الفصل الثامن عشر
تحزيم التطبيقات وإعدادها للتوزيع

بعد الانتهاء من تطوير تطبيقك، حان الوقت كى تقطف ثمار عملك بنشر هذا التطبيق إلى الآخرين. سنقوم فى هذا الفصل بالتعرف على كيفية إعداد تطبيقاتك للتوزيع بالإضافة إلى التعرف على كيفية تحديث المشروعات القديمة كى تعمل بتنسيق ***Visual Basic 2008***.

بانتهاء هذا الفصل، ستتعرف على:

- *إنشاء مشروع الإعداد.*
- *استخدام معالج مشروع الإعداد.*
- *تعيين خيارات مشروع الإعداد.*
- *توفيق نظام ملفات المستخدم.*
- *بناء مشروع الإعداد.*
- *نشر مشروع الإعداد.*
- *تحويل المشروعات القديمة.*

obeikandl.com

بمجرد الانتهاء من إنشاء تطبيقك واختباره من خلال التشغيل وإدخال البيانات المختلفة التى تتوقع أن يقوم مستخدم التطبيق بإدخالها، تأتى الخطوة الأخيرة والتى يتم فيها تجميع ملفات التطبيق أو المشروع مع بعضها البعض تمهيداً لنشرها وتوزيعها إلى المستخدمين. وسواءً كان هذا التطبيق خاص بك أو بمؤسستك أو شركتك أو حتى بعملائك، وسواءً ترغب فى بيع التطبيق بمبلغ من المال أو نشره مجاناً من خلال الإنترنت، فمن الضرورى فى كل هذه الحالات إعداد المثبت **Installer** الذى يقوم بالتأكد من تثبيت التطبيق على جهاز العميل أو المستخدم بصورة صحيحة.

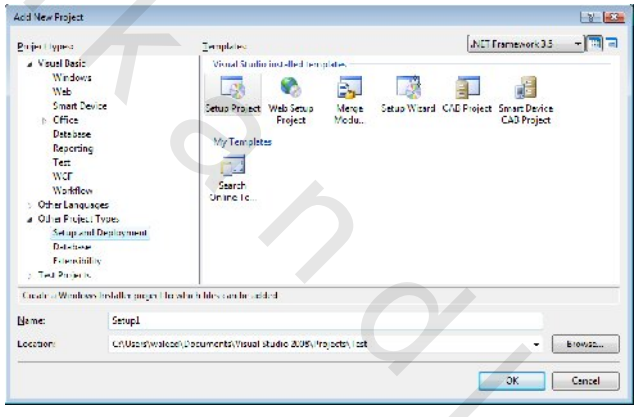
والطريقة المثلى لنشر التطبيقات النوافذية هى استخدام المثبت أو برنامج الإعداد **Installer** وهو عبارة عن نوع خاص من التطبيقات يستخدم مثبت **Microsoft Windows** المثبت بجميع إصدارات **Windows** بدءً من **Windows 2000**، حيث تعتمد عملية نشر تطبيقات **Visual Studio 2008** بشكل كبير على تقنية مثبت **Windows** وفيها يتم جمع ملفات التطبيق فى صورة حزمة برمجية يستطيع المستخدم من خلالها تثبيت التطبيق على حاسبه بنفس طريقة تثبيت البرامج الأخرى مثل **Windows** و**Microsoft Office**.

ويحتوى ملف الإعداد على الامتداد **MSI.** وهو اختصار **Microsoft Installer** والمعرف من خلال الإصدارات الأخيرة من **Windows** كحزمة تثبيت، ويتميز عن برامج الإعداد القديمة **Setup.exe** كما يتم من خلاله التأكد من تثبيت التطبيق لدى المستخدم وتسجيله بشكل صحيح. كما أن ملفات الإعداد لا تتضمن المزيد من الكود اللازم لعملية التثبيت، وذلك أن عملية التثبيت كما ذكرنا مبنية أساساً داخل **Windows**.

إنشاء مشروع الإعداد

أسهل الطرق المستخدمة فى إنشاء برنامج الإعداد أو التثبيت تتمثل فى إضافة مشروع الإعداد إلى الحل الذى يحتوى على تطبيقك وذلك باتباع الخطوات الآتية:

١. تأكد من فتح الحل الذى يحتوى على المشروع الذى ترغب فى نشره.
٢. افتح قائمة **File** ثم اختر **Add** و**New Project** من القوائم المنسدلة، يظهر المربع الحوارى **Add New Project** (انظر شكل ١٨-١).
٣. انقر المجلد **Other Project Types** من المربع **Project Types** بالجانب الأيسر ثم اختر **Setup and Deployment** واختر **Setup Project** من المربع **Templates** بالجانب الأيمن.
٤. قم بتعيين اسم مناسب لمشروع الإعداد ثم انقر زر **Ok**، يتم إضافة مشروع الإعداد إلى الحل المفتوح.



شكل ١٨-١ إضافة مشروع الإعداد إلى الحل المفتوح

ولتسهيل عملية الإعداد، يتم تضمين معالج لعملية الإعداد داخل **Visual Studio 2008**، حيث لا يقوم هذا المعالج بتضمين مشروع الإعداد داخل الحل المفتوح فقط وإنما يقوم أيضاً بملء الأجزاء الهامة من التطبيق بالبيانات اللازمة لإنشاء حزمة الإعداد القياسية من التطبيق. يمكنك بعد ذلك توفيق برنامج الإعداد تبعاً لحاجتك، إلا أن المعالج يقوم بالجزء الأكبر من هذه العملية.

سنقوم فيما يلى بالتعرف على كيفية استخدام معالج مشروع الإعداد فى إنشاء برامج إعداد مشروعات التطبيقات النوافذية ومشروعات تطبيقات الويب.

تحتوى حزمة مشروع الإعداد دائماً على إمكانية إلغاء تثبيت التطبيق. فبمجرد تثبيت التطبيق على الحاسب، يستطيع المستخدم فى أى وقت إلغاء هذا التثبيت من خلال نافذة Add/Remove Programs من لوحة التحكم Control Panel.

استخدام معالج مشروع الإعداد

يمكنك استخدام معالج مشروع الإعداد لإنشاء برنامج تثبيت تطبيقك من خلال أربع خطوات بسيطة للغاية. فإذا ما أردت إضافة المزيد من الوظائف إلى برنامج التثبيت، يمكنك أداء ذلك بنفسك بعد انتهاء عمل المعالج.

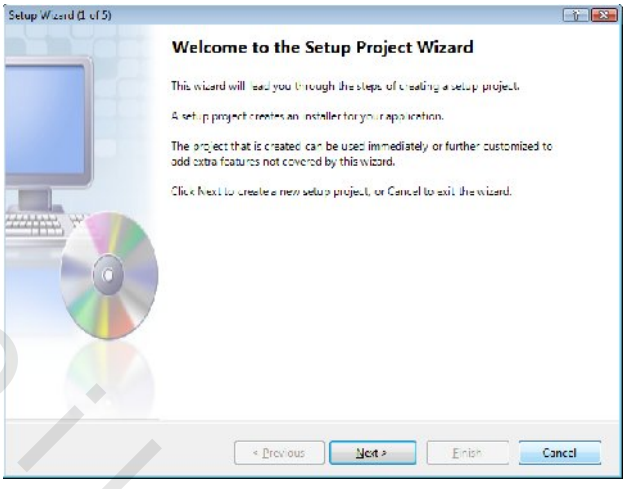
لتوضيح عملية إنشاء برنامج الإعداد لأحد التطبيقات النوافذية، قم بفتح أحد المشروعات الموجودة مسبقاً أو قم بإنشاء تطبيق جديد لاستخدامه فى عملية الإعداد. فالهدف هنا هو إنشاء حزمة تثبيت كى يتمكن المستخدم من تثبيت التطبيق على حاسبه حتى وإن لم يكن Visual Studio 2008 مثبتاً على حاسبه. تابع معنا الخطوات الآتية:

١. قم بفتح أحد المشروعات الموجودة من قبل داخل بيئة تطوير Visual Studio 2008.

٢. افتح قائمة File ثم اختر Add وNew Project من القوائم المنسدلة، يظهر المربع الحوارى Add New Project (راجع شكل ١٨-١).

٣. انقر المجلد Setup and Deployment من المربع Project Types ثم اختر Setup Wizard من المربع Templates.

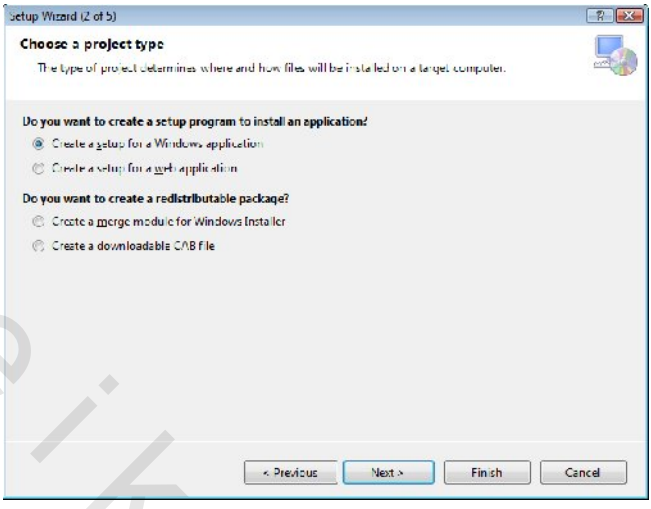
٤. قم بتعيين اسم مناسب لمشروع الإعداد ثم انقر زر Ok، تظهر النافذة الأولى من معالج مشروع الإعداد (انظر شكل ١٨-٢).



شكل ١٨-٢ النافذة الأولى من معالج مشروع الإعداد

٥. انقر زر **Next** للانتقال إلى الخطوة التالية من المعالج والتى يمكنك من خلالها اختيار نوع مشروع برنامج الإعداد الذى تقوم بإنشائه، حيث يوجد لديك أربعة خيارات هى (انظر شكل ١٨-٣):

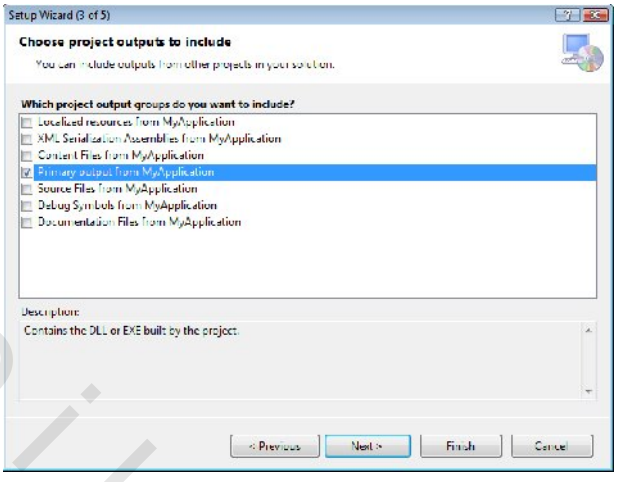
- إنشاء برنامج الإعداد لتطبيق نوافذى، ويستخدم فى حالة مشروعات التطبيقات النوافذية
- إنشاء برنامج إعداد لتطبيق ويب، ويستخدم فى حالة مشروعات تطبيقات الويب
- إنشاء نموذج مدمج لبرنامج الإعداد، ويستخدم إذا أردت إنشاء حزمة برمجية قابلة لإعادة التوزيع والدمج مع الحزم البرمجية الأخرى لإنشاء الحلول المخصصة التى يمكنك من خلالها تحديث المكونات الفردية كل على حدة بمعزلٍ عن المكونات الأخرى.
- إنشاء ملف **CAB** قابل للتحميل، وهو نفس الخيار السابق إلا أنه يكون فى صورة ملفات الضغط ذات الامتداد **cab**..



شكل ١٨-٣ تعيين نوع المشروع المستخدم

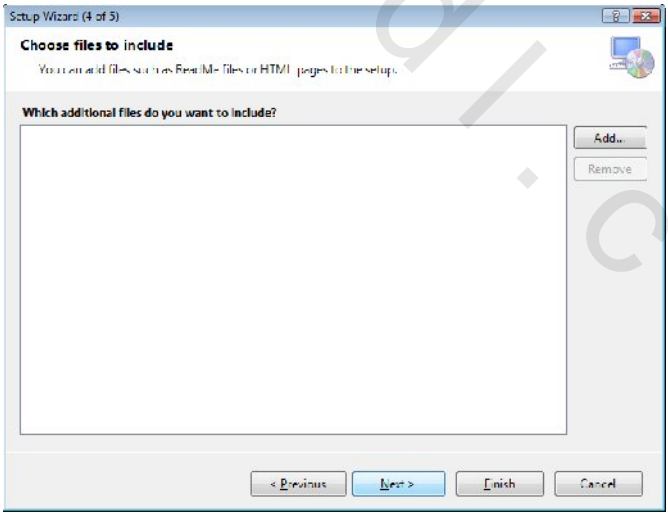
٦. لأننا نقوم بإنشاء مشروع إعداد لتطبيق نوافذى، انقر زر الاختيار الأول **Create a setup for a windows application** ثم انقر زر **Next**، تظهر النافذة التالية من المعالج (انظر شكل ١٨-٤).

٧. يمكنك من خلال هذه النافذة تعيين المكونات التى ترغب فى تضمينها داخل مشروع الإعداد، حيث يجب أن تختار على الأقل الخرج الأساسى لتطبيقك الممثل فى ملف التنفيذى **EXE** (أو **DLL**) الذى يتم إنشاؤه بمجرد بناء التطبيق. يمكنك أيضاً تضمين المزيد من المكونات مثل ملفات المصدر التى استخدمتها فى إنشاء التطبيق. اختر أى من مربعات الأخرى لترى شرحاً مختصراً لها بالجزء **Description** فى أسفل النافذة.



شكل ١٨–٤ تعيين المكونات التى ترغب فى تضمينها بحزمة الإعداد

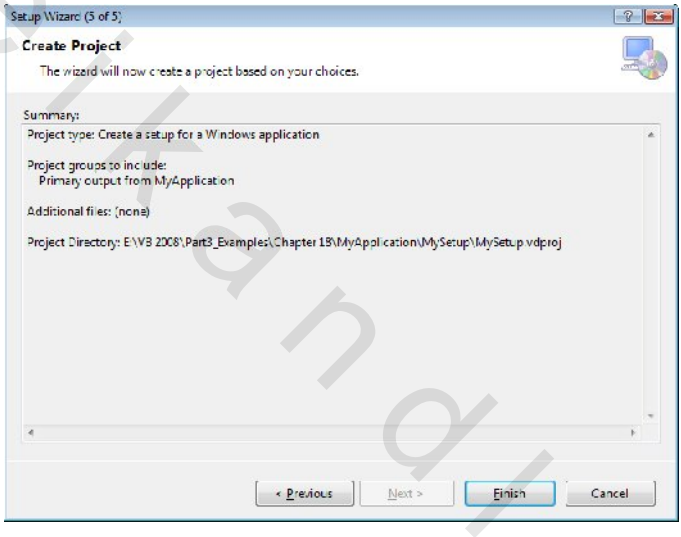
٨. قم بتنشيط مربع الاختيار **Primary Output** ثم انقر زر **Next**، تظهر النافذة التالية من المعالج والتى يمكنك من خلالها إضافة بعض الملفات الإضافية التى ترغب فى نشرها إلى المستخدم مثل الملفات **Read Me** أو ملفات التوثيق والمساعدة وقواعد البيانات وغيرها من الملفات الأخرى (انظر شكل ١٨–٥).



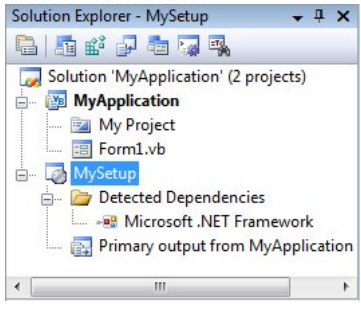
شكل ١٨–٥ إضافة الملفات إلى مشروع الإعداد

٩. لإضافة أيٍ من هذه الملفات، انقر زر **Add** ثم اختر الملف المطلوب من نافذة **Add File**. وبمجرد الانتهاء من إضافة ملفاتك، انقر زر **Next** للانتقال إلى النافذة الأخيرة من المعالج والتى تحتوى على ملخص للعمليات التى سيقوم المعالج بتنفيذها (انظر شكل ١٨-٦).

١٠. انقر زر **Finish**، يتم إنشاء مشروع إعداد جديد وإضافته إلى الحل المفتوح ودلالة ذلك ظهور المشروع داخل نافذة مستكشف الحل (انظر شكل ١٨-٧).



شكل ١٨-٦ تحتوى النافذة الأخيرة من المعالج على العمليات التى سيقوم المعالج بتنفيذها



شكل ١٨-٧ إضافة مشروع الإعداد إلى الحل المفتوح

حيث يحتوى مشروع الإعداد على ملف بالامتداد **msi.** الذى يستخدمه مثبت **Windows** لتثبيت البرنامج على جهاز المستخدم بالإضافة إلى الملف التنفيذى.

تعيين خيارات مشروع الإعداد

يحتوى مشروع الإعداد على العديد من الخصائص والخيارات التى تتحكم فى سلوكه مثله كجميع أنواع المشروعات الأخرى. وحينما يقوم المعالج بإنشاء مشروع الإعداد، يقوم بتعيين العديد من الخيارات نيابةً عنك، إلا أن بعض هذه الخيارات يحتاج إلى التعديل كى يتناسب مع ميولك وذوقك الخاص والوظائف التى يقوم التطبيق بتنفيذها. لتغيير أو توفيق خصائص مشروع الإعداد، قم باختيار المشروع من داخل نافذة مستكشف الحل، ثم قم بتغيير هذه الخصائص من نافذة مربع الخصائص المصاحبة. يوضح جدول ١٨-١ التالى أهم الخصائص التى تحتاج غالباً إلى التعامل معها عند إنشاء مشروعات الإعداد.

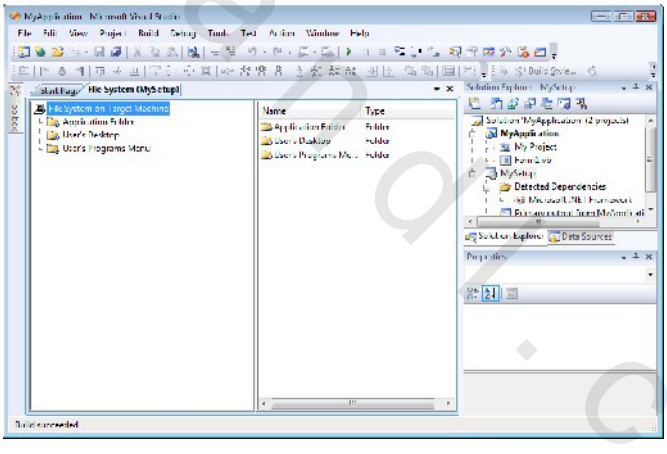
جدول ١٨-١ خصائص مشروع الإعداد

الخاصية	الاستخدام
AddRemoveProgramsIcon	تستخدم هذه الخاصية لتعيين اسم الملف الذى يحتوى على رمز التطبيق الذى يظهر بالمربع الحوارى Add/Remove Programs بنافذة لوحة التحكم Control Panel الموجودة لدى المستخدم
Author	تستخدم هذه الخاصية لتعيين اسم مؤلف البرنامج (القائم على إعداده)
Description	تستخدم هذه الخاصية لتعيين وصف مختصر للمشروع
DetectNewerInstalledVersion	إذا احتوت هذه الخاصية على القيمة True (القيمة الافتراضية)، يقوم برنامج الإعداد

الخاصية	الاستخدام
	بالبحث عن إصدار أعلى من نفس المشروع مثبتاً لدى المستخدم. فإذا وجد هذا الإصدار، لا يقوم بالاستمرار فى عملية التثبيت
Manufacturer	تستخدم هذه الخاصية لتعيين اسم الشركة المطورة للتطبيق
ProductName	تستخدم هذه الخاصية لتعيين اسم للتطبيق. والقيمة الافتراضية لهذه الخاصية هى اسم برنامج الإعداد نفسه، إلا أنه من المفضل تغيير هذا الاسم ليكون مقبولاً لدى المستخدم. فهى القيمة التى تظهر فى شريط عنوان معالج الإعداد وكذلك عند إظهار الرسائل للمستخدم
RemovePreviousVersions	إذا احتوت هذه الخاصية على القيمة True، يقوم برنامج الإعداد بحذف الإصدارات السابقة من التطبيق والتى يقوم باكتشافها أثناء عملية الإعداد. والقيمة الافتراضية لهذه الخاصية هى False
Title	تستخدم هذه الخاصية لتعيين عنوان لحزمة التثبيت، والقيمة الافتراضية لهذه الخاصية عبارة عن اسم المشروع، إلا أنه من المفضل أن تقوم بتعيين عنوان محبب للمستخدم كما ذكرنا فى الخاصية Product Name

توفيق نظام ملفات المستخدم

يعتبر تعديل نظام الملفات من الوظائف الأساسية لبرنامج الإعداد. فإذا أردت التحكم فى التغييرات التى يقوم برنامج الإعداد بتنفيذها، انقر العنصر **Primary Output** الموجود بمجلد الإعداد داخل نافذة مستكشف الحل نقراً مزدوجاً، تظهر نافذة نظام الملفات **File System** (انظر شكل ١٨-٨). فعادةً يقوم المعالج بإنشاء برنامج الإعداد بحيث يتم تثبيت ملفات الخرج الأساسية (ملف **EXE** أو ملف **Dll**) إلى مجلد **Program Files** الموجود على جهاز المستخدم، إلا أنه لا يقوم بإضافة أى رموز إلى قائمة **Start** أو أى اختصارات أخرى إلا إذا قمت بتعيين ذلك بنفسك. سنقوم فيما يلى بالتعرف على كيفية تعديل المكان الذى يتم وضع ملفات التطبيق فيه بالإضافة إلى كيفية التحكم فى إنشاء اختصارات التطبيق على حاسب العميل.



شكل ١٨-٨ التحكم فى تغييرات نظام الملفات من خلال نافذة **File System**

تعيين مجلد التطبيق

كما ذكرنا منذ قليل، فإن التعديل الوحيد الذى يقوم به المعالج هو وضع خرج البرنامج بمجلد خاص داخل المجلد **Program Files** الموجود لدى المستخدم، حيث توجد قيمة هذا المجلد داخل الخاصية **Default Location**. يمكنك تعيين اسم مختلف لمجلد التطبيق من خلال هذه الخاصية. لأداء ذلك، تابع معنا الخطوات الآتية:

١. انقر المجلد **Application Folder** من العمود الأيسر بنافـذة **File System**، وحينئذٍ تظهر نافذة الخصائص لتعكس خصائص مجلد التطبيق.

٢. بالنظر إلى الخاصية **Default Location** داخـل نافـذة الخصـائص، تلاحـظ احتواءها على القيمة الافتراضية الآتية:

[ProgramFilesFolder][Manufacturer]\[ProductName]

وهو ما يعنى وضع ملفات التطبيق داخل المجلد المحدد بالخاصية **ProductName** والذى يتم إنشاؤه داخل مجلد بالقيمة المحددة داخل الخاصـية **Manufacturer** والذى يقع بدوره داخل المجلد العام **Program Files**. لذا فإذا احتوت الخاصية **Manufacture** مثلاً على القيمة **Compuscience** بينما احتـوت الخاصـية **ProductName** على القيمة **CompuLibrary**، فـإن المكـان الافتراضـى لتخزين ملفات التطبيق هو:

C:\Program Files\Compuscience\CompuLibrary

هذا على فرض أن مكان المجلد **Program Files** هو **C:\Program Files**.

٣. إذا أردت تضمين المزيد من الملفات داخل المجلد **Application**، انقر هذا المجلـد بزر الفأرة الأيمن ثم اختر **Add,File** من القائمة الموضعية.

يستطيع المستخدم بالطبع تغيير المجلد الافتراضى للتطبيق فى بداية تشغيل عملية الإعداد على حاسبة.

إضافة اختصار التطبيق إلى قائمة Start

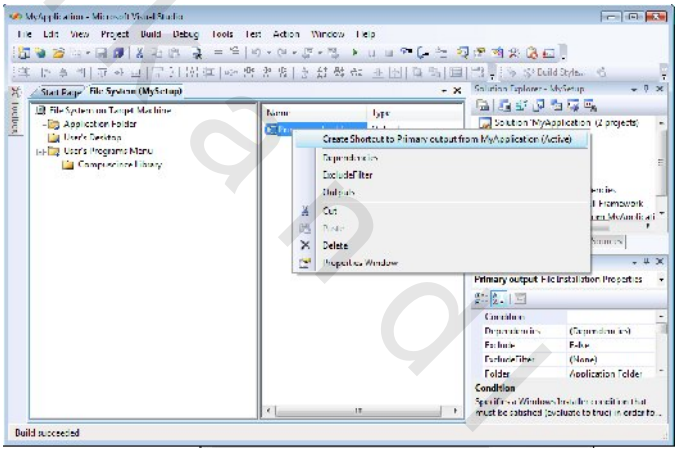
يمكنك إضافة اختصار أو أكثر إلى قائمة **Start** كى يتمكن المستخدم من تشغيل التطبيـق من خلاله، حيث لا يقوم معالج مشروع الإعداد افتراضياً بتعيين هذه الاختصارات. تحتوى نافذة **File System** على المجلد **User's Programs Menu** الذى يحتـوى بـدوره على مجلد التطبيق الموجود بمجموعة **All Programs** داخل قائمة **Start** الموجودة لدى المستخدم. لتعيين هذا المجلد بما فيه من اختصارات، تابع معنا الخطوات الآتية:

١. من نافذة **File System**، انقر المجلد **User's Programs Menu** بزر الفأرة

الأيمن ثم اختر Add وFolder من القائمة الموضعية الناتجة، يتم إنشاء مجلد جديد داخل المجلد User's Programs Menu.

٢. قم بإعادة تسمية المجلد الجديد بتعيين اسم مناسب ليحتوى على الاختصارات المختلفة للتطبيق.

٣. انقر المجلد Application Folder بالعمود الأيسر من النافذة File System ثم انقر رمز Primary Output من العمود الأيمن للنافذة بزر الفأرة الأيمن واختر Create shortcut to primary output من القائمة الموضعية (انظر شكل ١٨-٩).



شكل ١٨-٩ إنشاء الاختصار من خلال الملف التنفيذى للتطبيق

٤. يتم إنشاء اختصار للملف التنفيذى للتطبيق وإضافته إلى المجلد Application Folder. قم بتغيير اسم الاختصار كى يكون معبراً (قم على الأقل بحذف عبارة Shortcut to).

٥. استخدم زر الفأرة لسحب الاختصار من المجلد Application Folder إلى المجلد الفرعى الجديد الذى قمت بإنشائه وإضافته إلى المجلد User's Programs Menu فى الخطوة رقم ١.

والآن حينما يقوم المستخدم بتشغيل برنامج الإعداد على حاسبه، يتم إضافة المجلد باختصارات التطبيق إلى قائمة **Programs** المتفرعة من قائمة **Start**.

يمكنك استخدام نفس الخطوات السابقة لإضافة المزيد من الاختصارات الخاصة بملفات التوثيق أو المساعدة مثلاً إلى المجلد الخاص بالتطبيق داخل قائمة **Start**.

إضافة الاختصارات لسطح مكتب المستخدم

يمكنك تثبيت الملفات أو الاختصارات على سطح المكتب الموجود لدى المستخدم باتباع نفس الخطوات السابقة المستخدمة لتعيين اختصارات قائمة **Start** ولكن مع استخدام المجلد **User's Desktop** بدلاً من المجلد **User's Programs Menu**.

بناء مشروع الإعداد

بمجرد الانتهاء من تهيئة وتوفيق مشروع الإعداد بخياراتك، يجب أن تقوم ببناء هذا المشروع لإنشاء حزمة التثبيت. لبناء المشروع، تابع معنا الخطوات الآتية:

١. من نافذة مستكشف الحل، انقر اسم مشروع الإعداد بزر الفأرة الأيمن ثم اختر **Properties** من القائمة الموضعية، تظهر نافذة **Property Pages** الخاصة بالمشروع.

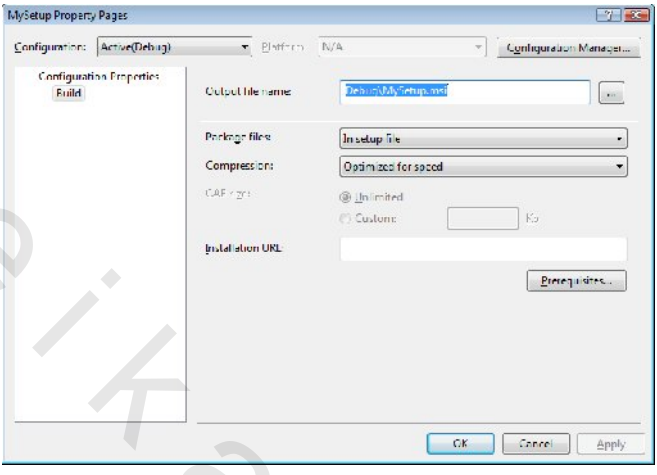
٢. من مربع السرد **Configurations**، قم باختيار شكل البناء الذى ترغب فيه. فإذا كنت لم تزل فى مرحلة الاختبار، ابق على الخيار الافتراضى **Active(Debug)** كما هو، وإلا اختر **Release** إذا قررت نشر التطبيق (انظر شكل ١٨-١٠).

٣. يمكنك أيضاً تغيير اسم ملف خرج التطبيق إذا أحببت وذلك من خلال مربع النص **Output File Name**.

٤. بمجرد الانتهاء من إعداداتك، انقر زر **Ok** لإغلاق مربع الخصائص.

٥. من نافذة مستكشف الحل، انقر اسم مشروع الإعداد بزر الفأرة الأيمن مرةً أخرى ثم اختر **Build** من القائمة الموضعية، يتم ترجمة المشروع وبنائه وإنشاء ملف **MSI**.

النهائى. (ربما تستمر هذه العملية بعض الدقائق. يتوقف ذلك على تطبيقك وإعدادات حاسبك).



شكل ١٨-١٠ مربع خصائص مشروع الإعداد

إذا قمت بالتعديل فى التطبيق الأساسى بعد آخر مرةً قمت فيها ببناء هذا التطبيق، يجب أن تقوم ببناء التطبيق مرةً أخرى قبل أن تقوم ببناء مشروع الإعداد حتى يتم تضمين تغييراتك فى النسخة التى سيتم نشرها.

٦. بمجرد اكتمال عملية بناء مشروع الإعداد، يتم إنشاء ملف الخرج النهائى **MSI.** وتخزينه داخل المجلد **Debug** أو **Release** تحت مجلد مشروع الإعداد وذلك تبعاً لنمط البناء الذى اخترته من مربع خصائص مشروع الإعداد فى الخطوة رقم ٢.

نشر مشروع الإعداد

بمجرد الانتهاء من مشروع الإعداد، يمكنك نشر ملف **MSI** وملف **EXE** الناتجين إلى مستخدميك. وحينما يقوم المستخدم بتشغيل هذا الملف، يتم مصاحبته فى عملية تثبيت رسومية كالتى تتم مع التطبيقات والبرامج الأخرى.

تحويل المشروعات القديمة

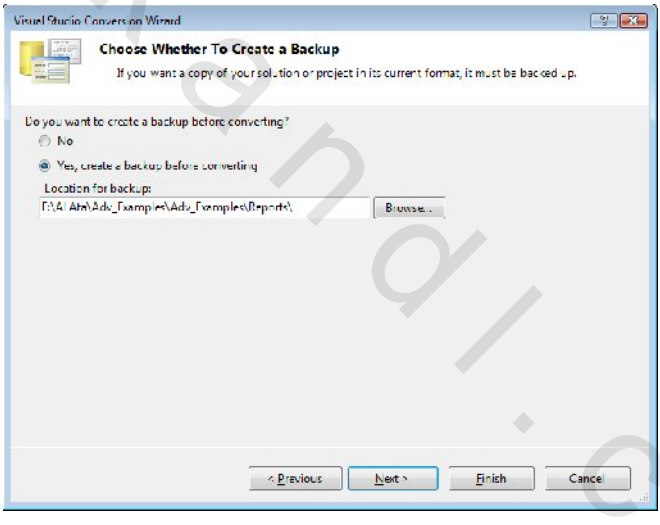
يحتوى Visual Studio 2008 على معالج تحديث Visual Basic الذى يمكنك استخدامه فى تحويل المشروعات المنشأة بواسطة Visual Basic 6.0 أو Visual Basic.NET أو Visual Basic 2005 إلى Visual Basic 2008، حيث يمكنك استخدام المعالج فى ترقية المشروع القديم إلى مشروع Visual Basic 2008 مع القدرة على الإبقاء على النسخة القديمة من المشروع كما هى. للتعرف على طريقة استخدام المعالج فى تحويل التطبيقات القديمة، تابع معنا الخطوات الآتية:

١. قم بفتح بيئة تطوير Visual Studio 2008 إذا لم تكن مفتوحةً بالفعل.
٢. افتح قائمة File من شريط القوائم ثم اختر Open Project من القائمة المنسدلة، يظهر المربع الحوارى Open Project.
٣. اختر الملف الخاص بمشروع Visual Basic القديم الذى ترغب فى تطويره إلى Visual Basic 2008 (يكون الملف عادةً بالامتداد vbp.أو vbproj.).
٤. انقر زر Open، تظهر النافذة الأولى بمعالج التحويل (انظر شكل ١٨-١١).

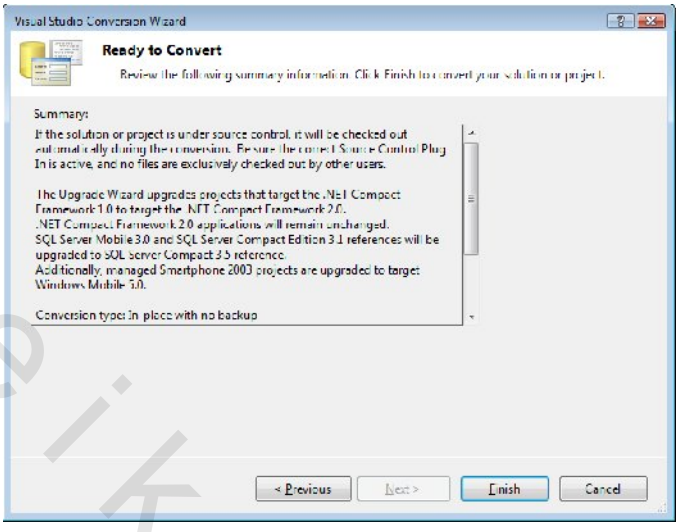
شكل ١٨-١١ النافذة الأولى بمعالج التحويل

٥. انقر زر **Next**، تظهر النافذة التالية للمعالج والتى تسألك إذا ما كنت ترغب فى الإبقاء على المشروع القديم كما هو مع إنشاء نسخة جديدة محولة، أو تحديث النسخة القديمة فقط (انظر شكل ١٨-١٢). نشط زر الاختيار **No** إذا أردت عدم إنشاء نسخة احتياطية من المشروع القديم، أو زر الاختيار **Yes, create a backup before converting** إذا أردت إنشاء نسخة احتياطية من المشروع القديم وقم فى هذه الحالة بتحديد مكان هذه النسخة بمربع النص **Location for backup** الذى يحتوى افتراضياً على المكان الحالى للمشروع.

٦. انقر زر **Next**، تظهر النافذة التالية من المعالج والتى تحتوى على ملخص لعملية التحويل (انظر شكل ١٨-١٣).

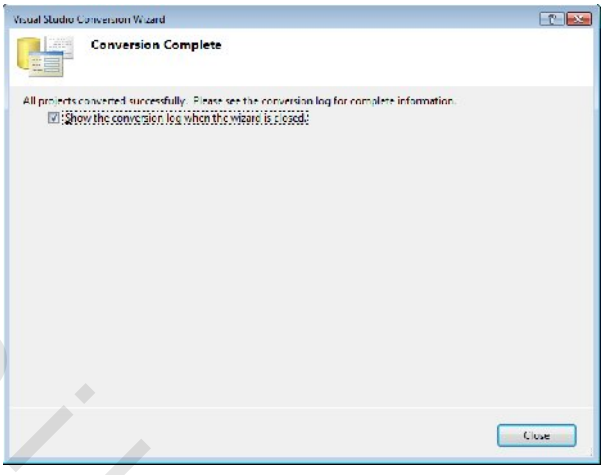


شكل ١٨-١٢ يسألك المعالج إذا كنت ترغب فى إنشاء نسخة احتياطية من المشروع القديم

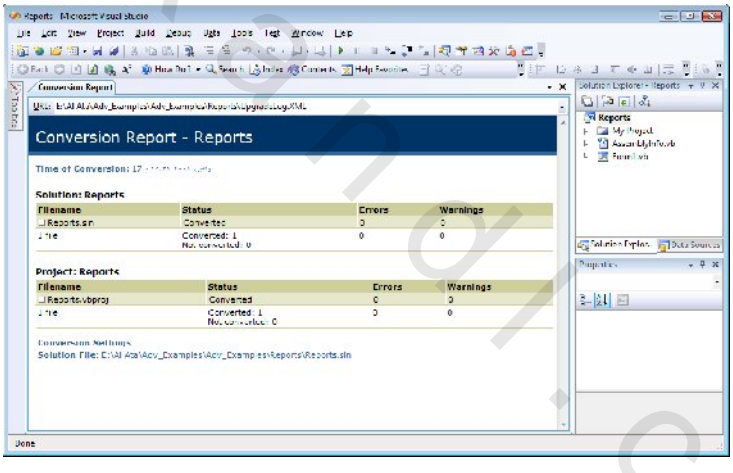


شكل ١٨-١٣ تحتوى هذه النافذة على ملخص لعملية التحويل

٧. انقر زر **Finish**، وحينئذٍ يقوم المعالج بإجراء عملية التحويل ويسألك إذا كنت ترغب فى التعرف على ما تم خلالها (انظر شكل ١٨-١٤). نشط مربع الاختيار **Show the conversation log when the wizard is closed** إذا أردت التعرف على تفاصيل عملية التحويل (انظر شكل ١٨-١٥) ثم انقر زر **Close** لإنهاء المعالج وقد تم تحويل المشروع القديم إلى مشروع بتنسيق **Visual Basic 2008** مع فتح المشروع الجديد داخل بيئة تطوير **Visual Studio 2008**.



شكل ١٨-١٤ تحويل المشروع بنجاح



شكل ١٨-١٥ المشروع الجديد داخل بيئة التطوير وملخص عملية التحويل