، والثانى لتحديد كيفية عرض الصور داخل شجرة البيانات كما فى جدول ١٦-٤ التالى.

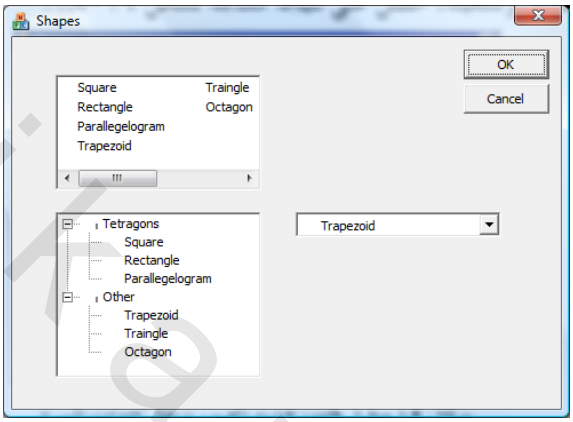
جدول ١٦-٤ طرق عرض الصور داخل شجرة البيانات

النوع	*المدلول*
TVSIL_NORMAL	تظهر قائمة بالصور المختارة وغير المختارة
TVSIL_STATE	تظهر قائمة الصور حسب حالة كل صورة

- فى السطور من ١١ إلى ١٣ يتم تعريف وإضافة العناصر الأبوية لشجرة البيانات وهى **Tetragons** أو **Other** وذلك باستخدام الدالة **InsertItem()** التى تحتوى فى هذه الحالة على ثلاثة معاملات، الأول هو نص العنصر والثانى يعبر عن مكان الصورة عند عدم اختيار العنصر (حيث يعبر الرقم ٦ فى هذه الحالة عن مجلد مغلق)، أما الثالث فيعبر عن مكان الصورة عند اختيار العنصر (حيث يعبر الرقم ٧ فى هذه الحالة عن مجلد مفتوح).
- فى السطور ١٤ و١٥ يتم إضافة العناصر الثلاثة الأولى الفرعية تحت العنصر الأبوى **Tetragons** والعناصر الثلاثة الأخيرة تحت العنصر الأبوى **Other** باستخدام الدالة **InsertItem()** التى تحتوى فى هذه الحالة على أربعة معاملات. الثلاثة معاملات الأولى كما سبق، أما المعامل الرابع فيحدد العنصر الأبوى للعنصر الفرعى المضاف، هل هو **Tetragons** أم **Other**.
- فى السطر رقم ١٦ يتم استدعاء الدالة **SetImageList()** مرةً أخرى لإضافة الصور لمتغير مربع السرد والتحرير **m_combo**.
- فى السطر رقم ١٧ يتم تعريف المتغير **cbItem** وهو من نوع التركيب **COMBOBOXEXITEM**. سيحتوى هذا المتغير عن بيانات العناصر التى سيتم إضافتها لمربع السرد والتحرير.

- فى السطور من ١٩ إلى ٢٥ يتم تحديد نص وصورة وترتيب العنصر المختار.
- فى السطر رقم ٢٤ يتم إضافة العنصر لمربع السرد والتحرير.

قم ببناء التطبيق وتشغيله، تحصل على نتيجة مشابهة لشكل ١٦-١٨.



شكل ١٦-١٨ ظهور قائمة الصور داخل أدوات السرد

obeikandl.com