

تعلم برمجة فيجوال بيسك ١

خطوة بخطوة

الجزء الأول : بناء قواعد بيانات أكسيس ولغة الإستعلامات الهيكلية

الجزء الثاني : بناء برنامج قواعد بيانات باستخدام فيجوال بيسك

الجزء الثالث : بناء برنامج سوبر ماركت

خطوة بخطوة

الجزء الرابع : بناء موقع باستخدام Asp للسبر ماركت

إعداد وتقديم

أحمد جمال خليفة

مقدمة .

بسم الله الرحمن الرحيم ، والصلاة والسلام على أشرف الأنبياء والمرسلين ، نبينا محمد وعلى آله وصحبه أجمعين .

اللهم لا سهل إلا ما جعلته سهلاً وأنت تجعل الحزن سهلاً إذا شئت

أما بعد ... فهذا الكتاب يحتوي على مجموعة من الدروس التي كتبت أجزاء منها في منتديات vb4arab ... وتختص بالتعامل مع الفيجوال بيسك وقواعد البيانات إضافة إلى التعامل مع برمجة المواقع من خلال Asp وهو موجه أساساً إلى المبتدئين في الفيجوال بيسك ٦ وبرمجة المواقع .

وهذا هو الجزء الثالث وهو لب الدورة والخاص ببناء البرنامج

خطوة بخطوة

أسأل الله أن يكون هذا الكتاب مفيداً ، فما كان فيه من صواب فمن الله ، وما كان من خطأ فمن نفسي والشيطان ، والله الهادي إلى سواء السبيل .

أخوكم : أحمد جمال خليفة عبد العال .

Hammada2091@Hotmail.com

القسم الثالث

عمل برنامج لإدارة

السوبر ماركت

باستخدام

Visual Basic 6.0

- المرحلة الأولى -

في هذا الفصل :

- تحليل النظم
- القوائم
- نوافذ البرنامج .

تحليل النظم

إن تحليل النظم Software Anylsis هو احد اهم الخطوات التي يتم تجاهلها بصورة كبيرة - على الاقل في البرامج العربية - مع أنها الخطوة الأولى والأساسية في أي من البرامج الناجحة ، وبرغم من أن البرنامج الذي ننوي انشاءه سوياً بإذن الله هو من النوع الصغير جداً ، إلا هذا لا يمنعنا من أن نلقي نظرة سريعة على اساسيات تحليل النظم .

فوائد تحليل النظم .

توفر هذه الخطوة العديد من المميزات ، منها على سبيل المثال :

- * تكوين نظرة مبدئية شاملة على كافة عناصر النظام .
- * تقدير صحيح بنسبة كبيرة للوقت والجهد الذي سينجز خلاله البرنامج .
- * التأكد من كل عناصر النظام قبل البدء مما يمنع بإذن الله حصول التقديرات الخاطئة .
- * امكانية التعديل بسهولة في حالة حدوث خطأ أو في حال طلب تطوير جزء من النظام .
- * توزيع واقعي للأعمال في مجموعات العمل .

قبل البداية .

قبل البدء بالعمل قم بالتأكد من استيعابك لكامل نقاط النظام ، وإذا بقيت نقاط غابت عن ذهنك فسوف تظهر لاحقاً في مراحل التحليل الأخرى .

تأكد من شمولية النظرة التي ترى فيها النظام ، من حيث طبيعة عمل النظام ، ما هي المدخلات المتاحة ، والمخرجات التي يتوقعها منك عمليات النظام شاملة العمليات التي يتوقع أن يقوم بها نظامك بالنيابة عنه سواء كعمليات حسابية أو تخزينية أو غير ذلك . كذلك معرفة حدود النظام وكمية بياناته ومقدار دقة البيانات وغير ذلك مهم جداً .

البداية .

في هذه الدورة سوف نقوم بإذن الله - كما أوضحنا سلفاً - بإنشاء برنامج بسيط لإدارة السوبر ماركت ، ولهذا سوف نقوم بدراسة حالة البرنامج ، ما الذي يطلبه قبل البدء في الخطوات اللاحقة .

مهام البرنامج .

- * تخزين جميع البضائع الموجودة في المحل وأسعارها وكمياتها - كافة المعلومات عنها - وعرضها عند الطلب ، كما يتم اضافة العناصر المتشابهة تحت قسم واحد (عصيرات - مياه ... الخ) .
- * تنفيذ عمليات البيع والشراء والتي تؤثر - بكل تأكيد - في كميات البضائع المتوفرة .
- * اضافة أسماء الموردين ومعلوماتهم (مندوبي المبيعات) وتخزين بضائع كل مندوب .
- * تقارير عن عمليات البيع والشراء . وعن الأنواع والبضائع ، ومندوبي المبيعات .
- * تقارير عن المبيعات وخلافه .

المدخلات المتوقعة :

- * اضافة البضائع تتطلب شاشة لادخال كافة معلومات البضاعة .
- * ونفس الأمر للأنواع ومندوبي المبيعات - الموردين - .
- * اضافة عملية بيع ، أو عملية شراء .

المخرجات المتوقعة :

المخرجات المتوقعة هي مجموعة من التقارير للمبيعات والبضائع ، وشاشات لعرض البضائع والكميات وغير ذلك .

إن هذه المقدمة السريعة تحدد هدف النظام ، والعمليات التي سوف تجري فيه لن أقوم بشرح خطوات تحليل النظم ، واكتفي بأن أحيلك إلى هذا الرابط السريع - باللغة الانجليزية - :

http://www.netnam.vn/unescocourse/sys_ana_des/21.htm

ويمكنك العثور على مزيد من الدروس والمواقع على شبكة الانترنت ، ولكن الجيد منها موجود باللغة الانجليزية - لم أعتز على رابط باللغة العربية - .

إذا كنت قد اطلعت على الرابط السابق ، فإنك الآن تستطيع تكوين DFD و ERD لبرنامجنا مع ما يحلقهما ، ولكن هذا ليس ضرورياً في سير الدورة ، ولذا فهو ليس اجبارياً .

ولن اطلع على الرابط ، وأراد تكوين DFD للبرنامج ، فإنه ما زال محتاجاً لمعرفة خطوات سير البرنامج ، في البداية يمكن أن تكون هذه الخطوات نموذج مبسط من سير البرنامج .

- * ادخال بيانات المورد - تخزينها في جدول الموردين .
- مطلوب في هذه الخطوة : بيانات المورد .
- * ادخال بيانات النوع - تخزينها في جدول الأنواع .
- مطلوب في هذه الخطوة : بيانات النوع .
- * ادخال بيانات البضاعة - تخزينها في جدول البضائع .
- مطلوب في هذه الخطوة : بيانات البضاعة + موردها + نوعها .
- * تطبيق عملية بيع على بضاعة .
- مطلوب في هذه الخطوة : بيانات عملية البيع + البضاعة .
- * تطبيق عملية شراء على بضاعة (وعملية الشراء الكاملة تشتمل على عدد من البضائع) .
- مطلوب في هذه الخطوة : بيانات عملية الشراء + البضاعة .
- * عرض تقرير بالبضائع الموجودة في المخزن .
- مطلوب في هذه الخطوة : بيانات البضائع والموردين والأنواع + عمليات البيع والشراء عليها .
- * عرض تقرير الأنواع .
- مطلوب في هذه الخطوة : بيانات الأنواع + بيانات بضائعه وعمليات البيع والشراء عليها .
- * عرض تقرير الموردين .
- مطلوب في هذه الخطوة : بيانات الموردين + بيانات بضائعه وعمليات البيع والشراء عليها .
- * عرض التقرير اليومي أو على فترة .
- مطلوب في هذه الخطوة : تاريخ اليوم أو الفترة + عمليات البيع والشراء التي تمت خلالها .

وكما هي الاضافة فهناك التعديل والحذف ايضاً في أغلب العمليات السابقة .

وهناك أيضاً عناصر أخرى للبرنامج مثل .

* عمليات البحث .

* حماية قاعدة البيانات . ومعلومات السوبر ماركت وخلافه .

* نسخ واستيراد قواعد البيانات .

وهكذا تستطيع تكوين DFD خاص بالبرنامج ،

لا تنسى ان ثم مراحل وصفية أخرى قبل الانتقال إلى المرحلة التالية وهي تكوين ERD .

ERD

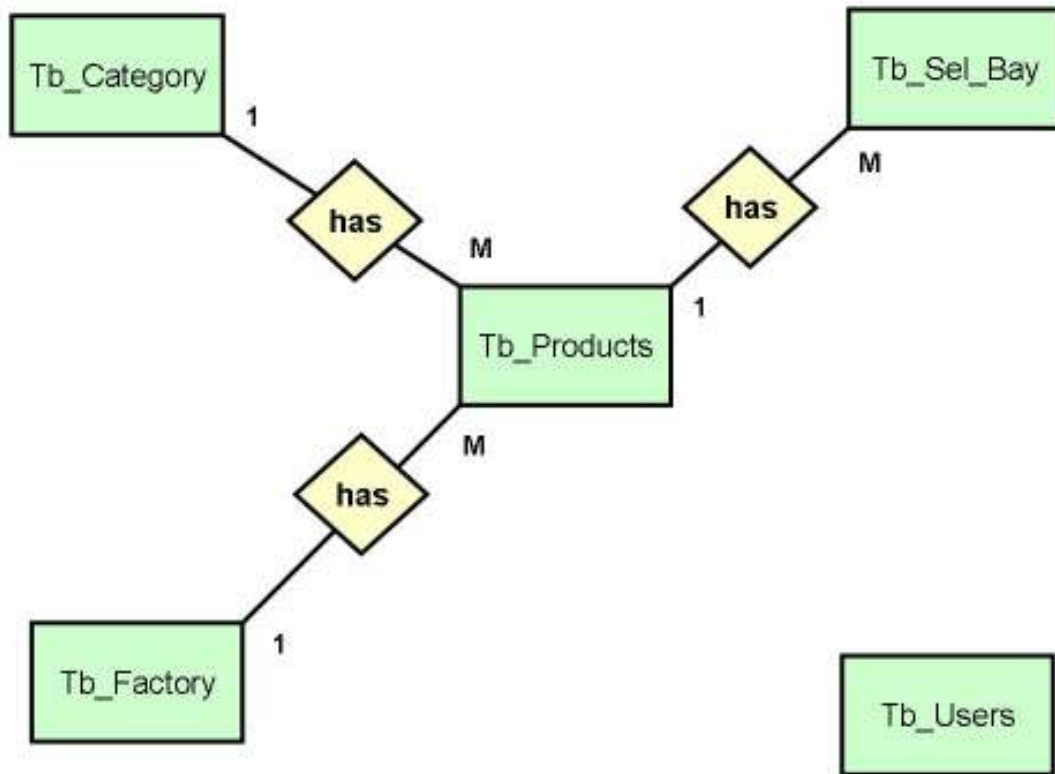
من سير البرنامج السابق نستطيع أن نستبين عناصر حفظ البيانات المطلوبة وهي :

- * جدول بيانات الموردين .
- * جدول بيانات الأنواع .
- * جدول بيانات البضائع .
- * جدول عمليات البيع والشراء .
- * وجدول اضافي آخر وهو الخاص بالحماية ومعلومات المحل وغير ذلك .

كما وسنجد العلاقات التالية بين الجداول السابقة :

- * علاقة One-Many بين الموردين والبضائع (مورد واحد يورد مجموعة من البضائع) .
- * علاقة One-Many بين الأنواع والبضائع (نوع واحد يحوي مجموعة من البضائع) .
- * علاقة One-Many بين البضائع وعمليات البيع والشراء (صنف بضاعة واحد يدخل في العديد من عمليات البيع والشراء) .

هذه الصورة توضح ERD الخاص بقاعدة البيانات الخاصة بنا :



والآن سنرى الحقول داخل كل جدول ، وسنلتزم منذ الآن فصاعداً بالأسماء لاضافتها لاحقاً في قاعدة البيانات .

أولاً : جدول الموردين . الاسم Tb_Factory

الحقول :

* Number رقم المورد ، وهو من نوع Auto Number – ترقيم تلقائي ، كذلك فهو Primary Key – مفتاح رئيسي .

* Name : اسم المورد ، وهو من النوع Text – نص ، وطول الحقل Field Size لا ينبغي أن يزيد على ٣٠ مثلاً .

ثانياً : جدول الأنواع . الاسم Tb_Category

الحقول :

* Number رقم النوع ، وهو من نوع Auto Number – ترقيم تلقائي ، كذلك فهو Primary Key – مفتاح رئيسي .

* Name : اسم المورد ، وهو من النوع Text – نص ، وطول الحقل Field Size لا ينبغي أن يزيد على ٣٠ مثلاً .

سنضيف أيضاً حقل تحت اسم View من نوع Yes/No – نعم / لا ، وهذا الحقل يقوم بتوضيح الموردين أو الأنواع الذين يتم حذفهم ، وذلك حتى إذا انتهت آخر بضاعة تحت أي منهم يتم حذفه بالكلية من الجدول .

أو بمعنى آخر : عند الحذف سوف يأخذ هذا الحقل قيمة False مؤقتاً حتى يتم حذف آخر بضاعة تقع تحته ، في هذه الحالة فقط سوف يحذف المورد أو النوع .

ثالثاً : جدول البضائع . الاسم Tb_Product

الحقول :

* Number رقم النوع ، وهو من نوع Auto Number – ترقيم تلقائي ، كذلك فهو Primary Key – مفتاح رئيسي .

* Name : اسم البضاعة ، وهو من النوع Text – نص ، وطول الحقل Field Size لا ينبغي أن يزيد على ٣٠ مثلاً .

* Factory المورد أو مصنع البضاعة ، وهو من النوع Number – رقم ، وحجم الحقل Field Size هو عدد صحيح طويل ، وسيتم ربطه لاحقاً بجدول الموردين .

* Category النوع ، وهو من النوع Number – رقم ، وحجم الحقل Field Size هو عدد صحيح طويل ، وسيتم ربطه لاحقاً بجدول الأنواع .

* Price وهو سعر الوحدة (الأساسي مع امكانية تغييره في عمليات البيع والشراء) وهو من نوع Number – رقم ، وطول الحقل Field Size هو (مزدوج – Double) وذلك لاتاحة الفرصة للكسور (مثلاً ٥.٥) .

* Count وهو العدد الأولي للبضاعة ، والذي سيزيد أو ينقص تبعاً لعمليات البيع أو الشراء . وهو من نوع عدد صحيح طويل .

ملاحظة هامة : هذا الحقل الأخير لا بد أن لا يكون له وجود ، وبديله هو طرح عمليات الشراء من عمليات البيع حيث تعطينا العدد الحالي وذلك لتوفير التكرار في قواعد البيانات . ولكن تم وضعه للتسهيل .

* Box_Count عدد الوحدات في كل صندوق ، ويفيد هذا الحقل في حالة اضافة بضائع بالصندوق أو بيعها بنفس الطريقة لمعرفة عدد الوحدات المباعة . وهذا الحقل من نوع Number – رقم ، وكذلك من طول : عدد صحيح طويل .

رابعاً : جدول عمليات البيع والشراء . الاسم Tb_Sel_Bay

*** تمثل كل عملية بيع بضاعة في هذا الجدول ، وإذا تم شراء عدد من البضائع مثلاً فإنه يتم اضافة سجل منفصل لكل منهم .

الحقول :

* Number رقم العملية ، وهو من نوع Auto Number – ترقيم تلقائي ، كذلك فهو Primary Key – مفتاح رئيسي .

* Product البضاعة ، وهو من النوع Number – رقم ، وحجم الحقل Field Size هو عدد صحيح طويل ، وسيتم ربطه لاحقاً بجدول البضائع .

* Date : تاريخ العملية ، وهو من النوع Date/Time – تاريخ /وقت . وفي حقل تنسيق – Format نقوم باختيار تنسيق Short Date .

* Count وهو العدد الأولي للبضاعة ، والذي سيزيد أو ينقص تبعاً لعمليات البيع أو الشراء . وهو من نوع عدد صحيح طويل .

* Price السعر الاجمالي لعملية البيع هذه (لبضاعة معينة) . وهو من نوع Number Kind هو حقل من نوع رقم Number لكنه من طول Byte – بايت ، وفي حالتنا فهو يحمل أحد قيمتين (٠ ، ١) للتفريق بين عمليتي البيع والشراء .

أيضاً يمكن جعل هذا الحقل من نوع Yes/No – نعم/ لا لتدل الحالة الأولى على النوع الأول (بيع مثلاً) والثانية على النوع الثاني .

خامساً : جدول المستخدمين . الاسم Tb_User

*** هذا الجدول لن يحتوي إلا على سجل واحد ويحتوي على معلومات المستخدم (ويمكن تطويره لعدة مستخدمين بسهولة) اما إذا ظل مستخدماً واحدة – كما هي العادة – فيمكن الاستعاضة عنه بتخزين هذه القيم في مستند نصي أو في مسجل النظام (الريجستري Rigistry) .

الحقول :

* User-Name : اسم المستخدم . وهو نص Text بطول أقصى ١٥ خانة .

* PassWord : كلمة المرور وهو أيضاً نص بحد أقصى ١٥ خانة .

* Name : اسم المحل أو خلافه ، وهو نص بطول ٥٠ خانة .

* Telephone : الهاتف ، وهو رقم من نوع عدد صحيح طويل .

*** يفضل تخزين الهاتف ، والهاتف الجوال وخلافه في حقول نصية دائماً . فهي أكثر توفيراً للمساحة كما تتيح لك اضافة رقم مثل ٠١٠ - صفر على اليمين - على عكس الحقول الرقمية .

* More : اي بيانات اضافية يرغب في اضافتها وعرضها ، وهو من نوع مذكرة .

والآن نقوم باضافة الجداول السابقة إلى قاعدة بيانات من نوع أكسيس كما تعلمنا في الدورة السابقة . وهنا لن نضطر لتحويل الاصدار إلى ٩٧ حيث أننا سنكمل مشوارنا مع Dao والتي تتعامل مع أنواع احدث من قواعد البيانات من Access .

بعد قيامك بتشغيل Microsoft Access قم بانشاء قاعدة بيانات تحت اسم Db1 ، بعد ذلك قم باختيار (جداول Tables) ، ومن ثم :

انشاء جدول في طريقة عرض التصميم - Create Table in Design View ، وبعد ذلك قم باضافة حقول كل جدول على حدة ، وبعد الانتهاء قم بحفظ الجداول .

بعد ذلك سنطبق العلاقات والتي تعرضنا لكيفية عملها في الأجزاء السابقة من الدورات ولكن باختصار : من قائمة Tools اختر RelationShip (أدوات . علاقات) من ثم اختر من الجدول الظاهرة الجداول التالية :

Tb_Product - Tb_Factory - Tb_Category - Tb_Sel_Bay
إذا لم يظهر جدول اضغط بزر الماوس الأيمن واختر (اظهر جداول - Show Tables) .

الآن قم بوضع زر الماوس على حقل Number من Tb_Factory واسحب منه على حقل Factory في الجدول Tb_Product .

سوف يظهر خط يصل بين الجدولين ، فيما ستظهر شاشة ، اجعل الثلاثة اختيارات صحيحة .

*** تحدد هذه الاختيارات لتحقيق تكامل بين الجدولين ، فمثلاً إذا تم حذف نوع فإن البضائع التي تحته يتم حذفها أيضاً . كذلك لن يتم اضافة بضائع تحت نوع غير موجود أصلاً ..

بنفس الطريقة انشأ علاقة من جدول Tb_Category مع حقل Category في جدول Tb_Product .

العلاقة الثالثة والأخيرة سوف تكون بين حقلي Number في Tb_Product وحقل Product في جدول Tb_Sel_Bay ... قم بسحب الماوس من الحقل Number باتجاه حقل Product بنفس الطريقة السابقة .

أخيراً نقوم بإضافة كلمة سر ، وكما تعلمنا سابقاً نقوم بإغلاق قاعدة البيانات ، ومن ثم نقوم بفتحها بطريقة Exclusive ، بعد ذلك قم بوضع كلمة المرور من قائمة : أدوات Tools – أمان Securty ، وبعد ذلك اختر تعيين كلمة مرور .

والآن هذه هي قاعدة البيانات كاملة وكلمة السر هي VB4ARAB .

ملف مرفق :
قم بمطالعة الملف db1.zip في المرفقات لعرض قاعدة البيانات

القوائم

✳ منذ هذا الدرس فصاعداً وحى نهاية هذه الدورة سنعمل بإذن الله من خلال برنامج فيجوال

بيسك Visual Basic .

أولاً : القوائم Menues .

إن إحدى أهداف التحليل لأي برنامج هو معرفة مهام هذا البرنامج بالضبط وبالتالي امكانية تكوين القوائم وشاشات البرنامج دون خوف من امكانية التعديل فيها لاحقاً – إلا قليلاً – ، ومن تحليلنا

السابق نستطيع تجميع كل مجموعة متشابهه من المهام تحت قائمة رئيسية واحدة ، وذلك بالشكل التالي :

ملف : عرض البضائع - تعليمات الاستخدام - حول البرنامج - انتهاء .
اضافة : اضافة بضاعة جديدة - مورد جديد - نوع جديد .
تقارير : تقارير البضائع - تقارير الموردين - تقارير الأنواع - تقارير سريعة : (التقرير اليومي - تقرير بين تاريخين) .
مهام : عملية بيع - عملية شراء - مسح (كافة البضائع - كافة الموردين - كافة الأنواع - عمليات البيع - عمليات الشراء) - التحكم بعمليات المبيعات .
بحث : البحث عن البضائع .
أدوات : تغيير معلومات المحل - تغيير كلمة المرور - النسخ الاحتياطي (عمل نسخة احتياطية - استيراد نسخة احتياطية) - تجهيز الطابعة - آلة حاسبة .

*** بعض هذه العناصر لا علاقة لها بقواعد البيانات بل قد تكون بناء على طلب من المستخدم ، مثل (آلة حاسبة) أو من المسلمات في برامج قواعد البيانات مثل (عمل نسخة احتياطية) .

سنقوم الآن بمعرفة أنواع الواجهات في فيجوال بيسك ، بعد ذلك سوف نقوم باضافة القوائم إلى واجهة البرنامج الرئيسية .

ثانياً : الشاشات والنماذج في فيجوال بيسك .

عندما تختار اضافة نافذة Form جديدة إلى برنامجك من خلال قائمة Project - Add Form ، فإنك تلاحظ ظهور شاشة تحتوي على العديد من أشكال النماذج ، ومن اشهرها استخداماً Form بينما يمكنك التحويل إلى باقي الأشكال من خلال الخاصية Border Style والتي تحدد طبيعة شكل هذا النموذج فمثلاً إذا قمت باختيار :
None - .

فإن هذا يعطيك نموذجاً خالياً من شريط العنوان والأدوات ، غير ممكن تحريكه إلا بواسطة بعض الأوامر الا باستخدام بعض دوال API والتي تتيح لك تحريك الفورم من أي مكان . وهكذا ...

على أن هذه النوعية تشترك في أنها نوافذ لا يمكن أن تحتضن بداخلها نوافذ أخرى ، وبمعنى آخر فإنه لا يمكن عرض نافذة أخرى بداخلها بصورة تلقائية ، بل إنك إذ ذاك ستكون مضطراً لاستخدام بعض دوال API لوضعها على هذا النحو دون الحصول على نافذة حاضنة بنسبة ١٠٠٪ .

لعمل مثل تلك النافذة من Project قم باختيار MDI Form ، ولن يتيح لك الفيجوال بيسك ٦ سوى اضافة نافذة واحدة فقط من هذا النوع ، وفي الواقع لن تكون مضطراً لاستخدام أكثر من MDI Form في برنامج واحد في الغالب ، إلا أنك قد تحتاج لها فقط في حالة وجود أكثر من برنامج فرعي تحت البرنامج الرئيسي مثلاً .

ما هي نوافذ MDI ولماذا نستخدمها ؟

نستخدم هذه النافذة كنافذة حاضنة ، بمعنى أن هذه النافذة لن تحتوي سوى على القوائم واختصارات سريعة فقط ... أما المحتويات الأخرى فهي عبارة عن نوافذ أخرى تعرض داخل النافذة الأولى .
ميزة هذه النافذة أنها غالباً ما لا تحتوي على أي شيء ، لهذا نضعها دائماً أكبر ما يمكن ونضع أحجام النوافذ داخلها ثابتة .

وعند تغيير أسلوب العرض سوف تكبر هذه النافذة دون التأثير على النوافذ الداخلية ، مما يملئ الشاشة دون التأثير بعيوب تغيير الدقة Reslution في هذه الحالة .

عيوب نماذج MDI :

- * لا يمكن وضع أي أدوات عليها سوى أداة Picture وبالإمكان التحايل على هذا الأمر باضافة الأدوات التي نريدها داخل Picture .
- * لا يمكن وجود أكثر من نموذج MDI في البرنامج الواحد - من عيوب فيجوال بيسك والتي تم تلافيها في .net - .

اضافة نماذج MDI Form الخاصة بنا .

قم بتشغيل برنامج فيجوال بيسك ... وكما هي العادة سيظهر فورم عادي ... اضغط عليه بزر الماوس اليمين ، اختر New ومن ثم MDI Form .

نريد لهذه الشاشة أن تكون أكبر ما يمكن كما أسلفنا ، لذا اختر من WindowState القيمة ٢-
Maximized .

الآن سوف نقوم بجعل الفورم الأول محتويًا في داخلها ... لذا اذهب إليه واختر القيمة True
للخاصية MDIChild .

جرب تشغيل البرنامج ولاحظ التنفيذ .

هذه هي شاشتنا الرئيسية والتي سوف نعمل عليها ... لذا سنقوم باضافة قوائم بالمهام التي نريدها .

وقبل البدء في اضافة القوائم اقترح زيارة الرابط التالي ففيه شرح مبسط عن القوائم .

<http://www.opendirectorysite.info/121.htm>

وقد كنت كتبت درساً عن القوائم وما يتعلق بها ، ولكنه ضاع مع منتدى فيجوال بيسك للعرب .

كما أنه من الشائع - نسبياً وفي بعض الاحيان فقط - أن توضع كل قائمة باسم واحد فقط و Index
متغير بحيث أن قائمة ملف تأخذ الاسم M_File1 وتأخذ قائمة (حول البرنامج) الفرعية من
القائمة الأولى باسم File ولكن بالإندكس ، كما أن (تعليمات) تأخذ الاسم File ب Index=1
... وهكذا . وسوف نتعلم كيفية التعامل مع Index في عجالة بسيطة في آخر هذا الدرس .

ولا تستخدم هذه الطريقة في العادة إلا إذا لم تكن القوائم تحتوي إلا على استدعاءات لدوال مثلاً أو
مجرد اظهار شاشات جديدة ، وهذا هو أغلب ما يوجد في برنامجنا .

سنقوم الآن باضافة القوائم التي حدناها في أول هذا الدرس .

ثم نقوم باضافة بعض الاختصارات السريعة للمهام الرئيسية مثل Ctrl+F4 للخروج ، F1 لحول
البرنامج و F2 للتعليمات ، Ctrl+S لعملية بيع و Ctrl+B لعملية شراء ، وبعض الاختصارات
التي سوف تجدها في الملف المرفق .

لا تنس ضبط خاصية Right To Left = True من أجل دعم اتجاهات اللغة العربية .

والآن لنأخذ مثلاً سريعاً على كيفية استخدام القوائم .

عندما نضع امراً ما في File على سبيل المثال فهناك احتمال لأن يكون الزر المضغوط هو (حول - تعليمات - البضائع - انتهاء) لأن لهم نفس الاسم جميعاً ، ومن أجل ذلك فهناك متغير اسمه Index يحتوي على رقم Index وهو أحد بارميترات دالة Click .

ونستطيع عمل العملية المطلوبة لاحقاً باستخدام هذه الخاصية ، ضع الكود التالي في زر File . والذي يوضح طرق استخدام Index .

```
(Private Sub File_Click(Index As Integer)
Select Case Index
Case 0
"MsgBox "View Product
Case 5
End
Case Else
MsgBox File(Index).Caption
End Select
End Sub
```

نوافذ البرنامج

قبل بداية الدرس اقترح تحميل الملف والذي سنتعرف على طريقة تكوينه هذا الدرس .

ملف مرفق :
قم بمطالعة الملف programme.zip في المرفقات لمزيد من التفاصيل

في هذا الدرس سوف نقوم باضافة النوافذ الخاصة بالبرنامج ، وهي بالشكل التالي :

AboutMe : حول البرنامج .

لعمل مثل هذا النموذج نقوم باختيار Project - Add Form ومن ثم نختار الشكل About Dialog .

نقوم ببعض التغييرات لنصل إلى الشكل المرفق في البرنامج .

وستجد أن بعض الأكواد قد تمت اضافتها ، وهي تلك الأكواد الخاصة بـ System Info . وسوف نقوم بشرحها بصورة مبسطة لاحقاً .

Frm_splash : شاشة البداية .

من المفترض أن تظهر هذه الشاشة في بداية تشغيل البرنامج ولعدد من الثواني قبل أن تختفي ، ولذلك فلن تظهر أشرطة الادوات في هذه الشاشة ، كما ولا بد أن تحتوي على مؤقت لاختفاء الشاشة بعد مدة وجيزة .

Frm_Add : اضافة بضاعة جديدة .

شاشة ستبرمج لاحقاً لاضافة بضاعة جديدة .

Frm_Add_2 : اضافة نوع او مورد جديد .

ستخصص هذه الشاشة للتحكم واطافة أنواع وموردين ، وسنفرق بينهما لاحقاً بالكود .

Frm_Calc : الآلة الحاسبة .

شاشة مخصصة لآلة حاسبة بسيطة ، وهي فرصة لتتعرف سوية على طريقة عمل آلة حاسبة .

Frm_Edit : شاشة تعديل البضائع .

لتعديل معلومات بضائع موجودة .

Frm_Edit_SP : لتعديل عمليات البيع أو الشراء .

الأدوات الجديدة : سنجد في هذا الفورم أداة جديدة غير القياسية وهي أداة MaskedTextBox حيث توفر لنا هذه الأداة خاصية التحكم بشكل النصوص المدخلة بسهولة ، ولاضافتها نقوم بالضغط على Ctrl+T ومن ثم نقوم باختيار Microsoft Masked Edit Control 6.0 وسنتعرف على طريقة استخدامها لاحقاً .

Frm_Find : للبحث عن البضائع بشروط معينة .

الأدوات الجديدة : سنجد في هذا الفورم أداة جديدة غير الأدوات القياسية وهي أداة الجدول ولاضافتها نقوم بالضغط على Ctrl+T ومن ثم نقوم باختيار Microsoft Flex Grid 6.0 ، وسنتعرف على طريقة استخدامه لاحقاً .

Frm_Help : تعليمات الاستخدام والتي سيتم ربطها بملف من نوع rtf لاحقاً .

الأدوات الجديدة : سنجد في هذا الفورم أداة جديدة غير الأدوات القياسية وهي أداة RichText حيث توفر لنا هذه الأداة اضافة نصوص متغيرة التنسيق ، ولاضافتها نقوم بالضغط على Ctrl+T ثم نقوم باختيار Microsoft Rich Text Control 6.0 ، وسوف نتعرف على طريقة استخدامها لاحقاً .

Frm_Info : هذه الشاشة ستظهر في خلفية الشاشة الرئيسية لعرض معلومات المحل .
وذلك بالشكل التالي مثلاً :



Frm_Login : شاشة تسجيل الدخول .

Frm_NewUser : إذا كانت هذه هي المرة الأولى لتشغيل البرنامج ، يتم اظهار هذه الشاشة .

Frm_Printer : تحديد طابعة لاستخدامها في طباعة التقرير .

Frm_Report1 : تقارير البضاعة .

الأدوات الجديدة : سنجد في هذا الفورم أداة جديدة غير الأدوات القياسية وهي أداة MSChart وهي إحدى أدوات عرض الرسوم البيانية ، ولأضافتها نقوم بالضغط على Ctrl+T ومن ثم نقوم باختيار (Microsoft Chart Control 6.0 (OLEDB ... وسنتعرف على طريقة استخدامها لاحقاً .

Frm_Report2 : تقرير الأنواع .

Frm_Report3 : تقرير الموردين .

Frm_Report4 : التقرير اليومي ، وتقرير بين تاريخين .

Frm_Sel_Bay : اضافة عملية بيع أو شراء .

Frm_Sel_Bay2 : التحكم بعمليات البيع والشراء .

Frm_Show : عرض البضائع الرئيسي .

Frm_User : تغيير معلومات المستخدم .

Frm_Wait : شاشة للانتظار في العمليات المطولة للعرض أو الاضافة .

الأدوات الجديدة : سنجد في هذا الفورم أداة جديدة غير القياسية ، وهي أداة ProgressBar حيث يمكننا من خلالها عرض مؤشر للتقدم ، ولإضافتها نقوم بالضغط على Ctrl+T ومن ثم نقوم باختيار Microsoft Windows Common Control 6.0(SP6) ، ستلاحظ اضافة مجموعة من الأدوات من ضمنها هذه الأداة ، وسنتعرف على طريقة استخدامها لاحقاً .

MDIForm1 : الشاشة الرئيسية والتي قمنا بعملها سابقاً .

الأدوات الجديدة : سنجد في هذا الفورم مجموعة من الأدوات الجديدة :

أداة ToolBar لعرض شريط الأدوات .

أداة StatusBar لعرض الشريط السفلي للنافذة .

وهما موجودتان ضمن المجموعة التي اضعناها سابقاً .

أداة CommonDialog تقوم بعرض عدد من النوافذ الحوارية مثل مربع حوار فتح - حفظ -

اختيار الألوان - اختيار الخطوط ... الخ .

ولإضافتها نقوم بالضغط على Ctrl+T ومن ثم نقوم باختيار

Microsoft CommonDialog Control 6.0 ، وسنتعرف على طريقة استخدامها لاحقاً .

وعندما نتطرق في المرحلة القادمة بإذن الله للبرمجة ، سنتعلم بصورة مفصلة كيفية بناء كل شاشة مما سبق ، والأدوات التي اضيفت عليها .

وحتى ذلك الحين ، يمكنك الاطلاع على الملف السابق والذي يحتوي على تصميم الشاشات المختلفة كاملة .

*** منذ الآن فصاعداً قم باضافة قاعدة البيانات على نفس مسار البرنامج ، لا تقلق إذا لم تكن قد استخدمت بعض الأدوات من قبل لأننا سنشرح بالتفصيل كيفية استخدامها في دروس لاحقة .
منذ سلسلة الدروس القادمة سوف يتم البدء ببرمجة البرنامج بإذن الله من خلال فيجوال بيسك + Dao .

القسم الثالث

عمل برنامج لإدارة

السوبر ماركت

باستخدام

Visual Basic 6.0

- المرحلة الثانية -

في هذا الفصل :

- الربط مع قاعدة البيانات .
- تصميم شاشة حول البرنامج .
- تصميم شاشة تسجيل الدخول ومستخدم جديد .
- محتويات قائمة ملف .
- تصميم شاشة عرض البضائع .
- الانتظار في البرنامج .
- الحذف في البرنامج .
- التعديل في البرنامج .
- اضافة بضاعة جديدة - موردين - أنواع .
- قائمة مهام (عمليات البيع والشراء) .
- التحكم في عمليات البيع والشراء .
- قائمة مسح .
- قائمة أدوات .
- تصميم الآلة الحاسبة .
- النسخ الاحتياطي .
- البحث عن البضائع .
- تقارير البضائع .
- تقارير المصانع والأنواع .
- الإحصائيات والرسوم البيانية .
- التقارير اليومية والتقارير بين تاريخين .
- الداتا ريبورت والطباعة باستخدام DataReport

الربط مع قاعدة البيانات

في هذا الدرس وما يليه ، سوف نبدأ ببرمجة برنامج السوبر ماركت في هذا الدرس سوف نقوم بعمليات الربط مع قواعد البيانات .

كما سنضيف في هذا الدرس سوف نقوم باضافة المكتبات اللازمة للتعامل مع قاعدة البيانات .

لذلك سنضيف المكتبات الخاصة بقواعد البيانات Dao وهي الطريقة التي سننشأ بها هذا البرنامج بإذن الله ، أما Ado فسوف نتعلمها سوية في دروس إنشاء قاعدة بيانات على الإنترنت كما هو موضح في مقدمة الدورة الأولى .

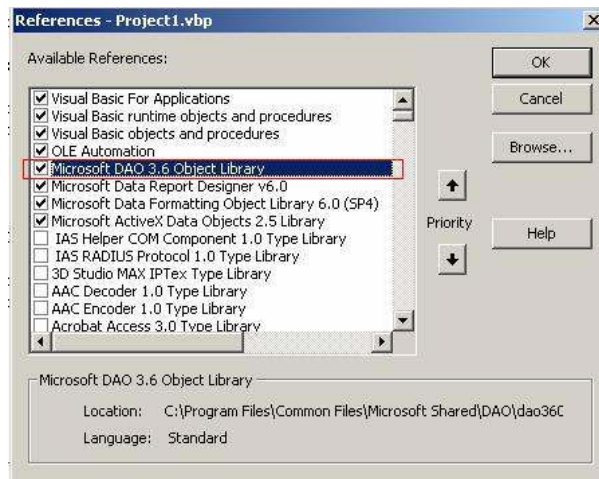
من قائمة Project اختر References ... ستظهر لك شاشة المكتبات المتوفرة ... لاحظ وجود أربع مكتبات افتراضية للفيجوال بيسك .

الآن ابحث عن المكتبة التالية :

Microsoft Dao 3.6 Object Library

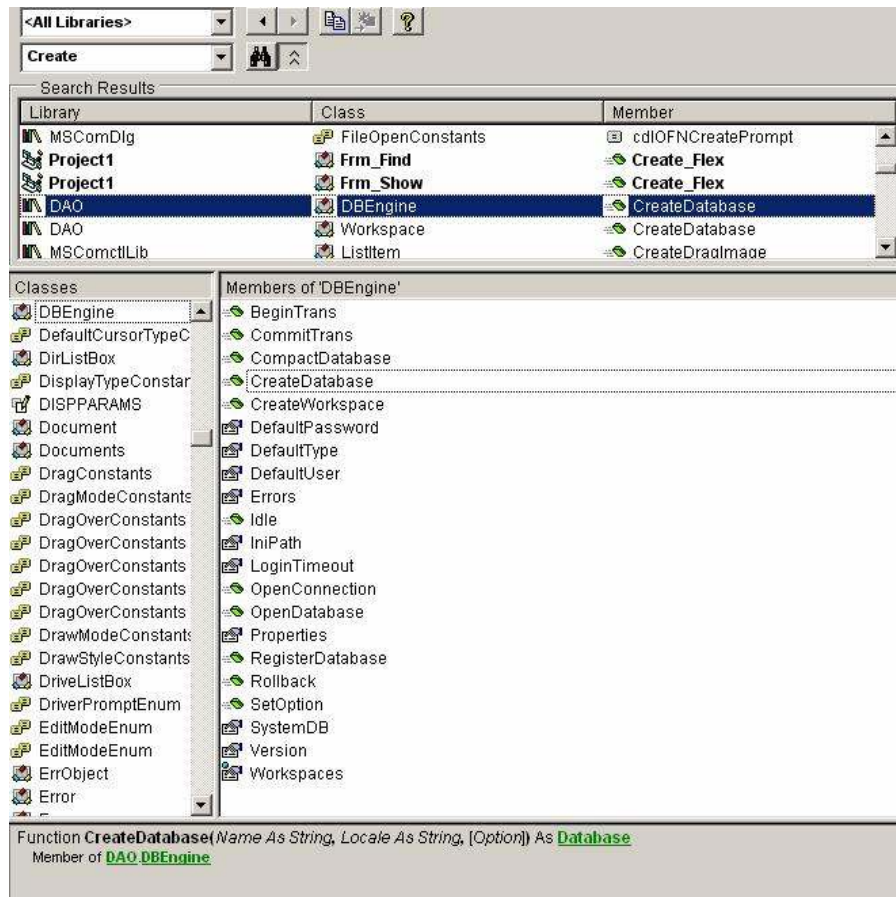
أو أي إصدار أحدث - أو أقدم في حالة عدم وجوده - .

كما في الشكل التالي :



والآن لكي تعرف الطرق Methods والخصائص Properties التي تحتويها هذه المكتبة - أو غيرها - قم بالضغط على زر F2 ومن القائمة العلوية اختر Dao وانظر إلى القائمتين في الأسفل .

تحتوي القائمة اليسرى على المفاتيح الرئيسية لهذه المكتبة ، فيما تحتوي القائمة اليمنى على المفاتيح الفرعية .



والآن لننتعرف على إحدى طرق الحصول على بعض المعلومات او (باراميترات Parameters) للدوال والإجراءات المختلفة ، وبفرض أنني أريد أن أعرف أمر انشاء قاعدة بيانات جديدة ، فكل ما عليك هو كتابة كلمة ذات دلالة في مربع البحث أعلاه وتكن مثلاً **Create** ستلاحظ ظهور قائمة من نتائج البحث ومنها مثلاً **CreateDatabase** والتي بمجرد النقر عليها سوف ترشدنا إلى الباراميترات المطلوبة لهذه الدالة . أيضاً يمكنك تدقيق نتائج البحث من خلال اختيار **DAO** من قائمة المكتبات أعلى مربع البحث . لاحقاً يمكنك البحث عن هذه الدالة فقط ومعرفة كل شيء عنها من مكتبة **MSDN** مثلاً ... بدل البحث في كل ما يتعلق بقواعد البيانات .

تكوين محتويات الموديول **Module** الرئيسية .

يعد الموديول أو الملف البرمجي وحدة برمجية عامة **Public** بحيث نستطيع الوصول إلى الدوال الموجودة فيها من أي مكان في البرنامج ، كما أننا يمكننا تعريف دالة **Main** بداخله بحيث تنفذ الأوامر الموجودة بداخلها قبل البدء في البرنامج .

لإنشاء Module اضغط على أي فورم بزر الماوس الأيمن ... اختر New ومن ثم Module .
ستظهر شاشة لكتابة الأكواد ... هذه هي كل محتويات Module .

عادة ما نبدأ في الموديول بإنشاء اجراء بهذا الشكل :

```
Public Sub Main()
```

```
End Sub
```

ومن ثم ومن قائمة Project - Project properties قم بالبحث عن StartUp Object والتي تحتوي على أول فورم سيتم البدء به ... اختر SubMain وبذلك تكون الدالة التي قمنا بإنشاءها هي الأولى في التنفيذ .



الآن سنعرف بعض المتغيرات أو الكائنات التي سنحتاج إليها ، فمثلاً نحتاج كائنين ليتعاملوا مع قاعدة البيانات ككل ... واحدة ثابتة والأخرى متغيرة حسب الحاجة خلال عمل البرنامج ويتم ذلك بالشكل التالي (خارج الدالة طبعاً) :

```
Public D1 As Database  
Public D2 As Database
```

كذلك نحتاج خمسة جداول ... لجدول البضائع ، المسوقين ، الأنواع ، واثنين آخرين لنتائج الاستعلام
وما شابه ذلك ، وذلك بالشكل التالي :

Public T1 As Recordset
Public T2 As Recordset
Public T3 As Recordset
Public T4 As Recordset
Public T5 As Recordset

بالامكان - بل المفترض - الاستغناء عن هذا الموضوع بتعريف المتغيرات فقط عند الحاجة إليها ،
واغلاق قاعدة البيانات فيما عدا ذلك ، كما يمكن ايضاً فتح قاعدة البيانات والجداول دائماً للتسهيل
في التعامل برغم أن هذا قد يسبب ثقلاً كبيراً على النظام ، ولكن هذه هي الطريقة التي قمت بها في
الدورة السابقة للعمل مع قواعد البيانات .

سنحتاج أيضاً لتعريف متغير عام بحيث نحتفظ فيه بمسار البرنامج لاستخدامه ضمن البرنامج ، وذلك
بالشكل التالي :

Public Folder

سنحتاج أيضاً إلى ثابت نضع فيه كلمة سر قاعدة البيانات وهي VB4ARAB بحيث نستخدمها
دائماً ، وعندما نقرر تغييرها نغيرها في هذا المكان فقط .

Public Const PassWord = "VB4ARAB"

وذلك لكي لا نقوم بالتعديل في جميع الكود حينما نحتاج إلى تغيير كلمة المرور - الاجمالية - لقواعد البيانات ، بل نكتفي بتغيير قيمة هذا الثابت فقط .

آخر ما نحتاج إليه هو متغير يحتوي على معلومات الجدول الأخير ، وفي هذه الحالة يمكننا عمل Array تحتوي على ٥ أجزاء يمثل كل جزء منها أحد المعلومات ... وذلك بالشكل التالي :

Public Info(5)

قم ايضاً بكتابة الأمر التالي لتحويل الرسائل إلى أسلوب عرض اللغة العربية ، وهو ما شرحناه في دروس سابقة :

```
Public Enum arabic
    arabic = vbMsgBoxRight + vbMsgBoxRtlReading
End Enum
```

الآن سنعود إلى الدالة التي قمنا بإنشاءها ونبدأ بالكتابة داخلها .
أول مهمة ينبغي أن يقوم بها البرنامج هو الحصول على المسار الحقيقي لقاعدة البيانات ، وذلك بالشكل التالي :

```
Folder = App.Path
If Right(Folder, 1) <> "\" Then Folder = Folder & "\"
```

الخطوة الثانية هي التأكد من وجود قاعدة البيانات ، فإن لم تكن موجودة فإننا نقوم باستدعاء وظيفة انشاءها ، وفي كلا الحالات سوف نتصل بها .

```
If Dir(Folder & "db1.mdb") = "" Then
MsgBox "لم يعثر البرنامج على قاعدة البيانات وسوف يقوم باضافة واحدة جديدة"،
vbExclamation ، "قاعدة بيانات غير موجودة"
'Here We will Create The Database
End If
```

يمكنك مراجعة اسلوب انشاء قاعدة بيانات على هذه السلسلة من الدروس ، على هذا الرابط :

<http://www.c4arab.com/showlesson.php?lesid=2332>

<http://www.c4arab.com/showlesson.php?lesid=2333>

<http://www.c4arab.com/showlesson.php?lesid=2334>

<http://www.c4arab.com/showlesson.php?lesid=2335>

وفي هذه الدورة سوف نكتفي بأن نظهر له رسالة للتواصل مع مبرمج البرنامج في حالة عدم وجود قاعدة البيانات ، لذلك سوف نحور الكود ليصبح بالشكل التالي :

```
If Dir(Folder & "db1.mdb") = "" Then
MsgBox "لم يعثر البرنامج على قاعدة البيانات الرجاء مراسلة المبرمج للحصول على نسخة
vbExclamation" ، "قاعدة بيانات غير موجودة" جديدة من قاعدة البيانات" ،
End
End If
```

بعد ذلك سوف نربط D1 بقاعدة البيانات بالشكل التالي :

```
Set D1 = DBEngine.Workspaces(0).OpenDatabase(Folder
& "Db1.mdb"
```

ولكن لأن لدينا PassWord فسوف نحتاج إلى تمرير بارميترات اضافية منها pwd والذي سوف يحتوي على كلمة المرور :

```
Set D1 = DBEngine.Workspaces(0).OpenDatabase(Folder  
& "Db1.mdb", False, False, ";pwd=VB4ARAB")
```

وبما أن لدينا ثابت يحتوي على كلمة السر ، لذا سوف يصبح الأمر بالشكل التالي :

```
Set D1 = DBEngine.Workspaces(0).OpenDatabase(Folder  
& "Db1.mdb", False, False, ";pwd=" & PassWord)
```

ومن ثم نربط T1,T2,T3 بالجدول الموجودة في D1 وذلك بالشكل التالي :

```
Set T1 = D1.OpenRecordset("Tb_Product", dbOpenTable)  
Set T2 = D1.OpenRecordset("Tb_Category",  
dbOpenTable)  
Set T3 = D1.OpenRecordset("Tb_Factory", dbOpenTable)
```

كما ذكرنا سابقاً يمكن الاعتماد على جمل الاستعلام أو فتح قاعدة البيانات حين الحاجة إليها فقط .
*** كما أننا سوف نستخدم الداول الخاصة بـ **Dao** مثل **AddNew** ، **Delete** ، وغير ذلك . كما يمكننا الاستفادة أيضاً من جمل لغة الاستعلامات التي تعلمناها سابقاً وهو ما سوف نفعله مع **Ado** في دروة تطوير المواقع باستخدام **Asp** إن شاء الله . وبنفس الطريقة يمكن عمل ذلك مع الفيچوال بيسك .

آخر خطوة هي اظهار الشاشة الإفتتاحية ، والتي سنتعرف على طريقة تكوينها في الدرس القادم ولكن هذا أمر اظهارها .

```
Frm_About.Show
```

وحتى ذلك الحين جرب هذا الجزء الصغير ... احذف قاعدة البيانات وراقب ما يحدث ، أو غير اسمها ... أيضاً جرب استعادتها ، وتغيير كلمة المرور .

ملف مرفق :

قم بمطالعة الملف prog1.zip في المرفقات لمزيد من التفاصيل .

تصميم شاشة حول البرنامج

عادة ما لا تحتوي شاشة **About** والتي تظهر في مقدمة البرنامج أي أزرار تحكم ولذا نستطيع تكوين هذه الشاشة من اختيار القيمة **None** من خاصية **BrderStyle** .
في هذه الحالة عادة ما يتم وضع **Shape** بمحيط الفورم بحيث يأخذ شكلاً جمالياً ، يمكن أيضاً جعل النافذة بارزة - وهذا ما سنقوم بعمله - ، ولهذا فإن علينا الإحتفاظ ببروز الفورم **BorderStyle = 3-Fixedialog** وذلك لمنع تغيير حجم الفورم مع اخفاء جميع أدوات التحكم وذلك بالشكل التالي :

* اضبط خاصية **ControlBox = False** .

* امسح جميع محتويات الخاصية **Caption** .

الخطوة الأخيرة هي جعل هذه الشاشة تظهر في المنتصف وذلك من خلال خاصية **StartUp** . **Position = Center Screen** .

طريقة أخرى قياسية - لكننا لن نستخدمها في هذا البرنامج وهي الضغط بزر الماوس الأيمن على الفورم ... **Add** ومن ثم **Form** ، واختر **Splash Screen** وسوف يظهر لك قالب بالشكل القياسي للشاشات الافتتاحية في ويندوز . كل ما عليك تغيير الصورة والمعلومات فقط ومن ثم استخدامها فوراً .

نعود الآن لشاشتنا ... ولكل منا أسلوب في تصميمها ، ولقد صممت الشاشة بطريقة بسيطة جداً بحيث وضعت صورة أخي مصطفى **Taftaf** من منتدى vb4arab فقط .

وسنبداً في التصميم البرمجي .

أول عنصر في التصميم البرمجي هو **وقت اختفاء هذه الشاشة** ، وعادة ما يتم اضافة **Timer** بزمان ٢٠٠٠ أو ٣٠٠٠ يتم بعده اخفاء هذه الشاشة واطهار الشاشة التالية .

هناك طريقة أخرى بعمل تأخير لأمر الاخفاء مدة معينة باستخدام **Sleep** مثلاً ، في البرامج الكبيرة لا يكون هناك تأخير ولكن في الواقع منذ ظهور هذه الشاشة يتم تحميل باقي عناصر البرنامج وبمجرد الانتهاء من هذه العملية يتم اخفاء الشاشة ، ولكن برنامجنا لن يأخذ وقتاً للتحميل أصلاً ولهذا أخرنا الشاشة بواسطة **Timer** .

والشاشة التالية في برنامجنا تنقسم إلى قسمين **إذا كان البرنامج يعمل لأول مرة فستظهر له شاشة يدخل فيها معلوماته ومعلومات شركته ، وإلا فإن شاشة ادخال اسم المستخدم وكلمة المرور سوف تظهر .**

ونستطيع عادة التعرف إن كانت هذه أول مرة أم لا بعدد من الطرق ، منها تسجيل قيمة في الـ **رجستري** مع تشغيل البرنامج والتأكد من وجودها كل مرة ، إلا أن هذا يعني أنه إذا تم عمل **Format** للجهاز فسوف يتم مسح هذه القيمة ولو شغلت البرنامج ولو على قاعدة البيانات القديمة فسوف يعتبر أنها المرة الأولى لتشغيل البرنامج .

من أجل ذلك سوف نعتمد في هذه الحالة على وجود سجل في جدول **Tb_User** أم لا ، ولذا فإننا نقوم بعمل ارتباط لهذا الجدول مع **T4** .

```
Set T4 = D1.OpenRecordset("Tb_User", dbOpenTable(
```

ومن ثم نتأكد من وجود سجل أم لا باستخدام خاصية **RecordCount** والتي تحتوي على عدد السجلات ، وعلى أساس ذلك يتم التوجه إلى فورم التسجيل أو فورم كلمة المرور - في جميع الحالات سوف تظهر الشاشة الحاضنة كخلفية لهذه الشاشات . من أجل الشكل ولكنهم ليسوا من الأبناء .

لاحظ أنه في حالة وجود سجل سوف يتم ملء المصفوفة التي أنشأناها سابقاً (5) info بمعلومات هذا الجدول .

وفي جميع الحالات سوف يتم تعطيل التايمر حتى لا يعمل لاحقاً ، وكذلك سوف نغلق الجدول ، وبذلك يكون الكود الكامل لهذا التايمر هو :

```
Private Sub Timer1_Timer()  
Set T4 = D1.OpenRecordset("Tb_User", dbOpenTable(  
  
MDIForm1.Show  
Unload Me  
  
If T4.RecordCount = 0 Then  
    T4.Close  
    Frm_NewUser.Show 1  
Else  
    Info(0) = T4!User_Name  
    Info(1) = T4!PassWord  
    Info(2) = T4!Name  
    Info(3) = T4!Telephone  
    Info(4) = T4!More  
    T4.Close  
    Frm_Login.Show 1  
End If  
End Sub
```

شاشة تسجيل الدخول والمستخدم

في هذه الشاشة سوف نلغي خاصيتي التكبير والتصغير من الفورم ، وذلك عن طريق ضبط خاصية BorderStyle للفورم بحيث تصبح = ١ .

أيضاً سيعد ضغط زر X بمثابة اغلاق البرنامج ، لذلك لا بد من كتابة ذلك في حدث Unload للفورم بالشكل التالي :

```
Private Sub Form_Unload(Cancel As Integer)
End
End Sub
```

والآن صمم الشاشة كما يحلو لك مع وجود خمسة مربعات ادخال للاسم - كلمة المرور - اسم المحل - التلفون - العنوان ، وزرين لالغاء الامر وللاستمرار .
لا تنس أن تعدل خاصية Caption للفورم بحيث تصبح (تسجيل مستخدم جديد) أو ما شابه لن يكون سيئاً أن تكتب التاريخ أيضاً .
كتابة التاريخ في خاصية Caption للفورم اكتب الأمر التالي في حدث Form_Load :

```
Me.Caption & " = DateTime.Date" تسجيل مستخدم جديد
```

كذلك اضبط خاصية RightToLeft بحيث تساوي True ، وذلك كما أسلفنا عند الحديث عن تصميم البرمجيات العربية .

ملاحظات سريعة :

* لا تنس أنه لكي تنشأ مربع نص خاص بكتابة كلمة السر اضبط خاصية PasswordChar إلى الحرف الذي تريد له أن يظهر بدل كلمة السر - غالباً * - .
* لا تنس أيضاً أنه لكي تنشأ حقل لكتابة ملاحظات (حقل More بالنسبة إلينا) فلا بد من ضبط خاصية MultiLine بحيث تصبح True وذلك لنستطيع الكتابة على أكثر من سطر .
* أيضاً يمكنك التحكم في ظهور أشرطة التمرير من عدمها من خلال خاصية ScrollBars

نعود الآن لحدث التحميل لهذا الفورم ، فإضافة لتغيير خاصية Caption للفورم وقت التنفيذ ، فإمكاننا أيضاً وضع اسم مستخدم الجهاز (افتراضياً) في خانة (اسم المستخدم) .
نقوم بذلك بالشكل التالي :

أولاً . استدعي دالة `GetComputerNameA` من مكتبة `kernel32` في الموديول بالشكل التالي :

```
Public Declare Function GetComputerName Lib "kernel32"
Alias "GetComputerNameA" (ByVal lpBuffer As String,
nSize As Long) As Long
```

أو تستطيع تعريفها في الفورم بالشكل التالي :

```
Private Declare Function GetComputerName Lib
"kernel32" Alias "GetComputerNameA" (ByVal lpBuffer As
String, nSize As Long) As Long
```

لكننا عرفناها في الموديول لأننا قد نحتاج إليها لاحقاً في أماكن أخرى .
وهذا الدالة أحد دوال `API` في الويندوز والتي تعطينا اسم المستخدم المسجل في جهاز الكمبيوتر .

ثانياً : الآن ضع الأمر التالي في حدث `Form_Load` :

```
Dim A As String * 256
Dim X
X = GetComputerName(A, 256)
Text1.Text = Left(A, InStr(A, Chr(0)) - 1)
```

نعود مرة أخرى ، وكما ذكرنا أن ضغط زر (الغاء الأمر) يعني الخروج من البرنامج ، لذلك سنكتب `End` في الحدث الخاص بها .

لكن ماذا عن ضغط زر موافق ؟ سنقوم بمهمتين :

* المهمة الأولى التأكد من المدخلات .

* المهمة الثانية هي اضافة المدخلات في حالة كانت صحيحة .

وبما أن الغرض من هذه الدروس التعليم ، وبناء على طلب الإخوة من أعضاء vb4arab سوف نقوم بعمل دالة نمرر لها قيم الحقول الخمسة ، وتعيد -١ في حالة وجود أي خطأ - لتعلم أسلوب الدوال في فيجوال بيسك - .

لن أتوسع في شرح أنواع الارسال - وفي حالة وجود وقت سوف أشرح الامر في درس منفصل ولكن لاحقاً - لكننا سنكتب صيغة هذه الدالة :

* لاحظ أنها دالة Function (لانها تعيد قيمة) وليست اجراء Sub أو Prcedure .

```
Private Function Check(A, B, C, D, E,
```

```
End Function
```

حيث أن كل حرف يمثل أحد قيم مربعات النص .

المهمة الأولى هي التأكد من وجود الاسم ، كلمة المرور ، كون كلمة المرور بين ٣ و ١٢ حرف . وفي حالة وجود أي خطأ سوف أعيد القيمة - ١ .

ونعيد القيمة من الدالة بمعنى أن نجعل اسم الدالة = القيمة التي نرغب باعادتها .

سنكتب الكود للمهمة الأولى والذي سيكون بالشكل التالي :

```
If A = "" Then
```

```
MsgBox vbExclamation "لا بد من ادخال اسم المستخدم" ,
```

```
جديد"
```

```
Check = -1
```

```
Exit Function
```

```
Elseif B = "" Then
```

```
MsgBox vbExclamation "تسجيل مستخدم جديد"
```

```
Check = -1
```

```
Exit Function
```

```
Elseif Len(B) < 3 Or Len(B) > 12 Then
```

vbExclamation "تسجيل" لا بد أن تكون كلمة المرور بين ٣ و ١٢ حرف" ، MsgBox

مستخدم جديد"

Check = -1

Exit Function

Else

Check = 1

End If

المهمة الثانية هي وضع قيم افتراضية للحقول المتبقية في حالة لم يقوم المستخدم بادخالها ، وذلك بالشكل التالي :

If C = "" Then

Text3.Text = "برنامج البائع الذكي"

End If

If D = "" Then

Text4.Text = "0"

End If

If E = "" Then

Text5.Text = "البرنامج من تصميم منتدى فيجوال بيسك للعرب"

& "بامكانك زيارة الموقع على

www.vb4arab.com/vb" هذا الرابط

& "أو مراسلة صاحب

Hammada2091@Hotmail.com" البرنامج على هذه البريد الإلكتروني

End If

※ تذكر أن VbNewLine مع TextBox أو Chr(13) عموماً تنقلنا لسطر جديد ، حيث أن رقم ١٣ يقابل زر Enter في جدول آسكي Ascii.

نعود الآن إلى تصميم زر موافق ، وسنبداً باستدعاء الدالة ووضع القيمة التي ترجعها في متغير :

```
Dim Y As Integer  
Y = Check(Text1.Text, Text2.Text, Text3.Text, Text4.Text,  
Text5.Text)
```

إذا وجدنا أن قيمة Y هي -١ فإننا نخرج ، وإلا فإننا سوف نضيف المستخدم وذلك باستدعاء (الإجراء هذه المرة) Add_User .

```
If Y = -1 Then Exit Sub  
Add_User
```

والآن سوف نقوم ببرمجة الإجراء Add_User والذي سيحتوي على فتح جدول المستخدمين في البداية بالشكل التالي :

```
Set T4 = D1.OpenRecordset("Tb_User", dbOpenTable)
```

ثم نبدأ بإضافة سجل جديد بالشكل التالي :

```
T4.AddNew
```

ومن ثم نبدأ بإسناد القيم المختلفة للحقول بالشكل التالي :

```
T4!User_Name = Text1.Text  
T4!PassWord = Text2.Text  
T4!Name = Text3.Text  
T4!Telephone = Text4.Text  
T4!More = Text5.Text
```

ومن ثم نقوم بحفظ السجل .

T4.Update

ثم نقوم بنفس العمليات التي تحدث عند وجود قاعدة بيانات ، فنضع القيم المختلفة للقاعدة في المصفوفة ونغلق قاعدة البيانات .

```
Info(0) = T4!User_Name  
Info(1) = T4!PassWord  
Info(2) = T4!Name  
Info(3) = T4!Telephone  
Info(4) = T4!More
```

T4.Close

لا تنس بعد ذلك أن تذكره بكلمة المرور التي أدخلها كيلا ينساها ، اظهر له رسالة بها :

MsgBox & " Info(1), vbInformation +
arabic" ، "تذكير بكلمة المرور"

أخيراً نقوم بإنشاء دالة في Module اسمها Start وتحتوي على الأوامر الرئيسية التي تحدث عند بداية تشغيل البرنامج ، وهو ما سوف نقوم بعمله لاحقاً .

Public Sub Start()

End Sub

ونستدعيها في دالة Add_New بعد الشاشة الحالية .

Me.Hide
Call Start

ولن ننسى بالطبع جعل بعض الحقول لا تقبل إلا أرقاماً فقط ، مثل رقم الهاتف بالشكل التالي :

```
Private Sub Text4_KeyPress(KeyAscii As Integer)
If KeyAscii < 48 Or KeyAscii > 59 Then
    Beep
    KeyAscii = 0
End If
End Sub
```

ويمكنك تطوير الأمر السابق كما أشرنا إلى ذلك في دروس سابقة .

تصميم شاشة تسجيل الدخول .

كما هو الحال مع الشاشة الافتتاحية ، فإن الفيجوال بيسك يوفر لنا شاشة قياسية لعملية تسجيل الدخول ... تستطيع تجربتها من خلال اختيار **New** ومن ثم **Log in Form** . ولكننا سوف نصمم يدوياً شاشة تسجيل الدخول .

ينبغي أن يتوفر في هذه الشاشة زرین (موافق - الغاء الامر) ومربعي نص يحتوي أحدهما على اسم المستخدم ، وكلمة المرور . وكما قمنا بتصميم الشاشة الماضية ، فسوف نجعل هذه الشاشة بدون زري التصغير والتكبير ، كما أن ضغط زر **X** سوف ينهي البرنامج .

والآن لنضع سوية أوامر الدخول :

المهمة الأولى هي التأكد من اسم المستخدم **Info(0)** وكلمة المرور **Info(1)** ، وذلك بالشكل التالي

:

*** بإمكاننا عند هذه العملية فتح قاعدة البيانات لقراءة القيم منها ومن ثم اغلاقها ثانية بدلاً من تخزينها في متغيرات .

```
If Text1.Text <> Info(0) Then
    MsgBox "لقد ادخلت اسم المستخدم خطأ ... فضلاً أعد المحاولة"،
    vbExclamation + arabic"تسجيل مستخدم جديد" ،
    Exit Sub
End If

If Text2.Text <> Info(1) Then
    MsgBox "كلمة المرور غير صحيحة ... فضلاً أعد المحاولة"،
    vbExclamation + arabic"تسجيل مستخدم جديد" ،
    Exit Sub
End If
```

الخطوة الثانية ، وهي أننا نريد أن نحدد (كلمة المرور أو اسم المستخدم) في حالة الخطأ .
وذلك باستخدام الخاصيتين SelStart و SelLength اضافة للدالة Len .
تحدد الخاصية SetStart بداية التحديد ، و SelLength نهاية التحديد والذي ينبغي أن
يكون طول النص الموجود Len وذلك بالشكل التالي :

```
Text1.SelStart = 0
Text1.SelLength = Len(Text1.Text)
```

ونفس الأمر بالنسبة لكلمة المرور .

في حالة كان كل من اسم المستخدم وكلمة المرور صحيحين فإننا نخفي هذه الشاشة ... ونستدعي الدالة Start التي انشأناها في الدرس السابق .

وبذلك يصبح الكود في النهاية :

```
If Text1.Text <> Info(0) Then
    MsgBox "لقد ادخلت اسم المستخدم خطأ ... فضلاً أعد المحاولة"،
vbExclamation + arabic"تسجيل مستخدم جديد" ،
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1.Text)
    Exit Sub
End If

If Text2.Text <> Info(1) Then
    MsgBox "كلمة المرور غير صحيحة ... فضلاً أعد المحاولة"،
vbExclamation + arabic"تسجيل مستخدم جديد" ،
    Text2.SelStart = 0
    Text2.SelLength = Len(Text2.Text)
    Exit Sub
End If

Me.Hide
Call Start
```

عمل عداد لتسجيل الدخول .

من الطبيعي دائماً ألا يترك الشخص ليدخل كلمات مرور كما يحلو له ، بل العادة أن تترك له ثلاث محاولات فقط قبل انتهاء البرنامج .

لذلك سوف نضيف متغير ما وليكن اسمه My_Count في قسم التصريحات Genral ونضع قيمته في حدث Form_Load بحيث يصبح ٠ .

ومع كل ضغطة على زر تسجيل الدخول تتم زيادة هذا المتغير بمقدار ١ ، والتأكد من أنه لم يصل إلى ٤ ، وفي هذه الحالة يتم الخروج مباشرة .

بذلك يصبح كود تسجيل الدخول بالشكل التالي :

```
My_Count = My_Count + 1

If LCase(Text1.Text) <> LCase(Info(0)) Then
    MsgBox "لقد ادخلت اسم المستخدم خطأ ... فضلاً أعد المحاولة"،
vbExclamation + arabic"تسجيل مستخدم جديد" ،
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1.Text)
    If My_Count = 4 Then End
    Exit Sub
End If

If Text2.Text <> Info(1) Then
    MsgBox "كلمة المرور غير صحيحة ... فضلاً أعد المحاولة"،
vbExclamation + arabic"تسجيل مستخدم جديد" ،
    Text2.SelStart = 0
    Text2.SelLength = Len(Text2.Text)
    If My_Count = 4 Then End
    Exit Sub
End If

Me.Hide
Call Start
```

ما هي العمليات التي يفترض أن تحدث عند بدء البرنامج . وهي التي سنضيفها في أمر Start سوف يكون هناك العديد من الأشياء التي سنضيفها لاحقاً ، ومن أجل الحفاظ على تسلسل الشرح سوف نضيف ما نحتاج إليه عند الحاجة إليه فقط .

* اظهار اسم البرنامج + اسم المحل في عنوان الشاشة الرئيسية .

* اظهار معلومات المحل في منتصف الشاشة الرئيسية .

بالنسبة للمهمة الأولى فسيكون الأمر سهلاً جداً ... كل ما عليك هو كتابة الأمر التالي في Start :

MDIForm1.Caption & "برنامج البائع الذكي - " = Info(2(

أما الجزء الثاني فهو سهل إن شاء الله ، ولكن ثم مشكلة متعلقة ب MDIForm .
لو لاحظت أنه لا يمكن وضع أي أدوات على الفورم مثل Lable وغيره مما نريد أن تظهر فيها
معلومات المحل ، ولا يمكن عرض إلا أداة مثل PictureBox ، ولو قمنا بوضعه على كامل
الشاشة فإن جميع الشاشات الأبناء للشاشة الرئيسية سوف تظهر أسفل منها ، وبهذا لا يمكننا رؤيتها
، لكننا سوف نتحايل على ذلك بأن ننشأ فورماً خاصاً Frm_Info ونعرضه بصورة دائمة في
منتصف الشاشة .

سنبدأ فوراً بتصميم هذا الفورم تأكد من أنك تضع Label1 لاسم المحل ، Label2 لرقم الهاتف
Label3 لعرض المعلومات الأخرى ، صمم الشكل العام كما يحلو لك .
وتذكر أن خاصية BorderStyle لهذا الفورم لا بد أن تكون ٠ وبنفس اللون ، حتى لا يظهر
مختلفاً عن خلفية الشاشة الرئيسية .
أيضاً لا تنس أن تضع خاصية MDIChild له True .

الآن وفي دالة Start سوف نحدث محتويات هذا الفورم اضافة إلى اظهاره :

Frm_Info.Label1 = Info(2(

Frm_Info.Label3 = Info(3(

Frm_Info.Label3 = Info(4(

Frm_Info.Show

ملف مرفق :

قم بمطالعة الملف prog4.zip في المرفقات لمزيد من التفاصيل .

محتويات قائمة ملف

والآن وخلال هذا الدرس سوف نصمم محتويات القائمة (ملف) من الأسفل إلى الأعلى (وسنصمم :
انهاء - تعليمات - حول البرنامج) فيما نؤجل عرض البضائع إلى الدرس القادم .

لا تنس في بداية هذه الدروس أن جميع القوائم تحت قائمة رئيسية واحدة لها نفس الاسم ، ونقوم
بالتفريق بينها بواسطة الخاصية Index وذلك بالشكل التالي :

Select Case Index

Case 0

Case 2

Case 3

Case 5

End Select

عمل الأمر الموجود في انهاء .

الأمر بسيط جداً ، فيما مضى كنا نضع أمر End مباشرة للانتهاء ، ولكننا نشاهد في كثير من البرامج
رسالة تظهر قبل انتهاء البرنامج .

هل تريد بالتأكيد انتهاء البرنامج ؟

لعمل هذه العملية كل ما نحتاج إليه هو اظهار رسالة يكون جوابها YesNo ونضع ناتجها في متغير للتأكد من أنه لا يساوي No قبل الانهاء .
لنرى الآن هذا الكود :

```
Dim Sure
Sure = MsgBox vbExclamation + ("هل تريد بالتأكيد انهاء البرنامج ؟") ،
vbYesNo + arabic("تأكيد الخروج") ،
If Sure = vbYes Then End
```

الأمر بسيط جداً ، ولكن تبقى شيء آخر ، وهو أننا نريد لهذه الرسالة أن تظهر حين يتم الضغط على زر X في الفورم ، وهو حدث Unload للفورم .

جرب نسخ الأمر السابق كما هو في حدث Unload ستلاحظ أنك حتى ولو ضغطت اخترت الغاء الأمر فلن يفيد ذلك ، وللتغلب على هذه المشكلة نغير قيمة Cancel إلى True قبل الأمر السابق فكيف يتم ذلك ؟

إن حدث Unload يستقبل قيمة في المتغير Cancel توضح إذا ما تم الغاء الأمر أم لا ، ويحتوي افتراضياً على False عند الضغط على زر X ولذا فإننا نغير قيمته ، وذلك بالشكل التالي :

```
Cancel = True
Dim Sure
Sure = MsgBox vbExclamation + ("هل تريد بالتأكيد انهاء البرنامج ؟") ،
vbYesNo + arabic("تأكيد الخروج") ،
If Sure = vbYes Then End
```

جرب ذلك الآن ، ستجد أن حدث Load يحدث من جديد عندما يحدث Cancel=True مما يؤدي إلى اخفاء الشاشة التي قمنا باظهارها ، وازالة عنوان الفورم ... وللتغلب على هذه المشكلة يتم استدعاء الأمر Start ثانية بعد هذا الحدث ، وبذلك يصبح الكود الكامل بالشكل التالي :

```
Cancel = True
```

```
Dim Sure
```

```
Sure = MsgBox (vbExclamation +  
vbYesNo + arabic("تأكيد الخروج") ،
```

```
If Sure = vbYes Then End
```

```
Start
```

عمل شاشة (حول البرنامج ، وشاشة تعليمات) .

لعمل الشاشة الأولى سوف نستخدم إحدى شاشات الفيجوال بيسك القياسية وهي شاشة About تستطيع اضافة واحدة بالضغط على أي فورم بزر الماوس الأيمن واختيار About Dialog ... غير اسمه إلى AboutMe لكي نتلافى التشابه بينه وبين الفورم الافتتاحي الذي كان ينبغي أن يسمى Frm_Splash - خطأ مني .

الآن صمم شكل الفورم كما يحلو لك ، ولا تنس أن تضع أيقونة البرنامج مكان أيقونة الفيجوال بيسك . لاحظ أمراً ما في حدث Load لذلك الفورم وهو موجود افتراضياً :

```
Me.Caption = "About " & App.Title  
IbVersion.Caption = "Version " & App.Major & "." &  
App.Minor & "." & App.Revision  
IbTitle.Caption = App.Title
```

يقوم هذا الكود بالتعرف على خصائص المشروع من اصدار واسم وغيره لوضعها في اماكنها المناسبة ... تستطيع الاعتماد على هذه الطريقة مع التأكد من ضبط هذه الخصائص بصورة صحيحة من قائمة

Project ثم Project Property ومن خلال التبويب Make ... أو تستطيع الاعتماد على عملها وقت التصميم وحذف هذه الأكواد .

وفي حدث النقر على هذه القائمة سوف نضيف أمر اظهار هذه الشاشة فقط :

AboutMe.Show 1

وماذا عن الشاشة الثانية ؟

من المعتاد دائماً ارفاق ملف التعليمات مع البرنامج بصيغة rtf وهي صيغة مطورة من Text لكنها تسمح بالألوان ويمثلها في برامج ويندوز برنامج الدفتر – Wordpad ... وبما أن الغالب هو اضافة نفس التعليمات في شاشة التعليمات ، فإننا نحتاج إلى أداة تدعم اختلاف الألوان ، وهذه الأداة هي أداة RichText .

سنبدأ أولاً بعمل فورم فارغ من كل شيء .. ولن يحتوي شريطه العلوي سوى على زر X فقط ... ونضيف هذه الاداة فيه .

لاضافة الأداة اضغط Ctrl+T ومن ثم اختر Microsoft RichText Control 6.0 .

ولنقم الآن بفتح الدفتر – Wordpad ولنضع بعض الكلام كتجربة ، ونحفظه في نفس مسار البرنامج تحت اسم Help.rtf .

تحتوي هذه الاداة على خاصية FileName والتي تحدد مكان الملف الذي سوف يظهر في هذه الاداة والتي سوف نضبطها وقت التصميم .. أيضاً خاصية ScrollBars والتي تحدد ظهور أشرطة التمرير من عدمه .

كل ما علينا أن نحدد مصدر الأداة في حدث التحميل للفورم ، وذلك بالشكل التالي :

RichTextBox1.FileName = Folder & "help.rtf"

والأمر الخاص باظهار هذا الفورم :

Frm_Help.Show 1

ملف مرفق :

قم بمطالعة الملف prog5.zip في المرفقات لمزيد من التفاصيل .

تصميم شاشة عرض البضائع

في هذا الدرس سوف نبدأ بعمل شاشة عرض البضائع ، وكما هي العادة عند التصميم فإن كل منا يصمم كما يحلو له ... لكن لا تنس أننا لا بد أن نشترك في المحتويات الرئيسية وهي :

قائمة في الأعلى لاختيار البضائع .

جدول لعرض جميع البضائع الموجودة .

مربعات نص لعرض البضاعة المختارة حالياً .

أزرار أوامر للتعديل والحذف ، وزرين لعملية بيع أو شراء تتصل بالشاشة الخاصة بها والتي سنقوم بعملها عندما يأتي دورها .

وأخيراً نتعين كأحد أبناء Children للفورم MDIForm ... وذلك عن طريق ضبط خاصية MDIChild = True .

لاضافة الجدول اضغط Ctrl+T واختر Microsoft Flex Grid 6.0 واخترنا هذا الجدول
لامكانية التحكم المطلق به ، حيث نقوم بعمل كل شيء فيه ، وحتى لو أردنا عمل DBGrid
متطور باستخدامه فسيمكننا ذلك .

لاظهار الشاشة من خلال القائمة نكتب الأمر التالي :

Case 0

If T1.RecordCount = 0 Then

MsgBox "لا توجد أي بضائع لعرضها ، لاضافة بضاعة جديدة اختر اضافة - اضافة vbExclamation + arabic" عرض البضائع ،

Exit Sub

End If

Frm_Show.Show

لو ظل الفورم على تصميمه القديم من حيث **BorderStyle=2** فستلاحظ تغير حجم الشاشة - لا أفهم لماذا - ولكن لو ضبطت الخاصية إلى ١ فلن تحدث هذه المشكلة .

لن ننسى اضافة السطرين التاليين ، حيث يقومون بتوسيط الفورم كما فعلنا مع Frm_Info ، لذلك سنضيفهما في حدث Load للفورم :

```
Frm_Show.Left = ((MDIForm1.Width - Frm_Show.Width) / 2 )
```

```
Frm_Show.Top = ((MDIForm1.Height - Frm_Show.Height) / 2) - 800
```

لنبدأ الآن سوية في تصميم اجراء Refresh_Me والذي ينطلق عند تشغيل الفورم

Form_Load ، وعند كل عملية حذف أو تعديل .

بما أن لدينا كائن T1 يحتوي على معلومات البضائع ، فلم لا نقوم بالدوران على هذا الكائن من

البداية إلى النهاية ووضع الاسماء فيه إلى القائمة :

```
Sub Refresh_Me()
```

```
Combo1.Clear
```

```
If T1.RecordCount <> 0 Then
```

```
T1.MoveFirst
```

```
For i = 0 To T1.RecordCount - 1
```

```
Combo1.AddItem T1!Name
```

```
T1.MoveNext
```

```
Next i
```

```
End If
```

هذا ولكي تعرض البضائع في هذه القائمة مرتبة أبجدياً فإننا نضبط خاصية **Sorted** للقائمة لتساوي **. True**

والآن فإننا وبنفس طريقة عرض البيانات في القائمة نريد لكامل البيانات أن تعرض في الجدول ، ولذا سوف نضعهم جميعاً داخل حلقة تكرار واحدة تقليلاً للجهد على الذاكرة . ولكنني أريد أن أشرح كيفية اضافة بيانات في **MSFlexGrid** .

ربما تجد بعض ما يفيدك عن هذه الاداة على هذا الرابط من الفريق العربي للبرمجة - الاستاذ مهند عبادي :

<http://www.arabteam2000-forum.com/index.php?showtopic=40959&hl=MSFlexGrid>

أيضاً درس الأستاذ حمد على الموسوعة العربية .

<http://www.c4arab.com/showlesson.php?lesid=215>

وإجمالاً: توجد خاصيتين على قدر كبير من الأهمية هما **Col** و **Row** واللذان تحددان في نقطة التقاءهما خلية **Cell** لها خصائص أشهرها **Text** لعرض النص فيها .

هناك أيضاً خاصيتي **Rows** و **Cols** واللذان تحددان عدد الصفوف والأعمدة في الجدول ، ووظيفة **Clear** والتي تقوم بمسح كافة محتويات الجدول .

سوف نقوم بعمل اجراء نسمية **Create_Flex** وذلك لتهيئة الجدول للتعامل معه ، يحتوي هذا الاجراء على وضع عدد الأعمدة = ٧ ، الصفوف = عدد السجلات + ١ ، ونضيف التعبيرات المناسبة على رأس كل جدول .
لعمل المهمتين الأولتين :

```
MSFlexGrid1.Clear  
MSFlexGrid1.Cols = 7  
MSFlexGrid1.Rows = T1.RecordCount + 1
```

أما للمهمة الثانية فسوف نتنقل بين الأعمدة ، ونضع على رأس كل عمود القيمة المناسبة له :

```
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 0
MSFlexGrid1.Text"ت" =
MSFlexGrid1.Col = 1
MSFlexGrid1.Text"رقم" =
MSFlexGrid1.Col = 2
MSFlexGrid1.Text" اسم البضاعة" =
MSFlexGrid1.Col = 3
MSFlexGrid1.Text"النوع" =
MSFlexGrid1.Col = 4
MSFlexGrid1.Text"المصنع" =
MSFlexGrid1.Col = 5
MSFlexGrid1.Text"السعر" =
MSFlexGrid1.Col = 6
MSFlexGrid1.Text"العدد" =
MSFlexGrid1.Col = 7
MSFlexGrid1.Text"وحدة/صندوق" =
```

ولكن ماذا عن توسيط الأعمدة ... نستطيع ذلك من خلال كود كالتالي :

```
MSFlexGrid1.ColAlignment(0) = 5
MSFlexGrid1.ColAlignment(1) = 3
MSFlexGrid1.ColAlignment(2) = 5
MSFlexGrid1.ColAlignment(3) = 5
MSFlexGrid1.ColAlignment(4) = 5
MSFlexGrid1.ColAlignment(5) = 5
MSFlexGrid1.ColAlignment(6) = 5
```

حيث أن ٥ ترمز للمنتصف ، أما ٣ فترمز إلى المحاذاة إلى اليمين .
نفس الأمر بالنسبة لعرض كل عمود ولن نغير سوى في عرض الأعمدة التي لا يناسبنا حجمها الافتراضي
وهي الأول (تسلسل) والثاني (اسم البضاعة) والثالث والرابع (النوع - المصنع) :

```
MSFlexGrid1.ColWidth(0) = 500  
MSFlexGrid1.ColWidth(1) = 1500  
MSFlexGrid1.ColWidth(2) = 1500  
MSFlexGrid1.ColWidth(3) = 1500
```

وسوف نستدعي هذا الإجراء في بداية Refresh_Me .
ولكن ماذا عن تعبئة الجدول داخل حلقة التكرار ، في الواقع بنفس الطريقة التي كتبنا بها رؤوس
الأعمدة سوف نكتب القيم الأخرى ، إلا أن رقم الصف سوف يصبح قيمة المتغير i+1 .
ومن ذلك نستطيع أن نحصل على كود اجمالي لدالة Refresh_Me - مؤقتاً - .

```
Sub Refresh_Me()  
Combo1.Clear  
Create_Flex  
If T1.RecordCount <> 0 Then  
T1.MoveFirst  
For i = 0 To T1.RecordCount - 1  
MSFlexGrid1.Row = i + 1  
MSFlexGrid1.Col = 0  
MSFlexGrid1.Text = T1!Number  
MSFlexGrid1.Col = 1  
MSFlexGrid1.Text = T1!Name  
MSFlexGrid1.Col = 2  
MSFlexGrid1.Text = T1!Category  
MSFlexGrid1.Col = 3  
MSFlexGrid1.Text = T1!Factory  
MSFlexGrid1.Col = 4  
MSFlexGrid1.Text = T1!price  
MSFlexGrid1.Col = 5
```

```

MSFlexGrid1.Text = T1.Fields("Count")
MSFlexGrid1.Col = 6
MSFlexGrid1.Text = T1!Box_Count

Combo1.AddItem T1!Name
T1.MoveNext
Next i
End If
End Sub

```

لاحظ أنني استخدمت طريقة ثانية في القراءة مع الحقل **Count** ذلك أنه ينبغي أن يوضع بين علامتي تنصيص لأنها كلمة محجوزة في الفيجوال بيسك .

ترتيب البضائع في الجدول .

بالتأكيد قد مر بك هذان الأمران خلال الدروس سابقاً :

```

MSFlexGrid1.ColAlignment(0) = 5

MSFlexGrid1.Col = 6
MSFlexGrid1.Text = T1!Box_Count

```

ذكرت بهذين الأمرين فقط لتدرك أننا حينما نريد تطبيق جميع خصائص التنسيق على هذه الأداة فإننا نستخدم أحد طريقتين - حسب الخاصية فإذا إن نذكر اسم الأداة **MSFlexGrid1** ثم الخاصية وأخيراً نضع رقم هذا العمود بين قوسين ، أو نذكر رقم العمود أو الصف - أو كليهما - ثم نطبق عليه هذه الخاصية وهو مثل خاصية **Sort** والتي تقوم بالترتيب ، لذا ف لترتيب الحقول حسب عمود الاسم فإننا نكتب الكود التالي :

```

MSFlexGrid1.Col = 1
MSFlexGrid1.Sort = 7

```

ومعنى الرقم ٧ واضح لمن اطلع على الدرس على الرابط السابق ...

ولكن ربما تكون قد لاحظت ان الأرقام لم تعد مرتبة ، هل هذا صحيح ؟

حل هذه المشكلة أن نضع عدداً جديداً كلياً وصفاً جديداً آخراً يحتوي على التسلسل غير الرقم ، فلنرى كيف نقوم بذلك .

أولاً قم بزيادة عدد الأعمدة بمقدار عمود واحد ... أيضاً قم بتحويل رقم كل عمود إلى الرقم + ١ .

قم بذلك وشاهد النتيجة ، ستجد ظهور صف فارغ على اليمين من الجدول .

سنغير الآن من اسم عمود الرقم ليصبح (الرقم) فيما يصبح العمود الجديد (ت) .

لاحظ أننا لن نقوم خلال الدوران بعملية تعبئة هذا العمود بالأرقام ، بل إن ذلك سوف يتم بعد الانتهاء من ترتيبها ، وإلا فلن نستفيد شيئاً - جرب وضعهما سوية وراقب ما يحدث - .
ولذلك نضع الكود التالي بعد الترتيب :

```
MSFlexGrid1.Col = 0
For i = 0 To T1.RecordCount - 1
    MSFlexGrid1.Row = i + 1
    MSFlexGrid1.Text = i + 1
Next i
```

وضع أسماء الموردين والأنواع بدلاً من أرقامها .

لو لاحظت أننا نقوم بوضع الأرقام لأن هذا هو المخزن في قاعدة البيانات أو تحديداً في الجدول الذي نقوم بالقراءة منه .

ولذا فإننا نحتاج إلى البحث في جدولي الأنواع والموردين قبل وضع الناتج وذلك في الأسطر التالية من الكود :

```
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = T1!Category
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = T1!Factory
```

وسنرى الآن كيف نقوم بهذه المهمة .
هل تتذكر كيف قمنا بفتح قاعدة T1 في البداية :

```
Set T1 = D1.OpenRecordset("Tb_Product", dbOpenTable(
```

لكننا لا نحتاج إلى فتحها بهذه الطريقة ، بل إننا نريد أن نقوم بفتح قاعدة بجملة استعلام للعلاقات التي بين الجداول ، لهذا الغرض سوف نستخدم T4 وذلك بالشكل التالي :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number"  
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset(
```

تقوم جملة الاستعلام هذه باستخراج جميع المعلومات من الجداول الثلاثة مع دمج كل معلومات المصانع والأنواع جنباً إلى جنب مع معلومات البضائع ، ولتتكون لديك نظرة أكثر تركيزاً قم بتجربة هذه الجملة في Access .

وإذا كنت تود معرفة المزيد عن مثل هذه الجمل ، وكيف تمت كتابتها ، فبإمكانك مراجعة الجزء الأول من هذا الكتاب .

وسيوضع هذا الأمر في حدث Refresh_Me .

الآن لا تنس أن تقوم بتحويل كل T1 إلى T4 في جميع هذه النافذة .
والآن لنر أسماء لحقول مثل Number و Name فإننا عندما نضعها فأيهما نقصد ، هل تلك التي في جدول البضائع أم التي في جدول المصانع أم الأنواع ، لذا فإننا لا بد أن نذكر مكان الحقل قبل اسمه ، ومن أجل ذلك أيضاً فإننا نغير أسلوب العرض من T1!Name إلى

```
T1.Fields("Tb_Product.name(
```


وكذا الأمر بالنسبة لحقل Number .

الهدف الأول من هذه العملية هو عرض اسم المصنع والنوع بدلاً من رقمه ولذا سنحول أول كود ذكرناه في هذا الدرس إلى الشكل التالي :

```
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = T4.Fields("tb_category.name("
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = T4.Fields("tb_factory.name("
```

أيضاً لا تنس أن تغلق T4 بنهاية هذه الدالة .

عرض السجل الذي تم النقر عليه في مربع النص والقائمة .

بما أننا نريد أن نعرض بيانات أي حقل بمجرد النقر عليه ، فإننا نحتاج بمجرد النقر عليه للحصول على رقمه ، ولنر سوية هذا الامر .
عندما انقر على أي خلية فإن خاصيتي Col و Row يتم ضبطهما طبقاً للوضع الجديد ، وبهذه الطريقة تستطيع الحصول على رقم الصف والعمود بعد أي نقرة باضافة الأمر التالي في حدث Click للجدول :

```
MsgBox "Row : " & MSFlexGrid1.Row & " Col : " & MSFlexGrid1.Col
```

وبنفس الطريقة يمكننا الحصول على محتويات الخلية التي قمنا بالنقر عليها بواسطة أمر كهذا :

```
MsgBox MSFlexGrid1.Text
```

ولكننا لا نريد محتويات هذه الخلية ، بل إننا نحتاج محتويات خلية الرقم في نفس الصف للبحث عنها وعرضها ... لذا فإننا نقوم بالتحويل إلى خلية الرقم :

```
MSFlexGrid1.Col = 1  
MsgBox MSFlexGrid1.Text
```

ستجد أن الأمر السابق يعطيك رقم الصف بغض النظر عن مكان النقر .
وهذا الرقم هو ما سوف نستخدمه في البحث .

سنبدأ الآن بكتابة جملة استعلام نبحث من خلالها عن السجل الذي له هذا الرقم ، تذكر أننا سوف نقوم بذلك من خلال استعلام مزدوج لأننا نريد معرفة اسم المصنع والنوع وليس أرقامهما ، لذلك سوف يكون مثل الاستعلام في الدرس السابق لكن مع اضافة شرط آخر ل **Where** بحيث يصبح **and** **Number = MSFlexGrid1.Text** حيث أننا لا نريد سوى سجل واحد فقط .

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number and  
tb_product.number =" & MSFlexGrid1.Text"" &
```

وبنفس طريقة الجدول سوف نضع البيانات المناسبة في مربعات النص :

```
If T4.RecordCount <> 0 Then  
    T4.MoveFirst  
  
    Text7.Text = T4.Fields("tb_product.Number")  
    Text1.Text = T4.Fields("tb_product.name")  
    Text2.Text = T4.Fields("tb_category.name")  
    Text3.Text = T4.Fields("tb_factory.name")  
    Text4.Text = T4!price  
    Text5.Text = T4.Fields("Count")  
    Text6.Text = T4!Box_Count
```

وماذا أيضاً ... نحتاج لأن تشير القائمة إلى البضاعة الحالية ، وبما أننا نتعامل في القائمة مع Tb_Product.name فسيصبح الأمر بالشكل التالي :

```
Combo1.Text = T4!.Fields("tb_product.name"
```

ولكن بما أننا سنستخدم هذا الأمر لاحقاً فسوف نضعه في إجراء جديد ونسميه ShowData ... وبذلك يصبح الأمر بالشكل التالي :

```
Private Sub ShowData()  
    Text7.Text = T4.Fields("tb_product.Number"  
    Text1.Text = T4.Fields("tb_product.name"  
    Text2.Text = T4.Fields("tb_category.name"  
    Text3.Text = T4.Fields("tb_factory.name"  
    Text4.Text = T4!price  
    Text5.Text = T4.Fields("Count"  
    Text6.Text = T4!Box_Count  
End Sub
```

أخيراً لا تنس أن تغلق قاعدة البيانات T4 .

ثمة أمر آخر ، وهو أننا نريد أيضاً عند اختيار اسم من القائمة أن يظهر في مربعات النصوص ، لذلك بنفس الطريقة السابقة في الاستعلام سوى أننا لن نبحث بدلالة الرقم بل بدلالة الاسم ، ولذا سنضع الاسم بين علامتي تنصيص مفردة ' ' .

```
Private Sub Combo1_Click()  
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and
```

```
tb_product.factory=tb_factory.number and  
tb_product.name ="" & Combo1.Text"" &  
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
```

```
If T4.RecordCount <> 0 Then  
    T4.MoveFirst  
    Call ShowData  
End If
```

```
T4.Close  
End Sub
```

ما زالت أماننا خطوة واحدة ، وهو عند حدث البداية للفورم ، فما هي المحتويات التي ستظهر في
مربعات النصوص ... بالتأكيد سوف تكون عن أول بضاعة ، لذا فسوف نضع الأمر التالي :

```
Combo1.ListIndex = 0
```

وبالتالي سوف ينفذ حدث Click للقائمة تلقائياً .

حل آخر لنقل المعلومات إلى مربعات النص عند النقر على الجدول .

بالتأكيد أنت لا زلت تتذكر الطريقة التي عرضنا فيها البيانات عند النقر على الجدول
ولو لاحظت أننا نضع البيانات الموجودة في الجدول ... أليس كذلك ؟
إذاً فلماذا نقوم بعملية استعلام رغم أن بإمكاننا قراءة المعلومات من الجدول .
في الطريقة السابقة كنا نقرأ الرقم فقط بالكود التالي :

```
MSFlexGrid1.Col = 1
```

لكننا في نفس الوقت نستطيع الانتقال إلى العمود الثاني والثالث والرابع وحتى الأخير ووضع المعلومات بهم في مربعات النصوص ... انظر الكود التالي :

```
MSFlexGrid1.Col = 1
Text7.Text = MSFlexGrid1.Text
MSFlexGrid1.Col = 2
Text1.Text = MSFlexGrid1.Text
MSFlexGrid1.Col = 3
Text2.Text = MSFlexGrid1.Text
MSFlexGrid1.Col = 4
Text3.Text = MSFlexGrid1.Text
MSFlexGrid1.Col = 5
Text4.Text = MSFlexGrid1.Text
MSFlexGrid1.Col = 6
Text5.Text = MSFlexGrid1.Text
MSFlexGrid1.Col = 7
Text6.Text = MSFlexGrid1.Text
```

هل يبدو الأمر سهلاً ؟ عموماً لك مطلق الحرية في استخدام ما تراه مناسباً وقد أرفقت الكودين في البرنامج لاختيار ما تريد .

وماذا بعد ؟ تخيل أن مربعات النصوص لدينا بها خاصية **Index** تبدأ من ٠ وتنتهي ب ٦ ولهم جميعاً اسم **Text1** .

هل تستطيع تخيل مدى بساطة الكود السابق في هذه الحالة :

```
For i = 0 To 7
  MSFlexGrid1.Col = i + 1
  Text1.Text(i) = MSFlexGrid1.Text
Next i
```

لكن هذا الأمر لن يناسبنا في هذه الشاشة لأننا قمنا بعمل مربعات النصوص
Text1,Text2,.....,Text7 . ولكننا سوف نتذكر ذلك في تصميم الشاشات القادمة .

الانتظار في البرنامج

من المناسب دائماً تحسب تضخم قاعدة بيانات البرنامج وما ينتج عنها من بطء في الاستخدام ، وبالتالي
ضرورة البحث عن أوامر أقل جهداً على المعالج ، وفي حال الضرورة نبحث عن وسيلة لانتظار انتهاء
العملية .

لادراك ذلك قم بنسخ السجلات الموجودة في قاعدة البيانات والخاصة بالبضاع حتى يصبح العدد ١٠٠٠
بضاعة مثلاً . إذا لم يكن حاسبك سريعاً فسوف تلاحظ بطء العرض ووجود فترة انتظار تتراوح بين (٢
ثانية للكمبيوتر السريع - ١٥ ثانية للجهاز البطيء) ، كما أن عليك دائماً توقع الأسوء خصوصاً وأن
هذا البرنامج موجه إلى سوبر ماركت (جهاز كومبيوتر متواضع غالباً) .

من أجل ذلك سوف نظهر للمستخدم أثناء فترة التحميل رسالة انتظار ، ولنصمم فورم بسيط جداً يحتوي
على رسالة (الرجاء الانتظار قليلاً) .

ايضاً قم بالضغط على Ctrl+T لتظهر لك شاشة الأدوات ... اختر منها Microsoft
Windows Common Coontrol 6.0 ، ثم ارسم الاداة ProgressBar على
الفورم .

لا تنس أن تخفي CntrolBox الخاصة بالفورم وأن يظهر في المنتصف .
أخيراً اجعل اسم الفورم Frm_Wait .

سوف نستخدم هذا الفورم منذ الآن ليظهر في جميع العمليات التي قد يوجد بها بعض التأخير . وأول
عملية سوف نطبق عليها هذا الفورم هي Refresh_Me .

بعد أن نتأكد من أن `RecordCount <> 0` سوف نظهر هذا الفورم بتعليمة `Show` وبنهاية الامر سوف نخفيه بواسطة `Unload Frm_Wait` .
والآن جرب البرنامج ...

لو لاحظت حتى الآن لم نتعامل مع أداة `ProgressBar` وقبل البدء في التعامل معها سوف نذكر ببعض الخصائص المهمة لهذه الأداة .

- Scrolling** : وتحدد أسلوب العرض .
- Max** : وتحدد القيمة القصوى للشريط .
- Min** : وتحدد القيمة الصغرى ... غالباً ٠ .
- Value** : وتحدد القيمة الحالية للشريط .

والآن سوف نرى ما هي القيم التي نحتاج إليها في الكود .
العملية التي سنحتاج إلى اظهار هذه الشاشة فيها هي عملية تعبئة الجدول حيث أنها تأخذ الوقت الأكبر . ولذا :

- Max = عدد السجلات** .
- Min = صفر** .
- Value** : تأخذ قيماً من ٠ وحتى عدد السجلات حسب رقم السجل الحالي .

لن نتأخر في الشرح كثيراً وسنبدا في الكود مباشرة .
في `Frm_Show` وبعد أن تظهر `Frm_Wait` مباشرة قم بكتابة .

```
Frm_Wait.Show  
Frm_Wait.ProgressBar1.Max = T1.RecordCount  
Frm_Wait.ProgressBar1.Min = 0  
Frm_Wait.ProgressBar1.Value = 0
```

والآن ... نعلم أن قيم هذا الشريط سوف تتغير مع كل سجل ، فتبدأ مع أول سجل = ٠ ، ثم تزيد حتى آخر سجل . لذلك سنكتب بعد حلقة التكرار التي سوف تملئ البيانات في الجدول :

```
Frm_Wait.ProgressBar1.Value =  
Frm_Wait.ProgressBar1.Value + 1  
DoEvents
```

السطر الأول واضح لكن السطر الثاني يعطي فرصة للمعالج لكي يقوم بعملية الزيادة - جرب ازلتها ولاحظ الفرق - .

بنهاية هذا الدرس يصبح حدث **Form_Load** بالشكل التالي :

```
Private Sub Form_Load()  
Frm_Show.Height = 5520  
Frm_Show.Width = 8145  
Frm_Show.Left = ((MDIForm1.Width - Frm_Show.Width) /  
2)  
Frm_Show.Top = ((MDIForm1.Height - Frm_Show.Height)  
/ 2) - 400  
Me.Show  
Refresh_Me  
End Sub
```

وحدث **Refresh_Me** :

```
Sub Refresh_Me()  
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where
```



```
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number"  
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
```

```
If T4.RecordCount = 0 Then Exit Sub
```

```
Frm_Wait.Show  
Frm_Wait.ProgressBar1.Max = T1.RecordCount  
Frm_Wait.ProgressBar1.Min = 0  
Frm_Wait.ProgressBar1.Value = 0
```

```
T4.MoveLast  
T4.MoveFirst
```

```
Combo1.Clear  
Create_Flex
```

```
If T4.RecordCount <> 0 Then  
T4.MoveFirst  
    For i = 0 To T4.RecordCount - 1  
        Frm_Wait.ProgressBar1.Value =  
Frm_Wait.ProgressBar1.Value + 1  
        DoEvents  
        MSFlexGrid1.Row = i + 1  
        MSFlexGrid1.Col = 1  
        MSFlexGrid1.Text = T4.Fields("tb_product.Number("'  
        MSFlexGrid1.Col = 2  
        MSFlexGrid1.Text = T4.Fields("tb_product.name("'  
        MSFlexGrid1.Col = 3  
        MSFlexGrid1.Text = T4.Fields("tb_category.name("'  
        MSFlexGrid1.Col = 4  
        MSFlexGrid1.Text = T4.Fields("tb_factory.name("'  
        MSFlexGrid1.Col = 5  
        MSFlexGrid1.Text = T4!price
```

```

MSFlexGrid1.Col = 6
MSFlexGrid1.Text = T4.Fields("Count")
MSFlexGrid1.Col = 7
MSFlexGrid1.Text = T4!Box_Count

Combo1.AddItem T4.Fields("tb_product.name")
T4.MoveNext
Next i

MSFlexGrid1.Col = 2
MSFlexGrid1.Sort = 7

MSFlexGrid1.Col = 0
For i = 0 To T4.RecordCount - 1
    MSFlexGrid1.Row = i + 1
    MSFlexGrid1.Text = i + 1
Next i
End If

T4.Close
Combo1.ListIndex = 0

Unload Frm_Wait
End Sub

```

ملف مرفق :
قم بمطالعة الملف prog7.zip في المرفقات لمزيد من التفاصيل .

عمليات الحذف في البرنامج

في هذا الدرس سوف نتعلم كيفية القيام بعملية الحذف ، والتي نستخدم لها الوظيفة Delete .
في البداية لا بد أن نتأكد من أن ثم معلومات في قاعدة البيانات وإلا فإننا لن نقوم بأي شيء ، وقبل ذلك لا بد من التأكد من نيته في الحذف عن طريق اظهار رسالة تحذيره :

```
Dim sure
sure = MsgBox ("هل أنت متأكد أنك تريد القيام بحذف هذه البضاعة ؟" ,
vbExclamation + vbYesNo + arabic("حذف بضاعة" ) ,
If sure = vbNo Then Exit Sub
```

والآن نريد أن نصل إلى السجل المحدد ، وهو الذي يظهر رقمه في Text7.Text ، لاحظ على سبيل المثال الأمر التالي :

```
If T1.RecordCount = 0 Then
MsgBox vbExclamation + arabic("لا توجد أي بضاعة !" ) ,
vbExclamation + arabic("بضاعة" )
Exit Sub
End If
```

والآن فإننا سوف نتنقل من خلالها حتى العثور على السجل المطلوب :

```
T1.MoveFirst
For i = 0 To T1.RecordCount - 1
If T1!Number = Text7.Text Then Exit For
T1.MoveNext
Next i
```

ولكن هذه الطريقة تصبح أقل فاعلية في حالة كون عدد السجلات كبيراً . من الأفضل دائماً استخدام جمل لغة الاستعلامات الهيكلية لحذف السجل في جملة بهذا الشكل مثلاً :

```
SQL = "delete * from Tb_Product where number=" &
Text7.Text
```

المشكلة أن الداو **Dao** لا تدعم أيّاً من جمل الإستعلام ما عدا جملة **Select** - حسبما أعلم - . ولهذا سوف اضطر - مرغماً - على استخدام حلول بديلة أكثر جهداً وأقل فاعلية .

والآن سنقوم بحذف السجل الذي عثرنا عليه :

```
T4.Delete
```

وأخيراً نقوم بتحديث المحتويات مرة أخرى .

```
Refresh_Me
```

تبقى أمر صغير ذو بال في عملية الحذف ، وهو أننا نختبر المصنع أو النوع الخاص بالعنصر المحذوف فإذا كان هذا المصنع محذوفاً **View=False** فإننا سنحذف ذلك المصنع ، ولقد شرحنا هذه النقطة في بداية الدروس .

الاحتفاظ بقيم هذين الحقليين في متغيرين :

```
A1 = T1!Category
A2 = T1!Factory
```

الانتقال إليها في **T2** و **T3**

```

T2.MoveFirst
For i = 0 To T2.RecordCount - 1
    If T2!Number = A1 Then Exit For
    T2.MoveNext
Next i

```

```

T3.MoveFirst
For i = 0 To T3.RecordCount - 1
    If T3!Number = A2 Then Exit For
    T3.MoveNext
Next i

```

التأكد من أن View <> False وإلا فإننا نحذفها لو كانت قد فرغت .

```

If T2!View = False Then
    SQL = "select count(*) as mycount from tb_product
where category=" & A1"" &
    Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
    If T4!mycount = 0 Then T2.Delete
    T4.Close
End If

```

```

If T3!View = False Then
    SQL = "select count(*) as mycount from tb_product
where factory=" & A2"" &
    Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
    If T4!mycount = 0 Then T3.Delete
    T4.Close
End If

```

أيضاً بالامكان عمل جملة استعلام واحدة بدل كل ما سبق من الأكواد هل تستطيع القيام بتكوينها !!!

وبهذا يصبح الكود الكامل لعملية الحذف :

```
Dim sure
sure = MsgBox ("هل أنت متأكد أنك تريد القيام بحذف هذه البضاعة ؟" ,
vbExclamation + vbYesNo + arabic("حذف بضاعة" ) ,
If sure = vbNo Then Exit Sub

If T1.RecordCount = 0 Then
    MsgBox vbExclamation + arabic "حذف " ,
    "حذف بضاعة !" ،
    بضاعة"
    Exit Sub
End If

T1.MoveFirst
For i = 0 To T1.RecordCount - 1
    If T1!Number = Text7.Text Then Exit For
    T1.MoveNext
Next i
Dim A1, A2
A1 = T1!Category
A2 = T1!Factory
T1.Delete

T2.MoveFirst
For i = 0 To T2.RecordCount - 1
    If T2!Number = A1 Then Exit For
    T2.MoveNext
Next i

T3.MoveFirst
For i = 0 To T3.RecordCount - 1
    If T3!Number = A2 Then Exit For
    T3.MoveNext
Next i
```

```

If T2!View = False Then
    SQL = "select count(*) as mycount from tb_product
where category=" & A1"" &
    Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
    If T4!mycount = 0 Then T2.Delete
    T4.Close
End If

If T3!View = False Then
    SQL = "select count(*) as mycount from tb_product
where factory=" & A2"" &
    Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
    If T4!mycount = 0 Then T3.Delete
    T4.Close
End If

Refresh_Me

```

التعديل في البرنامج

من أجل التعديل سوف نقوم بانشاء شاشة جديدة ولكنها ليست من Child الفورم الرئيسي ذلك أننا نريد أن نظهرها بالتعليمة 1 Show حتى ينهي التعديل قبل العودة للتحكم ثانية .

قم بتصميم الفورم كما يحلو لك مع مراعاة احتواءه على كافة الحقول ما عدا حقل الرقم فقط . واضف زرین للتعديل ، ولإلغاء الأمر .

ولا تنس ضبط خواص TabIndex لمزيد من السهولة في الاستخدام .
وأخيراً قم بتسميته Frm_Edit .

لا تنس أمراً هاماً جداً ، وهو أن حقلي المصنع والنوع سوف يظهران على هيئة قائمة بها كل المصانع والأنواع لتسهيل عملية الاختيار .

لا تنس أن تضع مكان أو متغير لتحتفظ فيه برقم البضاعة التي يتم تعديلها ، من أجل عملية التعديل لاحقاً .

ولا تنس أخيراً أن تجعل MaxLength لحقل الاسم ٣٠ للتوافق مع قاعدة البيانات ، أما الأرقام فاجعلها = ١٠ كحد أقصى .

وبعد أن قمنا بتصميم الشاشة ... سوف نقوم بتصميم حدث البداية لها Form_Load والذي سيتضمن وضع جميع المصانع والأنواع المتوفرة في القائمتين Combo1 و Combo2 .

```
T2.MoveFirst  
T3.MoveFirst  
Combo1.Clear  
Combo2.Clear
```

```
For i = 0 To T2.RecordCount - 1  
    Combo1.AddItem T2!Name  
    T2.MoveNext  
Next i
```

```
For i = 0 To T3.RecordCount - 1  
    Combo2.AddItem T3!Name  
    T3.MoveNext  
Next i
```

والآن إلى أمر تشغيل هذا الفورم ، فسوف نقوم بتشغيله ومن ثم ننقل القيم الحالية إلى أماكنها المناسبة في فورم التعديل . ولا تنس أن تحتفظ بالنوع والمصنع القديمين والتأكد من عدم خلوهما من البضائع - كما هو الأمر في عملية الحذف تماماً .


```

Dim A1, A2
SQL = "select number from tb_category where name=" &
Text2.Text & "' " &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
A1 = T4!Number
SQL = "select number from tb_factory where name=" &
Text3.Text & "' " &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
A2 = T4!Number

```

```

Load Frm_Edit

```

```

With Frm_Edit
. Text1.Text = Text1.Text
. Combo1.Text = Text2.Text
. Combo2.Text = Text3.Text
. Text2.Text = Text4.Text
. Text3.Text = Text5.Text
. Text4.Text = Text6.Text
. lbl_name & " : تعديل البضاعة" = Text1.Text
. lbl_num = "no." & Text7.Text
End With

```

```

Frm_Edit.Show 1

```

```

T2.MoveFirst
For i = 0 To T2.RecordCount - 1
. If T2!Number = A1 Then Exit For
. T2.MoveNext
Next i

```

```

T3.MoveFirst

```

```

For i = 0 To T3.RecordCount - 1
  If T3!Number = A2 Then Exit For
  T3.MoveNext
Next i

If T2!View = False Then
  SQL = "select count(*) as mycount from tb_product
where category=" & A1"" &
  Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
  If T4!mycount = 0 Then T2.Delete
  T4.Close
End If

If T3!View = False Then
  SQL = "select count(*) as mycount from tb_product
where factory=" & A2"" &
  Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
  If T4!mycount = 0 Then T3.Delete
  T4.Close
End If

Refresh_Me

```

إذا قمنا بالضغط على (الغاء الأمر) في فورم التعديل فلن يكون أكثر من اغلاق الفورم :

Unload Me

أما إذا ضغطنا على زر التعديل فنتنتظرنا جملة استعمال لعملية التعديل .
 وأولاً علينا أن نعرف ما هو رقم البضاعة ، ولو لاحظت في البرنامج أنني اخزن رقم البضاعة في أداة
 تحتوي قبل الرقم على no. ولذا يمكنني **استخلاص الرقم بالطريقة التالية** :

```
Dim My_Number
My_Number = Mid(lbl_num.Caption, 4,
Len(lbl_num.Caption))
```

مبدئياً تأكد من أن الاسم الجديد ليس مطابقاً لأحد الأسماء القديمة ، وبالتالي ينبغي ألا يكون له عدد ، غير نفس البضاعة طبعاً .

```
SQL = "select * from tb_product where name=" &
Text1.Text" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4.RecordCount <> 0 Then
If T4!Number <> Val(My_Number) Then
MsgBox vbExclamation "توجد بضاعة لها نفس الاسم ، فضلاً اختر اسماً آخر" ،
+ arabic"تعديل بضاعة" ،
Exit Sub
End If
End If
```

ثم لا تنس أن تعرف رقم المصنع والنوع الجديدين فالرقم هو من ستخزنه في جدول البضائع لا الاسم .

```
Dim Num1, Num2
SQL = "select number from tb_category where name=" &
Combo1.Text" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
Num1 = T4!Number
T4.Close
SQL = "select number from tb_factory where name=" &
Combo2.Text" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
Num2 = T4!Number
T4.Close
```

وأخيراً ... قم بعملية التعديل .

ومجدداً ظهرت لدي مشاكل مع لغة الاستعلامات الهيكلية ومع الأمر **Update** تحديداً ، لذا اضطررت إلى استخدام تلك الطريقة التي استخدمتها في الحذف وذلك للتعديل :

```
T1.MoveFirst
For i = 0 To T1.RecordCount - 1
    If T1!Number = Val(My_Number) Then Exit For
    T1.MoveNext
Next i

With T1
    . Edit
    ! Name = Text1.Text
    ! category = Num1
    ! factory = Num2
    ! price = Val(Text2.Text)
    ! Count = Val(Text3.Text)
    ! Box_count = Val(Text4.Text)
    . Update
End With
```

ولا تنس اخلاق الشاشة :

Unload Me

لا تنس التأكد من أن المدخلات عبارة عن ارقام فقط في خانات (العدد - السعر - وحدة/صندوق)
وذلك بالشكل التالي

```
If KeyAscii > 58 Or KeyAscii < 48 Then
    KeyAscii = 0
End If
```

وسنضع هذا الأمر في حدث **KeyPress** الذي يمثل ضغطة المفاتيح .
حيث أن رقم آسكي للأعداد يتراوح بين هذين العددين .
ولكننا سنحتاج لبعض الاستثناءات فرقم ٨ يمثل عملية **BackSpace** كما أن ٤٦ يمثل الفاصلة ، لذلك سوف نستثيهما :

```
If (KeyAscii > 58 Or KeyAscii < 48) And KeyAscii <> 46
And KeyAscii <> 8 Then
    KeyAscii = 0
End If
```

هذه الطريقة هي أبسط الطرق وأسهلها ، لكنها لا تفيد في حالة قام المستخدم بلمس الرقم باستخدام
الماوس ، كما أنها تمنع بعض العمليات المفيدة مثل النقر على **Ctrl+V** للقص **Paste** ،
Ctrl+X للقص **Cut** .

أخيراً ، كنا في شاشة التعديل عندما نشغل الفورم نضيف جميع المصانع والأنواع ، ومن ضمنها بعض
المصانع المحذوفة (شرحنا كيفية ذلك في بداية الدروس) ولذا إذا اختار مصنعاً أو مورداً محذوفاً
فسنذكره بذلك قبل عملية التعديل :

```
If T4!View = False Then MsgBox "لا تنس أنك اخترت مورداً سبق لك حذفه" +
vbExclamation +
"مورد محذوف" arabic
```

و

```
If T4!View = False Then MsgBox "لا تنس أنك اخترت نوعاً سبق لك حذفه" +
vbExclamation +
"نوع محذوف" arabic
```

اضافة بضاعة جديدة

في حال لم يكن لدينا أي أنواع أو مصانع فلن نستطيع اضافة أي بضاعة ، وهذا ما سنكتبه أيضاً عندما نظهر هذا الفورم .
في البداية سوف نختبر قيمة Index كما فعلنا مع القائمة الأولى .

```
Select Case Index
Case 0

Case 1

Case 2

End Select
```

والخيار الأول هو ما يهمنا خلال هذا الدرس . والآن سنتأكد من وجود مصانع - موردين و أنواع .

```
If T2.RecordCount = 0 Then
    MsgBox "لا يمكنك اضافة بضاعة مع عدم وجود أي أنواع مدرجة ، قم باضافة نوع أولاً" + vbExclamation + " من قائمة اضافة - اضافة نوع . قبل البدء باضافة البضائع" ،
    arabic" اضافة بضاعة"
    Exit Sub
End If
If T3.RecordCount = 0 Then
    MsgBox "لا يمكنك اضافة بضاعة مع عدم وجود أي موردين لديك ، قم باضافة مورد" + vbExclamation + " أولاً من قائمة اضافة - اضافة مورد . قبل البدء باضافة البضائع" ،
    arabic" اضافة بضاعة"
    Exit Sub
End If
```

وأخيراً اظهر الفورم الخاص بالاضافة :

Frm_Add.Show 1

والآن تصميم شاشة الازافة ، ولقد صممتهامثل شاشة التعديل تماماً .

*** من الممكن استخدام شاشة التعديل للاضافة مع اخفاء بعض العناصر فقط ، إلا أننا ابتعدنا عن

هذا الإسلوب تجنباً للتداخل .

ومجدداً ، سوف نقوم مع كل تشغيل بتعبئة القوائم بالأنواع والمصانع الحالية .

```
T2.MoveFirst
T3.MoveFirst
Combo1.Clear
Combo2.Clear
For i = 0 To T2.RecordCount - 1
    Combo1.AddItem T2!Name
    T2.MoveNext
Next i

For i = 0 To T3.RecordCount - 1
    Combo2.AddItem T3!Name
    T3.MoveNext
Next i
```

كما أننا سنشئ اجراء لافراغ الخانات الأربع ، وسنسميه Clear_Text :

```
Sub Clear_Text()
Text1.Text"" =
Text2.Text"" =
Text3.Text"" =
Text4.Text"" =
End Sub
```

وسنستدعيه في Form_Load كما سنجعل كل من القائمتين يؤشران لأول عنصر فيهما :

```
Clear_Text  
Combo1.ListIndex = 0  
Combo2.ListIndex = 0
```

وسنضع أمر التأكد من ادخال أرقام للخانات الثلاث الأخيرة . وقد شرحناه الدرس السابق :

```
If (KeyAscii > 58 Or KeyAscii < 48) And KeyAscii <> 46  
And KeyAscii <> 8 Then  
    KeyAscii = 0  
End If
```

كما لا ننسى أمر (الغاء الأمر) والخاص باغلاق الفورم :

```
Unload Me
```

والآن سننفذ أمر الاضافة عند النقر ، ومبدئياً لا بد أن يكون قد اختار اسماً للبضاعة ، أما العدد والسعر فإذا كان لا يحتاج إليهما فيمكنه عدم اضافتهما .
ولذا نتأكد من عدم فراغ الخانة في البداية :

```
If Text1.Text = "" Then  
    MsgBox vbExclamation + arabic "لا بد من ادخال اسم البضاعة !" ،  
    "اضافة بضاعة"  
    Exit Sub  
End If
```

ومن ثم نتأكد من عدم وجود بضاعة بنفس الاسم لدينا في قاعدة البيانات :


```

SQL = "select * from tb_product where name='" &
Text1.Text"'" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4.RecordCount <> 0 Then
    MsgBox vbExclamation "توجد بضاعة لها نفس الاسم ، فضلاً اختر اسماً آخر"،
    + arabic"إضافة بضاعة" ،
    Exit Sub
End If

```

والآن سوف نبدأ أولاً بالحصول على رقم النوع والمورد اللذان تم اختيارهما ، مع اعلامه فيما إذا اختار نوعاً أو مورداً سبق حذفه ، وذلك كما تعلمنا في الدرس السابق :

```

T4.Close
Dim Num1, Num2
SQL = "select * from tb_category where name='" &
Combo1.Text"'" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
Num1 = T4!Number
If T4!View = False Then MsgBox "لا تنس أنك اخترت نوعاً سبق لك حذفه
vbExclamation +
arabic"نوع محذوف" ،
T4.Close
SQL = "select * from tb_factory where name='" &
Combo2.Text"'" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
Num2 = T4!Number
If T4!View = False Then MsgBox "لا تنس أنك اخترت مورداً سبق لك حذفه
vbExclamation +
arabic"مورد محذوف" ،
T4.Close

```

والآن سوف نضع المعلومات في أماكنها المناسبة في قاعدة البيانات :

```
With T1
. AddNew
! Name = Text1.Text
! category = Num1
! factory = Num2
! price = Val(Text2.Text)
! Count = Val(Text3.Text)
! Box_count = Val(Text4.Text)
. Update
End With
```

لكننا هذه المرة لن نغلق الشاشة بعد انتهاء عملية الاضافة ، بل سوف نفرغ الخانات لاستقبال بضاعة جديدة ، وبإمكانه في حال أراد الخروج الضغط على زر انتهاء .

```
MsgBox "تمت الاضافة بنجاح ... إذا كنت تريد الانتهاء اضغط على زر انتهاء أو واصل اضافة MsgBox
، "اضافة بضاعة"vbInformation + arabic"البضائع"،
Clear_Text
Text1.SetFocus
```

هذا هو الملف بعد الانتهاء من عمليات الدروس السابقة :

ملف مرفق :
قم بمطالعة الملف prog9zip في المرفقات لمزيد من التفاصيل .

اضافة موردين (مصانع) أو أنواع .

بما أن الشاشتان يقومان بنفس المهمة فسوف نجعلهم جميعاً في شاشة واحدة توفيراً لعدد الشاشات في التصميم ... وسنفرد بينهم حسب محتويات عنوان الشاشة (اضافة مورد أو اضافة نوع) ، وسنضبط المحتوى عند تشغيل الفورم .

كما يمكننا التفريق بينهم من خلال خاصية Tag ولن لا يعرف هذه الخاصية فهي عبارة عن قيمة للفورم يمكننا التخزين فيها دون أن تظهر للمستخدم ، ويمكننا مثلاً اخبار البرنامج بأنه إذا كانت tag=0 فهذا معناه الشاشة الأولى ، وإلا فهي الثانية مثلاً ، عموماً لن نستخدم هذه الطريقة وسنعمد على الطريقة الأولى .

لنصمم فورماً صغيراً كل ما يحتوي عليه هو ؛ أزرار للاضافة ، حذف الحالي ، تعديل الاسم ، وللانتهاء كذلك يحتوي على قائمة بالموردين أو الأنواع الحالية ، ومربع نص لنضع به المورد أو النوع الذي نرغب في اضافته .

وأخيراً اسم الفورم Frm_Add_2

كذلك لا تنس أن تغير الاسم في القائمة من (اضافة مورد جديد - اضافة نوع جديد) إلى (التحكم بالموردين - التحكم بالأنواع) . حيث أن الأمر لن يقتصر على الاضافة بل سيشمل الحذف والتعديل أيضاً .

والآن إلى أمر اظهار هذه الشاشة :

Case 1

Frm_Add_2.Ibl_name "معلومات الأنواع" =

Frm_Add_2.Refresh_Me

Frm_Add_2.Show 1

Case 2

Frm_Add_2.Ibl_name "معلومات الموردين" =

Frm_Add_2.Refresh_Me

Frm_Add_2.Show 1

وهذا جزء من الأمر الكلي للقائمة ...

والآن ما هو أمر Refresh_Me ؟

إنه إجراء Sub لكنه من نوع Public بحيث يمكن مناداته من خارج الفورم ، ووظيفته تكمن في ملء القائمة بالمحتويات المناسبة :

```
Public Sub Refresh_Me()  
List1.Clear  
If lbl_name.Caption "معلومات الأنواع" = Then  
    If T2.RecordCount <> 0 Then T2.MoveFirst  
    For i = 0 To T2.RecordCount - 1  
        List1.AddItem T2!Name  
        T2.MoveNext  
    Next i  
Else  
    If T3.RecordCount <> 0 Then T3.MoveFirst  
    For i = 0 To T3.RecordCount - 1  
        List1.AddItem T3!Name  
        T3.MoveNext  
    Next i  
End If  
End Sub
```

لكن وبما أن هذه شاشة متخصصة بعرض الأنواع والموردين ، فلا بد من وضع كلمة بجانب الموردين والأنواع المحذوفة ، لاحظ الكود التالي :

```
If T2!View = False Then List1.AddItem T2!Name " &  
"(محذوف)Else List1.AddItem T2!Name
```

كما ينبغي أيضاً كتابة رقم النوع أو المورد بجانبه ، ولذا سيكون الكود كالتالي :

```
If T2!View = False Then List1.AddItem T2!Number & "-" &  
T2!Name "(محذوف)" & Else List1.AddItem T2!Number & "-" &  
T2!Name
```

والآن إلى كتابة أوامر الأزرار ، مبدئياً لا تنس أمر اغلاق الشاشة :

Unload Me

ولنر الكود الخاص بعملية الاضافة ... وسيكون النظام بالشكل التالي :

- * التأكد أولاً من كتابة اسم في مربع النص .
- * التأكد من نوع العملية (اضافة مورد أم اضافة نوع) .
- * التأكد من عدم وجود مورد أو نوع بنفس الاسم سابقاً .
- * اضافة النوع أو المورد الجديد إلى قاعدة البيانات .
- * تحديث القائمة عن طريق استدعاء أمر Refresh_Me ، وافراغ مربع النص .

سنضيف جميع هذه الأوامر في اجراء جديد نسميه MyAdd .

```
Sub MyAdd()  
If Text1.Text = "" Then  
    MsgBox vbExclamation + arabic ،  
    "لا بد من ادخال اسم البضاعة !" ،  
    "اضافة بضاعة"  
    Exit Sub  
End If  
  
If lbl_name.Caption = "معلومات الأنواع" Then  
    'add category...  
    SQL = "select * from tb_category where name='" &  
    Text1.Text"'" &  
    Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,  
    If T4.RecordCount <> 0 Then  
        MsgBox vbExclamation + arabic ،  
        "يوجد نوع له نفس الاسم ، فضلاً اختر اسماً آخر" ،  
        "اضافة نوع"  
        Text1.SetFocus  
        Text1.SelStart = 0
```

```

Text1.SelLength = Len(Text1.Text)
Exit Sub
End If

T2.AddNew
T2!Name = Text1.Text
T2.Update

Else

'add factory...
SQL = "select * from tb_factory where name=" &
Text1.Text &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4.RecordCount <> 0 Then
    MsgBox vbExclamation + "يوجد مورد له نفس الاسم ، فضلاً اختر اسماً آخر" ،
arabic"إضافة مورد" ،
    Text1.SetFocus
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1.Text)
Exit Sub
End If

T3.AddNew
T3!Name = Text1.Text
T3.Update

End If

Refresh_Me
Text1.Text =
End Sub

```

والآن ، **قبل البدء بأي عملية حذف أو تعديل** لا بد من التأكد أن ثم شيئاً محدداً من القائمة ، ويتم التأكد من ذلك بالشكل التالي :

```
If List1.ListIndex = -1 Then
    MsgBox vbExclamation + arabic ، "عليك اختيار عنصر من القائمة أولاً" ، "حذف"
    Exit Sub
End If
```

شيء آخر ولكنه موجود في التعديل فقط ، لابد من معرفة هل هناك بيانات في مربع النص أم لا ، ويتم ذلك بالشكل التالي :

```
If Text1.Text = "" Then
    MsgBox "لا بد من ادخال الاسم الجديد لاستبداله بالاسم القديم !" ، vbExclamation + arabic ، "إضافة بضاعة"
    Exit Sub
End If
```

الجزء الآخر المطلوب هو معرفة رقم ذلك العنصر ، ولو لاحظت أن هذا الرقم يبدأ من اليمين وباتجاه اليسار وتكون نهايته قبل علامة "-" مباشرة ، فلنر إن كان بإمكاننا تحويل ذلك إلى كود :

```
Dim txt
Dim i
i = 1
Do While (Mid(List1.Text, i, 1) <> "-")
    txt = txt & Mid(List1.Text, i, 1)
    i = i + 1
Loop

MsgBox txt
```

هذا صحيح تماماً ، فهنا ستحتوي txt على الرقم ، ولكن بصورة نصية ، وعند الاستعلام سوف نستعلم عن Val(txt) .

والآن لنضع هذا الكود في عمليتي الحذف والتعديل .

*** يمكننا وضع هذا الأمر بعد ذلك في دالة حيث نستخدمها من أي مكان عند الحاجة إليها .

سنبدأ في الأجزاء القادمة في التفصيل ، وسنتعامل الآن مع عملية التعديل .

بحسب نوع العملية ينبغي أن نبحث هل يوجد نوع أو مورد له نفس الاسم قبل الاكمال :

```
SQL = "select * from tb_category where name=" &
Text1.Text"'" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4.RecordCount <> 0 Then
    MsgBox vbExclamation + "يوجد نوع له نفس الاسم ، فضلاً اختر اسماً آخر"،
    ، "إضافة نوع" arabic
    Text1.SetFocus
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1.Text)
Exit Sub
End If
```

ونفس الامر بالنسبة للموردين .

الجزء الآخر هو الذهاب إلى هذا السجل عن طريق T2 أو T3 ، ويتم ذلك بالدوران حتى العثور على الرقم الذي قمنا بتخزينه في txt .

```
If T2.RecordCount <> 0 Then T2.MoveFirst
For i = 0 To T2.RecordCount - 1
    If T2!Number = Val(txt) Then Exit For
    T2.MoveNext
Next i
```


وكذلك وبنفس الطريقة مع الموردين ، وأخيراً أمر التعديل :

```
T2.Edit  
T2!Name = Text1.Text  
T2.Update
```

وبذلك يصبح الكود الكامل لزر التعديل :

```
If List1.ListIndex = -1 Then  
    MsgBox vbExclamation + arabic ،  
    "عليك اختيار عنصر من القائمة أولاً" ،  
    "تعديل"  
    Exit Sub  
End If  
  
If Text1.Text = "" Then  
    MsgBox "لا بد من ادخال الاسم الجديد لاستبداله بالاسم القديم !" ،  
    vbExclamation + arabic "إضافة بضاعة" ،  
    Exit Sub  
End If  
  
Dim txt  
Dim i  
i = 1  
Do While (Mid(List1.Text, i, 1) <> "-")  
    txt = txt & Mid(List1.Text, i, 1)  
    i = i + 1  
Loop  
  
If lbl_name.Caption = "معلومات الأنواع" Then  
    'edit category...  
    SQL = "select * from tb_category where name='" &  
    Text1.Text &
```

```

Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4.RecordCount <> 0 Then
    MsgBox vbExclamation + "يوجد نوع له نفس الاسم ، فضلاً اختر اسماً آخر" ،
    arabic"إضافة نوع" ،
    Text1.SetFocus
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1.Text)
Exit Sub
End If

```

```

If T2.RecordCount <> 0 Then T2.MoveFirst
For i = 0 To T2.RecordCount - 1
    If T2!Number = Val(txt) Then Exit For
    T2.MoveNext
Next i

```

```

T2.Edit
T2!Name = Text1.Text
T2.Update

```

```

Else

```

```

'edit factory...

```

```

SQL = "select * from tb_factory where name='" &
Text1.Text"'" &

```

```

Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4.RecordCount <> 0 Then
    MsgBox vbExclamation + "يوجد مورد له نفس الاسم ، فضلاً اختر اسماً آخر" ،
    arabic"إضافة مورد" ،
    Text1.SetFocus
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1.Text)
Exit Sub

```

```
End If
```

```
If T3.RecordCount <> 0 Then T3.MoveFirst
```

```
For i = 0 To T3.RecordCount - 1
```

```
    If T3!Number = Val(txt) Then Exit For
```

```
    T3.MoveNext
```

```
Next i
```

```
T3.Edit
```

```
T3!Name = Text1.Text
```

```
T3.Update
```

```
End If
```

```
Refresh_Me
```

```
Text1.Text"" =
```

والآن سنعود لاستئناف عملية الحذف .

كنا قد توقفنا عند استخلاص رقم العنصر المحدد ، والآن سننتقل إليه سواء كان مورداً أم نوعاً ، لكننا

لن نقوم بحذفه ، بل سوف نجعل `View = False`

وأخيراً سوف نتأكد من وجود بضائع تحته وإلا فإننا نحذفه نهائياً .

مبدئياً لا بد من تحذيره قبل عملية الحذف :

```
Dim sure
```

```
sure = MsgBox ("هل أنت متأكد أنك تريد القيام بعملية البحث هذه ؟") ،
```

```
vbExclamation + vbYesNo + arabic("تأكيد الحذف !") ،
```

```
If sure = vbNo Then Exit Sub
```

وبذلك يصبح الكود الكامل بهذه الطريقة :

```
If List1.ListIndex = -1 Then
    MsgBox vbExclamation + arabic ، "عليك اختيار عنصر من القائمة أولاً" ،
    "حذف"
    Exit Sub
End If

Dim sure
sure = MsgBox ("هل أنت متأكد أنك تريد القيام بعملية الحذف هذه ؟" ،
vbExclamation + vbYesNo + arabic("تأكيد الحذف !") ،
If sure = vbNo Then Exit Sub

Dim txt
Dim i
i = 1
Do While (Mid(List1.Text, i, 1) <> "-" & Mid(List1.Text, i, 1)
    txt = txt & Mid(List1.Text, i, 1)
    i = i + 1
Loop

If lbl_name.Caption = "معلومات الأنواع" Then
'delete category...

If T2.RecordCount <> 0 Then T2.MoveFirst
For i = 0 To T2.RecordCount - 1
    If T2!Number = Val(txt) Then Exit For
    T2.MoveNext
Next i
```

```
T2.Edit
T2!View = False
T2.Update
```

```
SQL = "select count(*) as mycount from tb_product where
category=" & Val(txt) & "
```

```
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
```

```
If T4!mycount = 0 Then T2.Delete
```

```
T4.Close
```

```
Else
```

```
'delete factory...
```

```
If T3.RecordCount <> 0 Then T3.MoveFirst
```

```
For i = 0 To T3.RecordCount - 1
```

```
    If T3!Number = Val(txt) Then Exit For
```

```
    T3.MoveNext
```

```
Next i
```

```
T3.Edit
```

```
T3!View = False
```

```
T3.Update
```

```
SQL = "select count(*) as mycount from tb_product where
factory=" & Val(txt) & "
```

```
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
```

```
If T4!mycount = 0 Then T3.Delete
```

```
T4.Close
```

```
End If
```

```
Refresh_Me
```

```
Text1.Text =
```

ملف مرفق :

قم بمطالعة الملف prog10.zip في المرفقات لمزيد من التفاصيل .

قائمة مهام

لأن التقارير هي أكثر شاشات البرنامج تشعباً فقد تركناها مؤقتاً لإكمال باقي الأجزاء ومن ثم العودة إليها ، ولذا سوف ننتقل إلى **قائمة (مهام)** مباشرة .

وكما قمنا مع شاشة (تعديل نوع ومصنع) والتي دمجنا فيها الشاشتين معاً ، فكذلك سنعمل مع شاشتي البيع والشراء . واللذان تحتاجان إلى المحتويات التالية :

- * أداة عنوان **Lable** لتوضيح نوع العملية .
- * قائمة تحتوي على البضائع الموجودة حالياً .
- * حقل التاريخ والذي سيكون من نوع جديد هو أداة **MaskEdit** وسنرى كيف نستخدمها .
- * الكمية المطلوبة للبيع أو الشراء . وبجانبها **Option** لتحديد هل العدد بالصندوق أم بالوحدة .
- واجعل خاصية **Value=True** للوحدة .
- * السعر المدفوع في العملية .
- * زر لإضافة العملية ، وزر لإغلاق الشاشة .

وقبل أن نبدأ في شرح استخدام أداة نعود إلى الموديول فكما جعلنا **T1** ترمز لجدول البضائع **T2,T3** لجدولي الأنواع والموردين ، فإننا سنجعل **T5** ترمز لجدول المبيعات ... أما **T4** فستظل من أجل الاستعلامات .

```
Set T5 = D1.OpenRecordset("Tb_Sel_Bay",  
dbOpenTable
```

والآن ... قم بالضغط على زري **Ctrl+T** واختر الاداة **Microsoft Masked Edit Control 6.0** ، وقم برسمها بالحجم المطلوب على الفورم .
والآن من الخصائص اضغط على (...) التي بجانب **Custem** ستظهر لك شاشة بأهم خصائص هذه الأداة ، ومنها :

Format : والتي سنضع فيها التنسيق المناسب لعملية التاريخ ، وهو : **dd/mm/yyyy** ، وتحدد هذه الخاصية كيفية قراءة وعرض الرقم لاحقاً . وبهذا نستطيع أن نقول أن الناتج سيكون على شكل تاريخ ، لكننا لا يمكن بهذه الخاصية أن نمنع المستخدم من ادخال أكثر من ١٠ خانات (عدد خانات التاريخ) ولا أن نمنعه من ادخال حروف .
Mask : تحدد هذه الخاصية ما يمكن أن ندعوه (قناع الإدخال) والذي سيحدد طبيعة المدخلات وعددها ، ولأننا لا نحتاج سوى أرقام فسوف نستخدم الرمز # لتشكيل قناع بالشكل التالي :
#####/###/##
Prompt Char : الحرف الذي سيوضع في مكان الفارغ (لاحظ أنه سيوضع كصورة فقط في حين أنه لن يؤثر على القيمة) ولنجعلها ٠ مثلاً .

بعدها قم باكمال الفورم بالطريقة التي نقوم بها مع كل فورم .
ولا تنس اضافة أمر **Refresh_Me** والذي سنقوم ببرمجته لاحقاً .

لماذا لم استخدم أداة **DTPicker** ... أفضل هذه الأداة في البرمجة لسبب وحيد ألا وهو أن أداة **DTPicker** تقرأ التاريخ بالميلادي ، لكن أسلوب العرض يختلف من جهاز لآخر ، وهذا ما قد يؤدي إلى عرض الشهور العربية في الأداة على رغم نفس قراءة التاريخ الميلادي مما يسبب بمشاكل بالبرنامج ، وحتى أجد حلاً لهذه المشكلة فسأظل على استخدام **MaskedEdit** .

والآن سنعود إلى **MDIForm1** وسنكتب أمر اظهار هذه الشاشة .

```

Private Sub Operation_Click(Index As Integer)
If T1.RecordCount = 0 Then
    MsgBox "لا توجد أي بضائع لعرضها ، لاضافة بضاعة جديدة اختر اضافة - اضافة vbExclamation + arabic" عرض البضائع ،
    Exit Sub
End If
Select Case Index
Case 0
    Frm_Sel_Bay.Ibl_name.Caption = "اضافة عملية بيع"
    Frm_Sel_Bay.Refresh_Me
    Frm_Sel_Bay.Show 1
Case 1
    Frm_Sel_Bay.Ibl_name.Caption = "اضافة عملية شراء"
    Frm_Sel_Bay.Refresh_Me
    Frm_Sel_Bay.Show 1
End Select
End Sub

```

ولكن ما هي محتويات أمر Refresh_Me ؟

سوف يحتوي هذا الأمر على ملء قائمة البضائع بجميع البضائع المتوفرة ، كما سيحتوي على أمر ملء خانة التاريخ بالتاريخ الحالي ، ولأننا لا نريد تغيير التاريخ مع كل عملية اضافة إلى التاريخ الحالي لمنع الازعاج لأنه قد يسجل عمليات تمت باليوم السابق ، وكما كنا نفعل مع كل شاشة فسنكتب الأمر بالشكل التالي :

```

Public Sub Refresh_Me()
Combo1.Clear
For i = 0 To T1.RecordCount - 1
    Combo1.AddItem T1!Name
    T1.MoveNext
Next i

```



```
Combo1.ListIndex = 0
```

```
Dim Myday, Mymonth, Myyear
```

```
Myday = Day(Now)
```

```
If Len(Myday) = 1 Then Myday = 0 & Myday
```

```
Mymonth = Month(Now)
```

```
If Len(Mymonth) = 1 Then Mymonth = 0 & Mymonth
```

```
Myyear = Year(Now)
```

```
MaskedTextBox1.Text = Myday & "/" & Mymonth & "/" &
```

```
Myyear
```

```
End Sub
```

يبدو الجزء الأول مفهوماً ، أما لماذا قمنا بذلك مع الجزء الثاني ولم نكتب =
DateTime.Date مباشرة .

وفي الواقع لعدة أسباب ، أولها أن هذه التعليمية تعطينا قيمة بالشكل التالي : DD/MM/YY أي
بصيغة Short ونحن نحتاج إلى صيغة Long للعمل في البرنامج .
أما عند اضافة ٠ فهذا لازم لأن تعليمة مثل Day(Now) قد تعطينا مثلاً : ١ أو ٢ ، فلا بد من
تحويلها إلى ٠١ و ٠٢ .

لا تنس أمر الاغلاق Unload Me في زر الإنهاء .

والآن سوف نبدأ بكتابة تعليمات زر الاضافة .

في البداية لا بد أن نتأكد من أنه ادخل كمية ، أما السعر فيمكن أن يكون ٠ ، وذلك حتى نتيح
للمستخدم استخدام البرنامج كحافضة لعمليات البيع والشراء ليس إلا .

```
If Val(Text2.Text) <= 0 Then
```

```
MsgBox vbExclamation + arabic "اضافة "لا بد من ادخال الكمية" ،
```

```
عملية"
```

```
Exit Sub
```

```
End If
```

وأخيراً نتأكد من نوع العملية المطلوبة ، والتي سننشأ لكل منها اجراء باسمي Sel و Buy ونستدعيهما من خلال الأمر السابق .

```
If lbl_name.Caption = "إضافة عملية بيع" Then  
    BUY  
Else  
    SEL  
End If
```

*** برمجة أمر SEL :**

لا تحتاج برمجة هذا الأمر إلى كثير من الجهد ، بل كل ما نحتاج إليه هو إضافة هذه العملية إلى قاعدة البيانات ، ومن ثم زيادة خاصية Count للبضاعة التي تم اختيارها . حسب العدد المطلوب .

```
T1.MoveFirst  
For i = 0 To T1.RecordCount - 1  
    If T1!Name = Combo1.Text Then Exit For  
    T1.MoveNext  
Next i  
  
Dim cnt  
cnt = Val(Text2.Text)  
If Option2.Value = True Then cnt = cnt * T1!Box_count
```

هنا يتم الانتقال للبضاعة ، ومن ثم تعريف المتغير cnt والذي سيحتوي على العدد ، والذي قد يتم ضربه بعدد الوحدات في الصندوق إذا كان مدخلاً بالصندوق .

With T5

```

. AddNew
! product = T1!Number
! Date = MaskedTextBox1.Text
! Count = cnt
! price = Val(Text1.Text)
! kind = 1
. Update
End With

T1.Edit
T1!Count = T1!Count + cnt
T1.Update

```

وهذا الجزء الخاص بالاضافة إلى قاعدة البيانات ، وزيادة عدد البضاعة الموجودة .
أخيراً سنظهر رسالة باتمام العملية والعدد المتوفر حالياً .

```

MsgBox & "T1!Count " & "تمت العملية بنجاح ، يوجد من البضاعة حالياً : " &
vbInformation + arabic" ، "اتمام العملية" وحدة" ،

```

ونفرض الخانات تمهيداً لعملية جديدة :

```

Text1.Text"" =
Text2.Text"" =

```

برمجة الأمر BUY :

يشبه هذا الأمر SEL إلى حد كبير ، ويختلف عنه في أننا نحتاج أولاً إلى أن نتأكد من توفر الكمية المطلوبة قبل العملية ، وإلا فإننا نظهر رسالة تحذيره بالكمية الموجودة حالياً .

وكما هي بداية أمر Sel يبدأ الأمر BUY بالانتقال إلى السجل المطلوب وتحديد الكمية في cnt :

```
T1.MoveFirst
For i = 0 To T1.RecordCount - 1
    If T1!Name = Combo1.Text Then Exit For
    T1.MoveNext
Next i

Dim cnt
cnt = Val(Text2.Text)
If Option2.Value = True Then cnt = cnt * T1!Box_count
```

ومن ثم أمر التأكد من توفر الكمية المطلوبة :

```
If cnt > T1!Count Then
    MsgBox & "الموجود حالياً : " & cnt & " الكمية التي تطلبها غير متوفرة وهي : " & T1!Count, vbExclamation + arabic"كمية غير متوفرة"
    Exit Sub
End If
```

ومن ثم أوامر الاضافة في قاعدة البيانات ونقص العدد بمقدار العملية :

```
With T5
    . AddNew
    ! product = T1!Number
    ! Date = MaskedTextBox1.Text
    ! Count = cnt
    ! price = Val(Text1.Text)
    ! kind = 0
    . Update
End With
```

```
T1.Edit  
T1!Count = T1!Count - cnt  
T1.Update
```

وأخيراً الرسالة التي تفيد اتمام العملية بنجاح :

```
MsgBox & " T1!Count " & "تمت العملية بنجاح ، يوجد من البضاعة حالياً : "  
vbInformation + arabic" ، " اتمام العملية " وحدة "
```

ونفرض الخانات تمهيداً لعملية جديدة :

```
Text1.Text"" =  
Text2.Text"" =
```

والآن عد مجدداً إلى Frm_Show واكتب في زري (عملية بيع ، عملية شراء) الأوامر التالية :

*** عملية بيع :**

```
Frm_Sel_Bay.lbl_name.Caption = "إضافة عملية بيع"  
Frm_Sel_Bay.Refresh_Me  
Frm_Sel_Bay.Combo1.Text = Text1.Text  
Frm_Sel_Bay.Text1 = Text4.Text  
Frm_Sel_Bay.Show 1  
  
Refresh_Me
```

*** هناك خطأ في تسمية الفورم Frm_Sel_Bay ، ولم الحظ انني كتبت a بدلاً من U إلا مؤخراً .

وهذا الخطأ مكرر أيضاً في قاعدة البيانات (جدول Tb_Sel_Bay) أيضاً .

* عملية شراء :

```
Frm_Sel_Bay.lbl_name.Caption = "إضافة عملية شراء"  
Frm_Sel_Bay.Refresh_Me  
Frm_Sel_Bay.Combo1.Text = Text1.Text  
Frm_Sel_Bay.Text1 = Text4.Text  
Frm_Sel_Bay.Show 1
```

Refresh_Me

حيث أن هذه الأوامر اظهار طبيعي للشاشة لكن مع وضع اسم البضاعة الحالية كافتراضي ، ووضع السعر الحالي كافتراضي ايضاً ، ومن ثم اطلاق أمر التحديث .

التحكم في عمليات البيع والشراء

سوف نصمم الفورم الخاص بهذه العملية بحيث يحتوي على جدول ، وفوقه زر اختيار (عمليات البيع - عمليات الشراء) اضافة إلى تاريخ العمليات المطلوب تعديلها . وذلك في أداة MaskedEdit كما شرحنا في الدروس السابقة .

وبأسفل الفورم يوجد زرین ، للحذف ، وللانتهاء . كما يوجد بأعلاه زر (نفذ) والذي يقوم بتحديث الفورم حسب التاريخ أو العملية الجديدة . ونسمي هذا الفورم Frm_Sel_Bay2 .

ولا تنس لهذا الفورم أن تجعل خاصية MDIChild = True .
وكا هي العادة عندما نشغل أي فورم تابع للفورم MDIForm1 .

Frm_Sel_Bay2.Show

لكن في حدث Form_Load نضبط الأبعاد ونستدعي حدث Refresh_Me ، ونجعل
حقل التاريخ يحتوي على تاريخ اليوم الحالي :

*** نضبط الأبعاد فيما إذا كنا نريد شاشة بارزة - لم أفهم سبب هذا الأمر بعد ! - .**

```
Me.Height = 4335
Me.Width = 5895
Me.Left = ((MDIForm1.Width - Me.Width) / 2)
Me.Top = ((MDIForm1.Height - Me.Height) / 2) - 400
Me.Show
```

```
Dim Myday, Mymonth, Myyear
Myday = Day(Now)
If Len(Myday) = 1 Then Myday = 0 & Myday
Mymonth = Month(Now)
If Len(Mymonth) = 1 Then Mymonth = 0 & Mymonth
Myyear = Year(Now)
MaskedTextBox1.Text = Myday & "/" & Mymonth & "/" &
Myyear
```

Refresh_Me

وسنرى الآن على ماذا سيحتوي أمر Refresh_Me :

في بداية الأمر نحتاج إلى استدعاء دالة باسم Flex تقوم باعداد الجدول .
وستقوم هذه الدالة بمعرفة عدد العمليات بناء على عمليات البيع أو الشراء أولاً .

```
If Option1.Value = True Then
    SQL = "select count(*) as cnt from tb_sel_bay where
    kind=0 and date=#" & Format(MaskedTextBox1.Text,
    "mm/dd/yyyy" & "#") & "
```

Else

SQL = "select count(*) as cnt from tb_sel_bay where
kind=1 and date=#" & Format(MaskedTextBox1.Text,
"mm/dd/yyyy" & " & "

End If

Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,

MyCnt = T4!cnt

حيث نكون قد عرفنا المتغير MyCnt في Genral لأننا سنستخدمه لاحقاً .
ثم تجهز الجدول :

MSFlexGrid1.Clear

MSFlexGrid1.Cols = 6

MSFlexGrid1.Rows = MyCnt + 1

MSFlexGrid1.Row = 0

MSFlexGrid1.Col = 0

MSFlexGrid1.Text"ت" =

MSFlexGrid1.Col = 1

MSFlexGrid1.Text"رقم" =

MSFlexGrid1.Col = 2

MSFlexGrid1.Text"اسم البضاعة" =

MSFlexGrid1.Col = 3

MSFlexGrid1.Text"التاريخ" =

MSFlexGrid1.Col = 4

MSFlexGrid1.Text"الكمية" =

MSFlexGrid1.Col = 5

MSFlexGrid1.Text"السعر" =

MSFlexGrid1.ColAlignment(0) = 5

MSFlexGrid1.ColAlignment(1) = 5

MSFlexGrid1.ColAlignment(2) = 3

MSFlexGrid1.ColAlignment(3) = 5


```
MSFlexGrid1.ColAlignment(4) = 5  
MSFlexGrid1.ColAlignment(5) = 5
```

```
MSFlexGrid1.ColWidth(0) = 500  
MSFlexGrid1.ColWidth(1) = 500  
MSFlexGrid1.ColWidth(2) = 1500  
MSFlexGrid1.ColWidth(3) = 1200
```

نستدعي هذا الأمر في بداية إجراء Refresh_Me .

ثم ننتقل إلى أول سجل في الجدول T5 إذا لم تكن فارغة ، ونبدأ في الدوران من أول سجل إلى آخر سجل ووضع السجل الذي له Kind والتاريخ المناسب في الجدول .

```
Sub Refresh_Me()  
Flex  
Dim Key, n As Integer  
n = 0  
If T5.RecordCount <> 0 Then T5.MoveFirst  
If Option1.Value = True Then  
    Key = 0  
Else  
    Key = 1  
End If  
  
For i = 0 To T5.RecordCount - 1  
    If T5!kind = Key And Format(T5!Date, "dd/mm/yyyy") =  
MaskedTextBox1.Text Then  
        n = n + 1  
        MSFlexGrid1.Row = n  
        MSFlexGrid1.Col = 0  
        MSFlexGrid1.Text = n  
        MSFlexGrid1.Col = 1  
        MSFlexGrid1.Text = T5!Number  
        MSFlexGrid1.Col = 2
```

```

SQL = "select name from tb_product where number="
& T5!product"" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset(
MSFlexGrid1.Text = T4!Name
T4.Close
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = T5!Date
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = T5!Count
MSFlexGrid1.Col = 5
MSFlexGrid1.Text = T5!price
End If
T5.MoveNext
Next i
End Sub

```

هل تعرف أين نحتاج هذا الامر أيضاً ... في زر (تنفيذ) . حيث سنستدعي أمر Refresh_Me

التعديل في عمليات البيع والشراء .

سنضيف زراً جديداً بجانب زر الحذف ، ونسميه تعديل . ويختص هذا الزر بتعديل محتويات البضاعة الحالية إلى محتويات جديدة .

سوف نقوم باعداد فورم سريع يحتوي على خانات التعديل ، وسيكون عبارة عن نسخة من فورم اضافة عملية بيع وشراء ، إلا أن النوع يوضع افتراضياً كما توضع البيانات افتراضياً أيضاً . كما ستحتوي على متغير يوضح رقم العملية التي نقوم بتعديلها حالياً . وسنقوم بذلك عن طريق تعريفه

Varibale
Public

Public Num As Integer

كما أننا لن نعيد اظهار (وحدة - صندوق) مرة اخرى .
أيضاً سيظهر اسم البضاعة على أنه ثابت Lable .

والآن إلى أمر اظهار هذا الفورم :

```
If MSFlexGrid1.Row = 0 Then Exit Sub

MSFlexGrid1.Col = 1
Frm_Edit_SP.Num = MSFlexGrid1.Text
MSFlexGrid1.Col = 2
Frm_Edit_SP.Label3.Caption = MSFlexGrid1.Text
MSFlexGrid1.Col = 3
Frm_Edit_SP.MaskedTextBox1.Text =
Format(MSFlexGrid1.Text, "dd/mm/yyyy")
MSFlexGrid1.Col = 4
Frm_Edit_SP.Text2.Text = MSFlexGrid1.Text
MSFlexGrid1.Col = 5
Frm_Edit_SP.Text1.Text = MSFlexGrid1.Text

Frm_Edit_SP.Show 1
Refresh_Me
```

ولنعد الآن إلى الفورم الجديد ولنر كيف ستتتم عملية التعديل .
في البداية نتأكد من ادخال عدد للبضاعة :

```
If Val(Text2.Text) <= 0 Then
    MsgBox vbExclamation + arabic "إضافة الكمية"،
    "عملية"
    Exit Sub
End If
```

ثم ننتقل إلى السجل صاحب العملية الحالية :

```

If T5.RecordCount = 0 Then Exit Sub
T5.MoveFirst
For i = 0 To T5.RecordCount - 1
    If T5!Number = lbl_num.Caption Then Exit For
    T5.MoveNext
Next i

```

والآن تعني عملية التعديل هذه أن نقوم باعادة الأوضاع كما كانت عليه قبل هذه العملية ، ومن ثم تنفيذ عملية البيع (حذف العملية القديمة ، اضافة العملية الجديدة) تقريباً .

لذا سننتقل أولاً إلى البضاعة المطلوبة :

```

If T1.RecordCount <> 0 Then T1.MoveFirst
For i = 0 To T1.RecordCount - 1
    If T1!Number = T5!product Then Exit For
    T1.MoveNext
Next i

```

ثم نقوم تحديث معلومات **T5** وذلك بالشكل التالي :

```

If T5!kind = 0 Then
    BUY
Else
    SEL
End If

```

حيث أن **Buy** و **Sel** هما العمليتان اللذان قمنا ببرمجتهم خلال الاضافة مع بعض التعديلات البسيطة :

```

Sub BUY()
If Val(Text2.Text) > T1!Count + T5!Count Then
    MsgBox & " الكمية التي تطلبها غير متوفرة وهي : " & Val(Text2.Text) & " : حالياً : " & T1!Count + T5!Count,
vbExclamation + arabic"كمية غير متوفرة" ،
    Exit Sub
End If

T1.Edit
T1!Count = T1!Count + T5!Count
T1!Count = T1!Count - Val(Text2.Text)
T1.Update

With T5
    . Edit
    ! product = T1!Number
    ! Date = MaskedTextBox1.Text
    ! Count = Val(Text2.Text)
    ! price = Val(Text1.Text)
    ! kind = 0
    . Update
End With

MsgBox & " T1!Count " & "تمت العملية بنجاح ، يوجد من البضاعة حالياً : " & T1!Count & " ،
    ، "اتمام العملية"vbInformation + arabic"وحدة"،
Unload Me
End Sub

```

```

Sub SEL()
T1.Edit
T1!Count = T1!Count - T5!Count
T1!Count = T1!Count + Val(Text2.Text)
T1.Update

```

```

With T5
    Edit
    ! product = T1!Number
    ! Date = MaskedTextBox1.Text
    ! Count = Val(Text2.Text)
    ! price = Val(Text1.Text)
    ! kind = 1
    Update
End With

```

```

MsgBox & " T1!Count " & "تمت العملية بنجاح ، يوجد من البضاعة حالياً : " &
vbInformation + arabic" ، "اتمام العملية" وحدة" ،
Unload Me
End Sub

```

تغيير لون الصف المحدد :

لنحقق عملية تغيير لون الصف الذي حددناه نحتاج بادئ ذي بدء إلى عمل دالة تقوم بمسح جميع
الالوان الموجودة في MSFlexGrid ، وسيشير التغيير بعدد الصفوف وهو العدد المخزن في المتغير
Mycnt ، وبالعرض بعدد الأعمدة وهو ه مع تجاهل العمود الرئيسي والصف الرئيسي ... ولنر الآن
هذه الدالة :

```

Sub Clear_Color()
For i = 1 To MyCnt
  For j = 1 To 5
    MSFlexGrid1.Row = i
    MSFlexGrid1.Col = j
    MSFlexGrid1.CellBackColor = QBColor(15)
  Next j
Next i

```

حيث أن QBColor(15) تعني الأبيض لمزيد من المعلومات حول الألوان في فيجوال بيسك ،
راجع الدرس التالي :

<http://www.c4arab.com/showlesson.php?lesid=2320>

ويتم اطلاق هذه الدالة بداية في حدث النقر على الجدول . ولكن بعد أن نحتفظ برقم الصف الحالي في متغير ما .

ماذا سيحدث أيضاً في أمر النقر على الجدول Click ، كل ما لدينا أننا سنبدأ من العمود الثاني وحتى السادس مع ثبات الصف ونجعل اللون أحمر .
لاحظ الكود التالي :

```

If MSFlexGrid1.Row = 0 Then Exit Sub
Now_Row = MSFlexGrid1.Row
Clear_Color
MSFlexGrid1.Row = Now_Row
For i = 1 To 5
  MSFlexGrid1.Col = i
  MSFlexGrid1.CellBackColor = QBColor(12)
Next i

```

*** ويمكن بدلاً من مسح كل الجدول باللون الأبيض مع كل ضغطه عليه ، يمكننا مسح فقط آخر سجل تم تحديده ، وذلك بتخزين قيمة آخر سجل بعد عملية التحديد .

سنبدأ الآن في برمجة زر الحذف :

في بداية عملية الحذف نقوم باظهار الرسالة التقليدية ، هل ما زلت تريد الاستمرار ؟ وذلك بعد التأكد من أننا لسنا في Row=0 .

```
If MSFlexGrid1.Row = 0 Then Exit Sub
Dim sure
sure = MsgBox ("هل تريد بالتأكيد حذف هذه العملية ؟" + vbQuestion + vbYesNo + arabic(" " ,
If sure = vbNo Then Exit Sub
```

والآن سنظهر خياراً جديداً ، هل تريد حذف هذه العملية وكافة النتائج التي ترتبت عليها (زيادة أو انقاص البضاعة) أم الحذف المجرد فقط ، أم الغاء الأمر بالكلية ؟

```
sure = MsgBox ("هل تريد حذف العملية وكافة النتائج المترتبة عليها ( زيادة أو نقص sure = MsgBox vbQuestion + vbYesNoCancel + arabic ("أم الحذف فقط ؟" ,
vbQuestion + vbYesNoCancel + arabic ("تأكيد الحذف"
If sure = vbCancel Then Exit Sub
```

والآن نتنقل كما هي العادة إلى السجل المطلوب :

```
MSFlexGrid1.Col = 1
If T5.RecordCount <> 0 Then T5.MoveFirst
MsgBox MSFlexGrid1.Text
For i = 0 To T5.RecordCount - 1
    If T5!Number = MSFlexGrid1.Text Then Exit For
    T5.MoveNext
Next i
```


والآن إذا كان يريد ازلتها مع كافة النتائج المترتبة عليها فسنقوم بعمل ذلك ، وذلك بالشكل التالي :

```
If sure = vbYes Then
  If T1.RecordCount <> 0 Then T1.MoveFirst
  For i = 0 To T1.RecordCount - 1
    If T1!Number = T5!product Then Exit For
    T1.MoveNext
  Next i
  T1.Edit
  Select Case T5!kind
  Case 0
    T1!Count = T1!Count + T5!Count
  Case 1
    T1!Count = T1!Count - T5!Count
  End Select
  T1.Update
End If
```

ثم يأتي دور أمر الحذف .

```
T5.Delete
```

أخيراً لا تنس أن تستدعي أمر Refresh_Me ثانية .

```
Refresh_Me
```

تعديلات بسيطة في أمر Refresh_Me الخاص بشاشة التحكم في المبيعات .

التعديل الأول بسيط جداً ، كل ما عليك هو اضافة الامر التالي لنهاية الإجراء :

```
MSFlexGrid1.Row = 0
```

وذلك لكي يقف المؤشر على (لاشيء) بعد هذه العملية .

التعديل الثاني هو استخدام Frm_Wait في العملية ، لذلك سنضع الأمر التالي قبل حلقة التكرار

```
Frm_Wait.Show  
Frm_Wait.ProgressBar1.Max = T5.RecordCount  
Frm_Wait.ProgressBar1.Min = 0  
Frm_Wait.ProgressBar1.Value = 0
```

وبداخل حلقة التكرار نكتب الأمر التالي :

```
Frm_Wait.ProgressBar1.Value =  
Frm_Wait.ProgressBar1.Value + 1  
DoEvents
```

وفي نهاية الكود كما تعلمنا سابقاً في Frm_Show .

Unload Frm_Wait

ملف مرفق :

قم بمطالعة الملف prog12.zip في المرفقات لمزيد من التفاصيل .

التعامل مع قائمة مسح

والآن سنتابع دروسنا ، وسنتعامل مع القائمة الفرعية (مسح) والتي تحتوي على ما يلي :

- * جميع البضائع .
- * جميع الأنواع .
- * جميع الموردين .
- * جميع عمليات البيع .
- * جميع عمليات الشراء .

وستقوم بعمل ذلك عبر اجراء مشترك .

وقبل بدء كل عملية لابد من اظهار رسالتي تحذير لهذه العملية - على الأقل - حيث أنها تمسح كما
كبيراً من البيانات .

والرسالتان موحدة لجميع العمليات ، وستكون بالشكل التالي :

```
Dim sure
sure = MsgBox ("هل تريد بالتأكيد اتمام عملية الحذف هذه ؟" , vbYesNo +
vbExclamation + arabic("تأكيد عملية الحذف" ,
If sure = vbNo Then Exit Sub
sure = MsgBox ("إذا قمت بالمتابعة فلن يمكنك التراجع ، هل ما زلت تريد الاستمرار ؟" ,
vbExclamation + vbYesNo + arabic("تأكيد عملية الحذف" ,
If sure = vbNo Then Exit Sub
```

ومن ثم سنبدأ في التفريعات المختلفة حسب قيمة خاصية Index ، ومع كل قيمة سوف نستدعي
اجراءً واحداً فقط ، لكننا نغير قيمة الرسالة التي تظهر به ، ونغير الجدول الذي تتم حذف عملياته ،
لاحظ الأمر التالي :

```
Sub DEL(ByRef Table As Recordset, ByVal msg As
String)
```

هذا يعني أننا في الاجراء سوف نستقبل Table من نوع RecordSet (يمكن أن يكون T1
أو T2 ...) اضافة إلى msg وهو نص رسالة التحذير التي ستظهر .
ومعنى ByRef أن التغيير الذي سيحدث في Table خلال الاجراء سوف يتم تطبيقه على
الجدول الذي أرسلناها ، أما ByVal فمعنى ذلك أن التغيير لن ينعكس على الأصل .

وسيكون هذا الاجراء بالشكل التالي :

```

Sub DEL(ByRef Table As Recordset, ByVal msg As String)
If Table.RecordCount = 0 Then
    MsgBox "عرض vbExclamation + arabic "عرض  

    البضائع  

    Exit Sub
End If

Dim sure
sure = MsgBox(msg, vbYesNo + vbExclamation + arabic " ،  

    "تأكيد عملية الحذف")
If sure = vbNo Then Exit Sub

Table.MoveFirst
For i = 0 To Table.RecordCount - 1
    Table.Delete
    Table.MoveNext
Next i

MsgBox "اتمام عملية vbInformation + arabic "تمت عملية الحذف بنجاح" ،  

    الحذف"
End Sub

```

أما الجزء الثاني وهو حذف المبيعات والمشتريات ، فعلى نفس الطريقة يمكننا عمل اجراء واحد لهم ، ولكنه يختلف في قيمة ٠ أو ١ والتي ترمز للبيع أو للشراء . لاحظ الكود التالي :

```

Sub DEL2(ByVal knd As Byte)
If T5.RecordCount = 0 Then
    MsgBox "عرض vbExclamation + arabic "عرض  

    البضائع  

    Exit Sub
End If

```

```

T5.MoveFirst
For i = 0 To T5.RecordCount - 1
    If T5!kind = knd Then T5.Delete
    T5.MoveNext
Next i

```

، "اتمام عملية vbInformation + arabic تمت عملية الحذف بنجاح" ، MsgBox
الحذف"
End Sub

من أجل ذا يكون الاستدعاء حسب الجدول بالشكل التالي :

```

Select Case Index
Case 0
    DEL T1 ، "إن حذف البضائع يقتضي حذف جميع عمليات البيع والشراء ، هل تريد الاستمرار؟"
Case 1
    DEL T2 ، "إن حذف الأنواع يقتضي حذف جميع البضائع ، هل تريد الاستمرار؟"
Case 2
    DEL T3 ، "إن حذف الموردين يقتضي حذف جميع البضائع ، هل تريد الاستمرار؟"
Case 4
    DEL2 0
Case 5
    DEL2 1
End Select

```

*** ملاحظة هامة : لو عدت إلى البرنامج لوجدت الكثير من الدوال التي يمكن اختصارها بهذه الطريقة ، وبعمل Array لأدوات عرض للجدول ، وغيرها .

ملف مرفق :
قم بمطالعة الملف prog13.zip في المرفقات لمزيد من التفاصيل .

قائمة أدوات

تغيير معلومات المحل :

في هذه الشاشة سنصمم شاشة صغيرة تحتوي على اسم المحل ، رقم الهاتف ، وأخرى .

ونسماه Frm_User .

والآن ، سنبدأ في أوامر استدعاء هذه الشاشة .

```
Select Case Index
Case 0
    Frm_User.Text1.Text = Info(2,
    Frm_User.Text2.Text = Info(3,
    Frm_User.Text3.Text = Info(4,
    Frm_NewUser.Show 1
End Select
```

وأمر Unload Me في زر اغلاق هذا الفورم .

والآن سنقوم بالآتي :

* اضافة حدث التأكد من الأرقام فقط في Key_Press لمربع رقم الهاتف . وذلك كما تعلمنا في

الدروس السابقة .

* وضع قيم افتراضية للمعلومات في حالة عدم وجودها في بداية زر (تعديل) .

```
If Text1.Text = "" Then
    Text1.Text = "برنامج البائع الذكي"
End If
```

```
If Text2.Text = "" Then
```

```

Text2.Text = "0"
End If

If Text3.Text = "" Then
    Text3.Text = "البرنامج من تصميم منتدى فيجوال بيسك للعرب" =
    Text3.Text = Text3.Text & vbNewLine & "بامكانك زيارة الموقع على
    www.vb4arab.com/vb" هذا الرابط
    Text3.Text = Text3.Text & vbNewLine & "أو مراسلة صاحب
    Hammada2091@Hotmail.com" البرنامج على هذه البريد الإلكتروني
End If

```

※ استخدام الأمر **Edit** لتعديل محتويات السجل .

```

Set T4 = D1.OpenRecordset("Tb_User", dbOpenTable(
T4.Edit
    T4!Name = Text1.Text
    T4!Telephone = Text2.Text
    T4!More = Text3.Text
T4.Update

```

※ تحديث معلومات المصفوفة **Info** واغلاق قاعدة البيانات .

```

T4.MoveFirst
    Info(2) = T4!Name
    Info(3) = T4!Telephone
    Info(4) = T4!More

T4.Close

```

※ اخفاء الشاشة واستدعاء الحدث Start .

```
Unload Me  
Call Start
```

تغيير كلمة السر :

سنقوم هذه العملية من خلال InputBox دون تصميم فورم ، وذلك بالشكل التالي .
※ نضيف ثلاث متغيرات لتخزين كلمات المرور وتأكيد تغيير كلمة المرور :

```
Dim A1, A2, sure
```

※ طلب كلمة المرور القديمة ، ومقارنتها بالكلمة الموجودة للاستمرار :

```
A1 = InputBox("فضلاً ادخل كلمة المرور القديمة" ، "تغيير كلمة المرور")  
If A1 = "" Then Exit Sub  
If A1 <> Info(1) Then  
    MsgBox vbCritical + arabic"كلمة مرور خاطئة !" ،  
    Exit Sub  
End If
```

※ طلب ادخال كلمة المرور الجديدة ونخزنها في A1 :

```
A1 = InputBox("ادخل كلمة المرور الجديدة" ، "تغيير كلمة المرور")  
If A1 = "" Then Exit Sub  
If Len(A1) < 3 Or Len(A1) > 12 Then  
    MsgBox vbExclamation +  
arabic"تسجيل مستخدم جديد" ،  
    Exit Sub  
End If
```


* طلب ادخال تأكيد كلمة المرور في A2 ، ومقارنتها بالكلمة الأولى في A1 :

```
A2 = InputBox("ادخل تأكيد كلمة المرور الجديدة"، "تغيير كلمة المرور")
If A2 = "" Then Exit Sub
If A1 <> A2 Then
    MsgBox "تغيير كلمة vbCritical + arabic "كلمتا مرور غير متطابقتين !"،
المرور"
Exit Sub
End If
```

* اظهار رسالة تأكيد تغيير كلمة المرور :

```
sure = MsgBox vbYesNo + "هل تريد بالتأكيد تغيير كلمة المرور"،
vbExclamation + arabic("تغيير كلمة المرور")
If sure = vbNo Then Exit Sub
```

* فتح قاعدة البيانات وتعديلها بالكلمة الجديدة :

```
Set T4 = D1.OpenRecordset("Tb_User", dbOpenTable)
T4.Edit
T4!PassWord = A1
T4.Update
```

* وأخيراً وضع الكلمة الجديدة في Info(1) واظهار رسالة تذكير بالكلمة القديمة :

```
T4.MoveFirst
Info(1) = T4!PassWord
T4.Close

MsgBox & " Info(1), vbInformation + "لا تنس أن كلمة المرور هي : "
arabic"تذكير بكلمة المرور"
```

ملف مرفق :
قم بمطالعة الملف prog14.zip في المرفقات لمزيد من التفاصيل .

سنتجاوز القائمة الفرعية (النسخ الاحتياطي) مؤقتاً حتى اكمال باقي عناصر القائمة .

تجهيز الطابعة .

سيحتوي فورم تجهيز الطابعة على قائمة بالطابعات الموجودة اضافة إلى زر (تمكين الطابعة) والذي يقوم بجعلها الطابعة الافتراضية . وزر آخر لالغاء الأمر .
والآن صمم هذا الفورم واسمه Frm_Printer .
وكما هي العادة مع كل الفورمات ، قم وضعه في المنتصف ، وأزل الشريط العلوي .

والآن إلى خطوات كتابة الكود :

※ أولاً في أمر اظهار هذه الشاشة نتأكد أولاً من وجود طابعات .

Case 5

```
If Printers.Count = 0 Then
    MsgBox vbExclamation + "لا توجد أي طابعات متصلة بالجهاز"،
    arabic"لا توجد طابعة"
    Exit Sub
End If
Frm_Printer.Show 1
```

※ في حدث **Form_Load** لهذه الشاشة لا بد أن نضع الطابعات المتصلة بالجهاز في القائمة :

```
List1.Clear
Dim I
For I = 0 To Printers.Count - 1
    List1.AddItem Printers(I).DeviceName
Next I
```

* في كود (تمكين الطابعة) سنعينها طابعة افتراضية للكمبيوتر ، ومن ثم نغلق الشاشة :

```
If List1.ListIndex < 0 Then Exit Sub  
Set Printer = Printers(List1.ListIndex)  
Unload Me
```

* وفي زر الغاء الأمر :

```
Unload Me
```

يحتوي الكائن **Printer** اضافة لما سبق على أهم الوظائف المتعلقة بالطابعة ويمكنك تصفحها بنفسك ، ويمكنك أيضاً استخدام المجموعة **Printers** ، لمعرفة المزيد حول كائن الطابعة يمكن مراجعة درس الأستاذ تركي العسيري (كائن الطابعة للمبتدئين) . على هذا الرابط :

<http://72.14.207.104/search?q=cache:1iNBcyeAXGwJ:www.vb4arab.com/vb/showthread.php>

وكما هو ظاهر من الرابط فهذه الصفحة إحدى صفحات **Google** المخبئة لمنتدى فيجوال بيسك للعرب قبل اغلاقه . وإذا لم تكن مرتاحاً لهذا الأسلوب من عرض المحتويات ، فإن بإمكانك العثور على بعض المعلومات في كتاب (نحو برمجة كائنية التوجه) .

تصميم الآلة الحاسبة

سنتعلم سوياً خلال هذا الدرس عمل آلة حاسبة بسيطة ، وسنحاول جاهدين تلافي كتابة أسطر كثيرة معتمدين في ذلك على خاصية **Index** للأزرار .

ولن لا يعرف خاصية **Index** فهي الخاصية التي تتيح تسمية أكثر من أداة بنفس الاسم مع تغيير قيمة هذه الخاصية فقط ، أو بمعنى آخر إنها تقوم بعمل مصفوفة **Array** من الأدوات .
وتتيح هذه الخاصية أيضاً وضع كود مشترك لجميع هذه الأدوات .

من أجل ذلك سنبدأ في تصميم فورم صغير ، ونضبط خاصية MDIChild بحيث تصبح True ، وذلك لمكن استخدام هذا الفورم أثناء عمل البرنامج .
يحتوي هذا الفورم على ما يلي :

Command1 وب **Index** من ٠ إلى ١٠ تحتوي في خاصية **Caption** على الأرقام بالترتيب من ٠ وحتى ٩ اضافة إلى الفاصلة .
Command2 ويحتوي في خاصية **Caption** على + / - وهي مختصة بتغيير الإشارة .
Command3 وب **Index** من ٠ إلى ٣ تحتوي على العمليات التالية : + ، - ، * ، / .
Command4 ويحتوي على = .
Command5 ويحتوي على **Index** من ٠ وحتى ٢ تحتوي على الرموز التالية **CE** لمسح العملية الحالية ، **C** لافراغ الخانات من جديد ، و **Off** لافراغ الآلة .
Command6 ويحتوي على **Index** من ٠ وحتى ٢ ويحتوي على **MS** لتخزين رقم ، **MR** لاسترجاع التخزين ، **MC** لافراغ التخزين .
Text1 وهو خانة العمليات .
Text2 وهي الخانة التي توضح الإشارة الحالية ، حاول وضعه على المربع الأول بحيث يبدو أنه بداخله .
واضبط خاصيتي **Locked** لهذين المربعين بحيث تصبح **True** لمنع التلاعب فيهما .

والآن أضف ثلاثة متغيرات ، متغير **Memory** من النوع **double** ليحل محل الذاكرة ، ومتغير **First** من نفس النوع والذي سيحتوي على الرقم الأول **There** وهو متغير من النوع **Boolean** وهو يبين لنا إذا ما كنا لتونا قد ضغطنا على أحد العمليات أو = بحيث تسمح الخانة مع أول ضغطة لرقم .

```
Dim First As Double  
Dim There As Boolean  
Dim Memory As Double
```

الأمر الثاني هو التعامل مع الأرقام ، وكما ترى فكل ما نحتاج إليه في الأرقام والفاصلة هو أن نزيد مربع النص Text1 بمحتويات الزر المضغوط عليه .

إذن سنجل مربع النص = مربع النص اضافة إلى Caption للزر المضغوط عليه .
ولكن قبل ذلك نتأكد من أن There تحتوي على False ، وإلا فإننا نفرغ الخانة أولاً :

```
If There = True Then
  There = False
  Text1.Text"" =
End If
Text1.Text = Text1.Text & Command1(Index).Caption
```

والآن إلى جزء العمليات الحسابية ، سنستدعي أولاً دالة نسميها Calc_Me ، ومن ثم نضبط There على True ، ونضع في First الرقم الحالية ، واطافة لذلك ، نضع العلامة الحالية في Text2 .

```
Call Calc_Me
There = True
First = Val(Text1.Text)
Text2.Text = Command3(Index).Caption
```

ولنر كيف سيكون شكل الاجراء Calc_Me ، والذي سنختبر قيمة Text2 وعلى أساسها سوف نقوم بعمل العملية الحسابية بين First و Text1 ونضعها في Text1 :

```
Select Case Text2.Text
Case "+"
  Text1.Text = First + Val(Text1.Text)
Case "-"
  Text1.Text = First - Val(Text1.Text)
Case "*"
  Text1.Text = First * Val(Text1.Text)
```

```
Case"/"
```

```
Text1.Text = First / Val(Text1.Text)  
End Select
```

لكن ولأنه قد تحدث أخطاء ناتجة عن قسمة على الصفر مثلاً ، أو Overflow ، ولكي لا يتوقف البرنامج فإننا نضع العبارة التالية في بداية الاجراء :

```
On Error Resume Next
```

وفي آخر الكود نضع ماذا يحدث لو تم اكتشاف خطأ ، حيث نظهر رسالة بهذا الخطأ ، وتخزن كل معلومات الخطأ في الكائن Err :

```
If Err Then
```

```
MsgBox Err.Description, vbCritical & "خطأ " & Err
```

```
Err.Clear
```

```
End If
```

لنعد مرة أخرى إلى أمر = ، ولو لاحظنا أننا سنستدعي الاجراء Calc_Me ، ومن ثم نفرغ خانة العملية Text2 ونضع There=True .

```
Call Calc_Me
```

```
There = True
```

```
Text2.Text"" =
```

والآن أوامر Command5 والتي تحتوي على C و CE و Off .

```
Select Case Index
```

```
Case 0
```

```
Text1.Text"" =
```

```
Text2.Text"" =
```

```
Case 1
    Text1.Text"" =
```

```
Case 2
    Unload Me
End Select
```

وأخيراً أوامر MS و MR و MC :

```
Select Case Index
Case 0
    Memory = Val(Text1.Text)
Case 1
    Text1.Text = Memory
    There = False
Case 2
    Memory = 0
End Select
```

ويمكنك جلياً أن تلاحظ كم من الأكواد وفرت علينا بفضل الله خاصية **Index** واستدعاء الاجراء **Calc_Me** بدلاً من كتابته مع كل حدث .

والآن ، نرى كود اظهار هذا الفورم ، وسنظهره في الطرف الأعلى والأيسر للفورم .

```
Case 7
    Frm_Calc.Width = 3105
    Frm_Calc.Height = 2640
    Frm_Calc.Top = 0 ' MDIForm1.Height -
Frm_Calc.Height
    Frm_Calc.Left = 0
    Frm_Calc.Show
```

استقبال أرقام من لوحة المفاتيح .

سنضبط أولاً خاصية KeyPreview للفورم بحيث تصبح True وبهذا أي ضغطة على الكيبورد في هذا الفورم يحس بها حدث Key_Press لهذا الفورم .

والآن نعرف أن حدث Key_Press يستقبل متغيراً هو كود آسكي Ascii للزر المضغوط ، ومن خلال ذلك نستطيع أن نضغط على الزر المقابل لها .
فمثلاً الرقم ٤٨ يعني الصفر ، من أجل ذلك سنضغط على Command1(0) بالشكل التالي :

```
Command1(0).Value = True
```

وكذلك مع باقي الأرقام والتي تأخذ أرقاماً من ٤٨ وحتى ٥٧ ، أي الرقم + ٤٨ .
ومن العلاقة السابقة نستطيع أن نكتب الكود التالي في حدث Key_Press :

```
If KeyAscii < 58 And KeyAscii >= 48 Then  
    Command1(KeyAscii - 48).Value = True
```

والآن نكمل بمزيد من التخصيص ، ف ٤٦ تعني الفاصلة ، ٤٢ و ٤٣ و ٤٥ و ٤٧ العمليات الأربع ، ٢٧ زر Esc والذي سنجعله يرمز لـ C في الآلة ، أيضاً ١٣ يرمز لـ Enter والذي سيكون = .
من أجل ذلك نكتب الكود التالي :

```
Elseif KeyAscii = 46 Then  
    Command1(10).Value = True  
Elseif KeyAscii = 13 Then  
    Command4.Value = True  
Elseif KeyAscii = 42 Then  
    Command3(2).Value = True  
Elseif KeyAscii = 43 Then  
    Command3(0).Value = True  
Elseif KeyAscii = 45 Then  
    Command3(1).Value = True
```



```
Elseif KeyAscii = 47 Then  
    Command3(3).Value = True  
Elseif KeyAscii = 27 Then  
    Command5(1).Value = True
```

وأخيراً ٨ ترمز لـ BackSpace . وسنستخدم لذلك دالة Left والتي ستقتصص نصاً من اليسار بطول النص - ١ ، وذلك بالشكل التالي :

```
Elseif KeyAscii = 8 Then  
    If Len(Text1.Text) <> 0 Then Text1.Text =  
    Left(Text1.Text, Len(Text1.Text) - 1)
```

بهذا تكون آلتنا جاهزة لاستقبال الأرقام من لوحة المفاتيح .

ملف مرفق :
قم بمطالعة الملف prog16.zip في المرفقات لمزيد من التفاصيل .

النسخ الاحتياطي

لن نقوم بالنسخ في هذا الدرس بما يمكن أن نسميه نسخاً احتياطياً كاملاً ، بل إن ما سنقوم به هو نسخ ملف قاعدة البيانات .

وسنبداً بعمل اجراءين Copy_Me لعمل نسخة احتياطية ، و Copy_2 لاستعادة نسخة احتياطية ، وسنكتب الأمر بالشكل التالي :

```
Select Case Index  
Case 0  
    Copy_Me  
Case 1  
    Copy_2  
End Select
```

والآن سنضيف أداة **CommonDialog** والتي من ضمن اختصاصاتها المربعات الحوارية للفتح والحفظ ، ولما كان **MDIForm** لا يقبل اضافة أي أدوات عليه سوى **Picture** فإننا سنضيف واحدة ونجعل **Visible=False** ، ونضع عليها الأداة .
ولإضافة الأداة اضغط **Ctrl+T** واختر **Microsoft CommonDialog Control**
6.0 .

برمجة الأمر **Copy_Me** :

في بداية هذا الأمر سوف نكتب عبارة تلافي الأخطاء :

```
On Error GoTo 11
```

وفي آخر الأمر :

```
:11  
Exit Sub
```

وهذا يخبر البرنامج أنه في حال وجدت خطأ توجه إلى السطر ١١ ، وفي السطر ١١ نأمره بإنهاء الاجراء ولكي نجعل المستخدم عندما يضغط على (الغاء الأمر - **Cancel**) يخرج فإننا نجعل الغاء الأمر خطأ بالشكل التالي :

```
CommonDialog1.CancelError = True
```

ومن ثم نضبط العنوان ، ونوع الملفات التي سيتم حفظها :

```
CommonDialog1.CancelError = True  
CommonDialog1.DialogTitle = "اختر المكان المطلوب للحفظ"  
CommonDialog1.Filter = "Access File|*.mdb|
```

وأخيراً نظهر المربع الحواري للحفظ :

CommonDialog1.ShowSave

ستعيد هذه الوظيفة الاسم الكامل للمسار + الملف في الخاصية **FileName** فيما تعيد اسم الملف فقط في **FileTitle** .

سنجري الآن سلسلة من الاختبارات على اسم الملف ، فنتأكد من أنه فارغ ، وأنه ليس ثم ملف يحمل نفس الاسم في نفس المكان :

```
If CommonDialog1.FileTitle = "" Then Exit Sub
```

```
If Dir(CommonDialog1.FileName) <> "" Then  
    MsgBox vbExclamation + "توجد ملف بنفس الإسم في المسار المحدد"،  
    arabic"نسخ واستيراد"  
    Exit Sub  
End If
```

والآن تمهيداً للنسخ لابد من اغلاق قاعدة البيانات :

D1.Close

ومن ثم أمر النسخ :

FileCopy Folder & "db1.mdb", CommonDialog1.FileName

ونعيد فتح قاعدة البيانات من جديد :

```
Set D1 = DBEngine.Workspaces(0).OpenDatabase(Folder  
& "Db1.mdb", False, False, ";pwd=" & PassWord(
```

```
Set T1 = D1.OpenRecordset("Tb_Product", dbOpenTable(
```

```
Set T2 = D1.OpenRecordset("Tb_Category",  
dbOpenTable(
```

```
Set T3 = D1.OpenRecordset("Tb_Factory", dbOpenTable(
```

```
Set T5 = D1.OpenRecordset("Tb_Sel_Bay",  
dbOpenTable(
```

ومن ثم رسالة باتمام العملية بنجاح :

تم نسخ قاعدة البيانات بنجاح إلى : MsgBox & "

، "نسخ CommonDialog1.FileName, vbInformation + arabic

واستيراد"

وبهذا يصبح الكود الكامل لأمر **Copy_Me** بالشكل التالي :

```
Private Sub Copy_Me()
```

```
On Error GoTo 11
```

```
CommonDialog1.CancelError = True
```

```
CommonDialog1.DialogTitle = "اختر المكان المطلوب للحفظ"
```

```
CommonDialog1.Filter = "Access File|*.mdb|
```

```
CommonDialog1.ShowSave
```

```
If CommonDialog1.FileTitle = "" Then Exit Sub
```

```

If Dir(CommonDialog1.FileName) <> "" Then
    MsgBox vbExclamation + "توجد ملف بنفس الإسم في المسار المحدد" ،
    arabic"نسخ واستيراد" ،
    Exit Sub
End If

D1.Close

FileCopy Folder & "db1.mdb", CommonDialog1.FileName

Set D1 = DBEngine.Workspaces(0).OpenDatabase(Folder
& "Db1.mdb", False, False, ";pwd=" & PassWord

Set T1 = D1.OpenRecordset("Tb_Product", dbOpenTable
Set T2 = D1.OpenRecordset("Tb_Category",
dbOpenTable
Set T3 = D1.OpenRecordset("Tb_Factory", dbOpenTable
Set T5 = D1.OpenRecordset("Tb_Sel_Bay",
dbOpenTable

MsgBox & "تم نسخ قاعدة البيانات بنجاح إلى : " &
CommonDialog1.FileName, vbInformation + arabic"نسخ" ،
"واستيراد"
: ١١
Exit Sub
End Sub

```

برمجة الأمر Copy_2 :

كما هو الحال مع الأمر السابق ، نضع أكواد الخطأ ونضبط خصائص المربع الحوار في بداية الكود ،
إلا أننا سنظهر Open بدلاً من Save .

```
On Error GoTo 12
CommonDialog1.CancelError = True
CommonDialog1.DialogTitle = "اختر مكان القاعدة المطلوبة" =
CommonDialog1.Filter = "Access File|*.mdb"|
CommonDialog1.ShowOpen
```

ومن ثم نتأكد من وجود الملف :

```
If CommonDialog1.FileTitle = "" Then Exit Sub

If Dir(CommonDialog1.FileName) = "" Then
    MsgBox vbExclamation + arabic "نسخ قاعدة بيانات خاطئ" ،
    واستيراد"
    Exit Sub
End If
```

ونظهر رسالة تحذيرية بأن استيراد القاعدة سوف يحذف القاعدة الحالية - يمكن لتلافي ذلك نسخها احتياطياً قبل ذلك - .

```
Dim SuRe
SuRe = MsgBox ("إن استيرادك لقاعدة البيانات هذه سوف يؤدي إلى حذف القاعدة  
+ vbExclamation + arabic "الحالية ، هل ما زلت تريد الاستمرار؟" ،  
vbYesNo "نسخ واستيراد" ،  
If SuRe = vbNo Then Exit Sub
```

نغلق قاعدة البيانات الحالية ، ونزيل خاصية للقراءة فقط إن وجدت ، ثم نحذفها ونجلب القاعدة الجديدة مكانها :

D1.Close

SetAttr Folder & "db1.mdb", vbNormal
Kill Folder & "db1.mdb"

FileCopy CommonDialog1.FileName, Folder & "db1.mdb"

نفتح قاعدة البيانات مجدداً ، ونخزن البيانات الجديدة في Info ثم نستدعي الدالة Start .
* لم نقوم بذلك في الاجراء السابق لأن قاعدة البيانات لم تتغير .

Set D1 = DBEngine.Workspaces(0).OpenDatabase(Folder
& "Db1.mdb", False, False, ";pwd=" & PassWord)

Set T1 = D1.OpenRecordset("Tb_Product", dbOpenTable)

Set T2 = D1.OpenRecordset("Tb_Category",
dbOpenTable)

Set T3 = D1.OpenRecordset("Tb_Factory", dbOpenTable)

Set T5 = D1.OpenRecordset("Tb_Sel_Bay",
dbOpenTable)

Set T4 = D1.OpenRecordset("Tb_User", dbOpenTable)

Info(0) = T4!User_Name

Info(1) = T4!PassWord

Info(2) = T4!Name

Info(3) = T4!Telephone

Info(4) = T4!More

T4.Close

Call Start

ونظهر رسالة باتمام العملية بنجاح :

”تم استيراد قاعدة البيانات بنجاح من : “ & MsgBox
& ” سوف يتم الآن بدء تشغيل هذه Chr(13 & CommonDialog1.FileName
، ”نسخ واستيراد“vbInformation + arabic القاعدة“،

وبهذا يكون الأمر الكامل كما يلي :

```
Private Sub Copy_2()  
On Error GoTo 12  
CommonDialog1.CancelError = True  
CommonDialog1.DialogTitle = ”اختر مكان القاعدة المطلوبة“  
CommonDialog1.Filter = "Access File|*.mdb"|  
CommonDialog1.ShowOpen  
  
If CommonDialog1.FileTitle = "" Then Exit Sub  
  
If Dir(CommonDialog1.FileName) = "" Then  
    MsgBox ”نسخ vbExclamation + arabic اسم قاعدة بيانات خاطئ“،  
    ”واستيراد“  
    Exit Sub  
End If  
  
Dim SuRe  
SuRe = MsgBox ”(إن استيرادك لقاعدة البيانات هذه سوف يؤدي إلى حذف القاعدة  
vbExclamation + arabic + الحالية ، هل ما زلت تريد الاستمرار؟“،  
vbYesNo(”نسخ واستيراد“)  
If SuRe = vbNo Then Exit Sub  
  
D1.Close
```



```
SetAttr Folder & "db1.mdb", vbNormal  
Kill Folder & "db1.mdb"
```

```
FileCopy CommonDialog1.FileName, Folder & "db1.mdb"
```

```
Set D1 = DBEngine.Workspaces(0).OpenDatabase(Folder  
& "Db1.mdb", False, False, ";pwd=" & PassWord)
```

```
Set T1 = D1.OpenRecordset("Tb_Product", dbOpenTable)
```

```
Set T2 = D1.OpenRecordset("Tb_Category",  
dbOpenTable)
```

```
Set T3 = D1.OpenRecordset("Tb_Factory", dbOpenTable)
```

```
Set T5 = D1.OpenRecordset("Tb_Sel_Bay",  
dbOpenTable)
```

```
Set T4 = D1.OpenRecordset("Tb_User", dbOpenTable)
```

```
Info(0) = T4!User_Name
```

```
Info(1) = T4!PassWord
```

```
Info(2) = T4!Name
```

```
Info(3) = T4!Telephone
```

```
Info(4) = T4!More
```

```
T4.Close
```

```
Call Start
```

```
MsgBox & "تم استيراد قاعدة البيانات بنجاح من :
```

```
CommonDialog1.FileName & Chr(13) & "سوف يتم الآن بدء تشغيل هذه
```

```
، "نسخ واستيراد"vbInformation + arabic"القاعدة"،
```

```
: ١٢
```

```
Exit Sub
```

```
End Sub
```

ملف مرفق :

قم بمطالعة الملف prog17.zip في المرفقات لمزيد من التفاصيل .

البحث عن البضائع

سنحاول بإذن الله خلال هذه الدروس اتمام عملية البحث الموجودة في القائمة ، وسوف نقوم بجعل المستخدم يتحكم في كامل شروط البحث .

نحتاج في هذه الشاشة إلى زر للبحث ، زر لغلاق الشاشة ، أداة جدول لعرض النتائج ، مربع نص لكتابة قيمة البحث .

اضافة لذلك نحتاج إلى Combo1 لنحدد هل نريد البحث في الاسم ، المصنع ، الكمية ... الخ ، وأيضاً Combo2 لنعرف شرط البحث - في الحالة الرقمية - أكبر من ، أصغر من ، = . سنجعل هذا الفورم MDICHild ، وفي أمر اظهاره في الفورم الرئيسي سنكتب :

```
If T1.RecordCount = 0 Then
    MsgBox "لا توجد أي بضائع للبحث عنها ، لاضافة بضاعة جديدة اختر اضافة - اضافة" vbExclamation + arabic ،
    Exit Sub
End If

Frm_Find.Show
```

والآن سننشأ حدث Create_Flex والذي ستكون مهمته تكوين الجدول ، وهو نفس الكود الذي قمنا بكتابته في Frm_Show دون أي تغيير ، اللهم ما عدا أن السطر التالي :

```
MSFlexGrid1.Rows = T4.RecordCount + 1
```

سيتم تحويل إلى :

```
On Error Resume Next
MSFlexGrid1.Rows = T4.RecordCount + 1
If Err = 3420 Then
    MSFlexGrid1.Rows = 2
End If
```

وسبب هذا أننا سنستدعي الأمر في حالة كون القاعدة مغلقة مما يسبب الخطأ رقم ٣٤٢٠ ، وهذا الكود لعدم اظهار رسالة خطأ .
وبهذا يصبح الكود الكامل :

```
Sub Create_Flex()
MSFlexGrid1.Clear
MSFlexGrid1.Cols = 8
On Error Resume Next
MSFlexGrid1.Rows = T4.RecordCount + 1
If Err = 3420 Then
    MSFlexGrid1.Rows = 2
End If
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 0
MSFlexGrid1.Text"ت" =
MSFlexGrid1.Col = 1
MSFlexGrid1.Text"رقم" =
MSFlexGrid1.Col = 2
MSFlexGrid1.Text"اسم البضاعة" =
MSFlexGrid1.Col = 3
MSFlexGrid1.Text"النوع" =
MSFlexGrid1.Col = 4
MSFlexGrid1.Text"المصنع" =
MSFlexGrid1.Col = 5
MSFlexGrid1.Text"السعر" =
```

```
MSFlexGrid1.Col = 6
MSFlexGrid1.Text"العدد" =
MSFlexGrid1.Col = 7
MSFlexGrid1.Text"وحدة/صندوق" =
```

```
MSFlexGrid1.ColAlignment(0) = 5
MSFlexGrid1.ColAlignment(1) = 5
MSFlexGrid1.ColAlignment(2) = 3
MSFlexGrid1.ColAlignment(3) = 5
MSFlexGrid1.ColAlignment(4) = 5
MSFlexGrid1.ColAlignment(5) = 5
MSFlexGrid1.ColAlignment(6) = 5
MSFlexGrid1.ColAlignment(7) = 5
```

```
MSFlexGrid1.ColWidth(0) = 500
MSFlexGrid1.ColWidth(1) = 500
MSFlexGrid1.ColWidth(2) = 1500
MSFlexGrid1.ColWidth(3) = 1500
MSFlexGrid1.ColWidth(4) = 1500
End Sub
```

: Form_Load من أجل هذا سنكتب في حدث

```
Me.Width = 5625
Me.Height = 5400
Me.Left = ((MDIForm1.Width - Me.Width) / 2)
Me.Top = ((MDIForm1.Height - Me.Height) / 2) - 40
Me.Show

Create_Flex
```

ومن ثم نضيف العناصر المناسبة إلى Combo1 و Combo2 :

```
Combo1.AddItem "اسم البضاعة"  
Combo1.AddItem "النوع"  
Combo1.AddItem "المورد - المصنع"  
Combo1.AddItem "الكمية"  
Combo1.AddItem "السعر"
```

```
Combo2.AddItem ">"  
Combo2.AddItem "<"  
Combo2.AddItem "="
```

```
Combo1.ListIndex = 0
```

والآن لنرَ أمراً ما ، ففي حالة اختيار المستخدم البحث عن اسم البضاعة أو المصنع أو النوع فليس ثم خيار من Combo2 إلا (=) لأن البيانات في هذه الحالة نصية فلن نستخدم < و > .
من أجل هذا نكتب الأمر التالي في حدث Combo1_Click :

```
If Combo1.ListIndex <> 3 And Combo1.ListIndex <> 4  
Then  
    Combo2.ListIndex = 2  
    Combo2.Enabled = False  
Else  
    Combo2.Enabled = True  
End If
```

الحدث الثاني الذي سننشأه هو حدث Refresh_Me ، ولن نغير في الحدث الموجود في Frm_Show إلا نقطتين رئيسيتين :

- * النقطة الأولى : سنزيل كل الأوامر التي كانت مخصصة لتعبئة القائمة بالبيانات .
- * النقطة الثانية : سنغير جملة الاستعلام .

لو نتذكر أمر الاستعلام السابق :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number"  
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
```

سنزيله ، ونستدعي بدلاً منه حدثاً اسمه **Search** تكون مهمته تكوين جملة الاستعلام ، ومن أجل ذلك يصبح كود الحدث بالشكل التالي :

```
Public Sub Refresh_Me()  
Call Search  
  
Create_Flex  
If T4.RecordCount = 0 Then Exit Sub  
  
Frm_Wait.Show  
Frm_Wait.ProgressBar1.Max = T1.RecordCount  
Frm_Wait.ProgressBar1.Min = 0  
Frm_Wait.ProgressBar1.Value = 0  
  
T4.MoveLast  
T4.MoveFirst  
Create_Flex  
  
If T4.RecordCount <> 0 Then  
T4.MoveFirst  
For I = 0 To T4.RecordCount - 1  
Frm_Wait.ProgressBar1.Value =  
Frm_Wait.ProgressBar1.Value + 1  
DoEvents  
MSFlexGrid1.Row = I + 1
```

```
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = T4.Fields("tb_product.Number")
MSFlexGrid1.Col = 2
MSFlexGrid1.Text = T4.Fields("tb_product.name")
MSFlexGrid1.Col = 3
MSFlexGrid1.Text = T4.Fields("tb_category.name")
MSFlexGrid1.Col = 4
MSFlexGrid1.Text = T4.Fields("tb_factory.name")
MSFlexGrid1.Col = 5
MSFlexGrid1.Text = T4!price
MSFlexGrid1.Col = 6
MSFlexGrid1.Text = T4.Fields("Count")
MSFlexGrid1.Col = 7
MSFlexGrid1.Text = T4!Box_count
```

```
T4.MoveNext
Next I
```

```
MSFlexGrid1.Col = 2
MSFlexGrid1.Sort = 7
```

```
MSFlexGrid1.Col = 0
For I = 0 To T4.RecordCount - 1
    MSFlexGrid1.Row = I + 1
    MSFlexGrid1.Text = I + 1
Next I
End If
```

```
T4.Close
```

```
Unload Frm_Wait
End Sub
```

والآن سنرى كيف نقوم بعمل أمر Search .

في البداية تعتمد جملة الاستعلام على المحدد من Combo1 لذا سنقوم بعمل جملة Select بالشكل التالي :

```
Select Case Combo1.ListIndex
Case 0

Case 1
Case 2
Case 3
Case 4

End Select
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
```

وجميع جمل الاستعلام ستنبى على الجملة الاساسية التي كانت بالشكل التالي :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from
tb_product,tb_factory,tb_category where
tb_product.category=tb_category.number and
tb_product.factory=tb_factory.number"
```

إلا أننا سنضيف شرط البحث في آخر هذه الجملة .

من أجل ذلك تكون جملة البحث عن Name أي الحالة الأولى :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from
tb_product,tb_factory,tb_category where
tb_product.category=tb_category.number and
tb_product.factory=tb_factory.number and
tb_product.name='" & Text1.Text"'" &
```


* لاحظ أننا لا بد أن نكتب tb_product.name = لأن الاسم الذي نبحث عنه هو اسم البضاعة من Tb_Product .

والحالة الثانية أي البحث عن النوع :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number and  
tb_category.name="" & Text1.Text"" &
```

* لاحظ أننا هنا بحثنا عن Name ولكنه من جدول الأنواع .

والحالة الثالثة أي حالة البحث عن المصنع :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number and  
tb_factory.name="" & Text1.Text"" &
```

أما الحالتين الأخريتين فليس الشرط (=) هو الموجود دائماً ، بل إنه يمكن أن يكون < أو > ، وهذا يعتمد على قيمة Combo2 ، ولهذا تكون جملة الاستعلام في الحالة الرابعة :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number and count" &  
Combo2.Text & Val(Text1.Text
```

وفي الحالة الأخيرة تكون جملة الاستعلام :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number and price" &  
Combo2.Text & Val(Text1.Text(
```

ملف مرفق :

قم بمطالعة الملف prog18.zip في المرفقات لمزيد من التفاصيل .

تقارير البضائع

سنبدأ بإذن الله في هذا الدرس في عمل قائمة التقارير وسنقوم تحديداً بعمل تقارير البضائع ، وسنحاول جاهدين من خلال هذا الدرس تطبيق مفهوم المصفوفات (مصفوفات الأدوات) والتي نتعامل معها من خلال خاصية Index وسننوه كلما سنحت الفرصة عن كمية الأكواد التي تم اختصارها اعتماداً على المصفوفات .

وسنبدأ في البداية بتصميم الفورم .

فعن محتويات الفورم فنحتاج إلى قائمة نعرض فيها البضائع ، وأداتي اختيار لتحديد نوع التقرير (شامل أو بين تاريخين) ، وبناء عليه سنضع اثنين من أداة MaskedTextBox . سنضع أيضاً زر (تنفيذ) لتنفيذ التقرير ، وزر اغلاق للفورم ، كما سنضع أداتي MsFlexGeid ولكن لهما نفس الاسم ، ولها Index يساوي ٠ أو ١ . وتختص بعرض عمليات البيع والشراء لهذه البضاعة .

ونضع أيضاً خمس مربعات عناوين Labels لنضع فيها معلومات حول البضاعة المختارة في التقرير (الاسم - المصنع - النوع - الكمية - السعر) .

فيما سؤجل عمليات الاحصائيات إلى درس لاحق إن شاء الله .
والآن انتهى التصميم ، ولنبدء في كتابة الكود .
في الفورم الرئيسي لن يكون هناك سوى أمر التأكد من وجود بضائع ، ومن ثم معرفة Index القائمة
وبناء عليه نظهر الفورم وذلك بالشكل التالي :

```
If T1.RecordCount = 0 Then
    MsgBox "لا توجد أي بضائع للبحث عنها ، لاضافة بضاعة جديدة اختر اضافة - اضافة" vbExclamation + arabic, "عرض البضائع"
    Exit Sub
End If
Select Case Index
Case 0
    Frm_Report1.Show
End Select
```

أما في حدث **Form_Load** للفورم الخاص بالتقرير فسنضبط أولاً خصائص الارتفاع :

```
Me.Width = 4800
Me.Height = 6615
Me.Left = ((MDIForm1.Width - Me.Width) / 2)
Me.Top = ((MDIForm1.Height - Me.Height) / 2) - 40
Me.Show
```

ثم نحدد تاريخ اليوم لنضعه في مربعي **MaskedTextBox** :

```
Dim Myday, Mymonth, Myyear
Myday = Day(Now)
If Len(Myday) = 1 Then Myday = 0 & Myday
Mymonth = Month(Now)
If Len(Mymonth) = 1 Then Mymonth = 0 & Mymonth
```

```
Myyear = Year(Now)  
MaskedTextBox1.Text = Myday & "/" & Mymonth & "/" &  
Myyear  
MaskedTextBox2.Text = MaskedTextBox1.Text
```

ومن ثم نملئ القائمة بالبضائع الحالية :

```
Combo1.Clear  
T1.MoveFirst  
For i = 0 To T1.RecordCount - 1  
    Combo1.AddItem T1!Name  
    T1.MoveNext  
Next i
```

وأخيراً نجعل **Option1** محدداً – لم نضبطه وقت التصميم لغرض ما سنشرحه لاحقاً – ، ومن ثم نحدد العنصر الأول في القائمة ، ونطلق زر التنفيذ ، حيث أن تقريراً عن البضاعة الأولى سيكون هو الموجود بصورة افتراضية :

```
Option1.Value = True  
Combo1.ListIndex = 0  
Command2.Value = True
```

والآن لنر لماذا ضبطنا **Option1** وقت التصميم ، ذلك أننا نريد في حالة اراد المستخدم تقريراً شاملاً أن لا نعطيه التحكم في مربعي التاريخ ، ونقوم بذلك عبر خاصية **Click** لأزرار الاختيار بالشكل التالي :

```
Private Sub Option1_Click()  
MaskedTextBox1.Enabled = False  
MaskedTextBox2.Enabled = False  
End Sub
```

```
Private Sub Option2_Click()
MaskedTextBox1.Enabled = True
MaskedTextBox2.Enabled = True
End Sub
```

*** ويمكن أيضاً الاستعاضة عن **OptionButton** باستخدام **CheckBox** .

والآن سنضيف متغيران لمعرفة عدد سجلات البيع والشراء ، ولأننا نتعامل مع مصفوفة فسنكتب الأمر التالي في قسم التصاريح **Genral** :

```
Dim MyCnt(1
```

مما يعني أن لدينا متغيرات **MyCnt(0** و **MyCnt(1**،

والآن سنبرمج أوامر **Command2** وهو زر تنفيذ التقرير .

في البداية سنقوم بعمل استعلام عن اسم هذه البضاعة لمعرفة المعلومات المطلوبة عنها ووضعها في أماكنها

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from
tb_product,tb_factory,tb_category where
tb_product.category=tb_category.number and
tb_product.factory=tb_factory.number and
tb_product.name='" & Combo1.Text'" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
```

```
Lbl_name & " = اسم البضاعة : " = T4.Fields("tb_product.name"
```

```
lbl_cate & " = النوع : " = T4.Fields("tb_category.name"
```

```
lbl_fact & " = المصنع : " = T4.Fields("tb_factory.name"
```

```

lbl_prc & " : السعر " = T4.Fields("price")
lbl_cnt & " : الكمية الموجودة " = T4.Fields("count")

Dim num, i
num = T4.Fields("tb_product.Number")
T4.Close

```

وقد سبق شرح جملة الاستعلام أكثر من مرة .

الجزء الثاني : سنقوم بالاستعلام حسب قيمة Option1 (بين تاريخين أو عام) وباستخدام رقم البضاعة المخزن في المتغير Num وذلك عن الـ Kind مرة عندما يساوي الصفر ومرة عند الواحد ... مما يمكننا من جعلها في حلقة تكرار ... لاحظ الكود التالي :

```

For i = 0 To 1
If Option1.Value = True Then
    SQL = "select * from tb_sel_bay where product=" & num
    & " and kind=" & i
Else
    SQL = "select * from tb_sel_bay where product=" & num
    & " and kind=" & i & " and date between#" &
    Format(MaskedTextBox1.Text, "MM/DD/YYYY") & "# and #" &
    Format(MaskedTextBox2.Text, "MM/DD/YYYY" & "#" & "
End If
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)

```

حيث يأخذ i قيمة ما بين ٠ و ١ .

لن نغلق حلقة التكرار فما زلنا في حاجة إليها ، وعلينا أولاً معرفة عدد السجلات الموجودة في الاستعلام وتخزينها في MyCnt رقم ٠ أو ١ حسب قيمة i وذلك بالشكل التالي :

```

If T4.RecordCount <> 0 Then
    T4.MoveLast
    T4.MoveFirst
End If
MyCnt(i) = T4.RecordCount

```

والآن علينا عمل اجراءين ، اجراء خاص بتكوين الجدول ، واجراء خاص بوضع البيانات فيها وبدلاً من عمل اجراءات منفصلة لكل جدول فسنعتمد مجدداً على قيمة ٠ أو ١ ، وسنعد اجراء اسمه FlexGrid لتكوين الجدول ويستقبل قيمة المتغير ليضبط الجدول المحدد . لاحظ كود الاجراء :

```

Private Sub FlexGrid(r As Integer)
MSFlexGrid(r).Clear
MSFlexGrid(r).Cols = 5
MSFlexGrid(r).Rows = MyCnt(r) + 1

MSFlexGrid(r).Row = 0
MSFlexGrid(r).Col = 0
MSFlexGrid(r).Text"ت" =
MSFlexGrid(r).Col = 1
MSFlexGrid(r).Text"رقم" =
MSFlexGrid(r).Col = 2
MSFlexGrid(r).Text"التاريخ" =
MSFlexGrid(r).Col = 3
MSFlexGrid(r).Text"الكمية" =
MSFlexGrid(r).Col = 4
MSFlexGrid(r).Text"السعر" =

MSFlexGrid(r).ColAlignment(0) = 5
MSFlexGrid(r).ColAlignment(1) = 5
MSFlexGrid(r).ColAlignment(2) = 5
MSFlexGrid(r).ColAlignment(3) = 5
MSFlexGrid(r).ColAlignment(4) = 5

```

```
MSFlexGrid(r).ColWidth(0) = 500
MSFlexGrid(r).ColWidth(1) = 500
MSFlexGrid(r).ColWidth(3) = 1200
End Sub
```

وسنستدعيه بالشكل التالي :

```
FlexGrid (i
```

وينفس الطريقة كود Put_Data لوضع البيانات :

```
Private Sub Put_Data(r As Integer)
n = 0
With MSFlexGrid(r)
For i = 0 To T4.RecordCount - 1
    n = n + 1
    . Row = n
    . Col = 0
    . Text = n
    . Col = 1
    . Text = T4!Number
    . Col = 2
    . Text = T4!Date
    . Col = 3
    . Text = T4!Count
    . Col = 4
    . Text = T4!price
    T4.MoveNext
Next i
End With
End Sub
```


ونستدعيه ، ثم نغلق القاعدة ، ونغلق حلقة التكرار :

```
Put_Data (i,  
T4.Close  
Next i
```

أخيراً لا تنس كود اغلاق الفورم :

```
Private Sub Command1_Click()  
Unload Me  
End Sub
```

ملف مرفق :

قم بمطالعة الملف prog19.zip في المرفقات لمزيد من التفاصيل .

تقارير المصانع والأنواع

هناك الكثير مما يمكن أن تقدمه كتقرير لنوع أو مصنع ، لكننا لن نتوسع كثيراً بل كل ما سوف نعرضه هو منتجات المصنع والنوع .

كل ما نحتاج إليه لتصميم هذا الفورم هو :

أداة مربع تحرير وسرد **Combo** ، وثلاث مربعات عناوين **Label** لعرض اسم النوع أو المصنع ، عدد البضائع تحته ، وحالته (محذوف - غير محذوف) .

نحتاج أيضاً في كلا الفورمين إلى جدول **MSFlexGrid** لعرض فيه البضائع تحت هذا المصنع أو ذاك النوع ، وأخيراً زر أمر نكتب فيها **Unload Me** وذلك للاغلاق .

ولنضبط خاصية **MDIChild** لكلا الفورمين بحيث تصبح **True** ، ونسميهما

Frm_Report2 و **Frm_Report3** على التوالي .

والآن سنبدأ في كتابة الأكواد ، ونبدأ بحدث `Form_Load` لكلا الفورمين .
في هذا الحدث سوف نضبط احداثيات الموقع كما تعودنا مع كل فورم ، ومن ثم نعبئ القائمة بالموردين
أو بالمصانع :

```
Me.Width = 4800
Me.Height = 3600
Me.Left = ((MDIForm1.Width - Me.Width) / 2)
Me.Top = ((MDIForm1.Height - Me.Height) / 2) - 40
Me.Show

Combo1.Clear
T2.MoveFirst
For i = 0 To T2.RecordCount - 1
    Combo1.AddItem T2!Name
    T2.MoveNext
Next i

Combo1.ListIndex = 0
```

مع ملاحظة أنه في الفورم الثالث الخاص بالمصانع سوف نستخدم `T3` بدلاً من `T2` .
والآن سنقوم بعمل حدث `Flex` خاص بتكوين الجدول ، والفرق بين الفورمين في هذا التكوين أننا في
الأول نضع معلومة (مصنع البضاعة) دون النوع ، والعكس في الفورم الثاني :

```
Private Sub Flex()
MSFlexGrid1.Clear
MSFlexGrid1.Cols = 6
MSFlexGrid1.Rows = T4.RecordCount + 1
MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 0
MSFlexGrid1.Text = "ت"
MSFlexGrid1.Col = 1
MSFlexGrid1.Text = "رقم"
```

```
MSFlexGrid1.Col = 2
MSFlexGrid1.Text"اسم البضاعة" =
MSFlexGrid1.Col = 3
MSFlexGrid1.Text"المصنع" =
MSFlexGrid1.Col = 4
MSFlexGrid1.Text"السعر" =
MSFlexGrid1.Col = 5
MSFlexGrid1.Text"العدد" =
```

```
MSFlexGrid1.ColAlignment(0) = 5
MSFlexGrid1.ColAlignment(1) = 5
MSFlexGrid1.ColAlignment(2) = 3
MSFlexGrid1.ColAlignment(3) = 5
MSFlexGrid1.ColAlignment(4) = 5
MSFlexGrid1.ColAlignment(5) = 5
```

```
MSFlexGrid1.ColWidth(0) = 500
MSFlexGrid1.ColWidth(1) = 500
MSFlexGrid1.ColWidth(2) = 1500
MSFlexGrid1.ColWidth(3) = 1500
End Sub
```

وأخيراً نقوم بعمل الحدث Combo1_Click والذي يطلق عند اختيار المستخدم لنوع أو مصنع ما .

في البداية نقوم بعمل جملة الاستعلام والتي ستكون في حالة المصانع :

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from
tb_product,tb_factory,tb_category where
tb_product.category=tb_category.number and
tb_product.factory=tb_factory.number and
tb_factory.name='" & Combo1.Text"'" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,
```

وسنغير شرط البحث من Tb_Factory.name إلى Tb_Category في حالة الأنواع .
والآن سنختبر شرط وجود سجلات ، فإن لم يكن هناك أي بضائع تحت هذا النوع-المصنع :

```
If T4.RecordCount = 0 Then
    Lbl_name & " : اسم المصنع " = Combo1.Text
    lbl_cnt & " : عدد بضائع المصنع " = T4.RecordCount
    lbl_sta = "الحالة : عادي"
    Flex
    Exit Sub
End If
```

بالرغم من عدم وجود سجلات سنضبط بعض الخصائص ، كما سنستدعي الحدث Flex والذي يكون الجدول دون أي صفوف .
لاحقاً لن يكتمل التنفيذ إلا في حالة وجود سجلات ، حيث نقوم بالانتقال للسجل الأخير ثم إلى الأول وذلك للتحديث :

```
T4.MoveLast
T4.MoveFirst
```

ثم نضبط مجموعة من المعلومات المتعلقة بهذا النوع ... لاحظ الأمر التالي :

```
Lbl_name & " : اسم المصنع " = Combo1.Text
lbl_cnt & " : عدد بضائع المصنع " = T4.RecordCount
If T4.Fields("tb_category.view") = False Then
    lbl_sta = "الحالة : محذوف"
Else
    lbl_sta = "الحالة : عادي"
End If
Flex
```

والآن ... لماذا في حالة وجود بضائع تحت النوع أو المصنع تأكدنا من الحالة (محذوف أم غير ذلك) ، أما في حالة عدم وجود بضائع فإننا نضع مباشرة (عادي) ؟؟؟ ذلك انه لو كان محذوف ولا توجد تحته بضائع لكان قد تم حذفه سابقاً .

بعد ذلك نقوم باظهار فورم الانتظار ونضبط خصائصه كما أسلفنا شرحه :

```
Frm_Wait.Show  
Frm_Wait.ProgressBar1.Max = T4.RecordCount  
Frm_Wait.ProgressBar1.Min = 0  
Frm_Wait.ProgressBar1.Value = 0
```

ومن ثم نبدأ في وضع البيانات في أماكنها الصحيحة :

```
For i = 0 To T4.RecordCount - 1  
    Frm_Wait.ProgressBar1.Value =  
Frm_Wait.ProgressBar1.Value + 1  
    DoEvents  
    MSFlexGrid1.Row = i + 1  
    MSFlexGrid1.Col = 1  
    MSFlexGrid1.Text = T4.Fields("tb_product.Number")  
    MSFlexGrid1.Col = 2  
    MSFlexGrid1.Text = T4.Fields("tb_product.name")  
    MSFlexGrid1.Col = 3  
    MSFlexGrid1.Text = T4.Fields("tb_category.name")  
    MSFlexGrid1.Col = 4  
    MSFlexGrid1.Text = T4!price  
    MSFlexGrid1.Col = 5  
    MSFlexGrid1.Text = T4.Fields("Count")  
    T4.MoveNext  
Next i
```

لاحقاً نقوم بعمل Sort ترتيب للجدول مع اعادة الترتيب كما سبق :

```
MSFlexGrid1.Col = 2  
MSFlexGrid1.Sort = 7  
MSFlexGrid1.Col = 0
```

```
For i = 0 To T4.RecordCount - 1  
    MSFlexGrid1.Row = i + 1  
    MSFlexGrid1.Text = i + 1  
Next i
```

أخيراً نغلق قاعدة البيانات ، ونخفي فورم الانتظار :

```
T4.Close  
Unload Frm_Wait
```

ملف مرفق :

قم بمطالعة الملف prog20.zip في المرفقات لمزيد من التفاصيل .

الإحصائيات والرسوم البيانية

إن مستخدماً لبرنامج حسابي يتوقع منك من خلال تقنية التقارير أن تقدم له دعماً ولو محدوداً من خلال بعض الاحصائيات والمخططات البيانية ..

ففي برنامج ما لشركة توزيع وتسويق البضائع التجارية المختلفة ، يتوقع منك مستخدم البرنامج أن تقدم له تقريراً بأرباح بضاعة معينة ، أو تخطيطاً لمبيعات نوع من البضاعة بحيث يستطيع أن يدرك موضعه من السوق ، وتنامي الربح لهذه البضاعة من عدمه .

وهكذا ، وبما أن برنامجنا لا يعدو أن يكون برنامجاً لسوبر ماركت فقط فلن يتوقع المستخدم الكثير من المخططات أو الاحصائيات ، ولكن لتطبيق المفهوم سوف نتعلم سوية كيفية عمل مخطط بياني لنسب مبيعات ومشتريات البضاعة من ضمن المشتريات . وسنضيفها ضمن شاشة تقارير البضائع ... وسنبدأ سوية في طريقة عامة لعمل المخططات البيانية .

قم بالنقر على **Ctrl+T** لعرض الأدوات ، ومنها قم باختيار الأداة : **Microsoft Chart Control 6.0(OLEDB)**

والآن قم برسمها على الفورم كما يحلو لك ، وسنحتاج إلى ضبط بعض الخصائص .
خاصية **Charttype** : أحد أهم خصائص الأداة ، وذلك أنها تحدد شكل المخطط البياني (ثلاثي أو ثنائي الأبعاد - دائري أو اعتيادي أو X,Y ...) وهكذا .
وسنختار **VtchChartType2dPie**
وآخر ثلاثة أحرف من الأسم للنوع وقبلها حرفان لتحديد (ثلاثي أم ثنائي الأبعاد) .

خاصيتا **Row** و **Column** وكل ما يتعلق بهما .

تحدد خاصية **Row** عدد الصفوف ، أي عدد السجلات أو التقارير المطلوبة ، فمثلاً لو كنا نريد أن نعرض في نفس الوقت أكثر من بضاعة للمقارنة فإننا نضبط خاصية **RowCount** بعدد البضائع ، أم الآن فسنبسطها بحيث تساوي ١ .

أما **Column** فتحدد عدد النقاط أو البيانات ، وفي مثالنا هذا لدينا أربع بيانات (اجمالي المبيعات - اجمالي المشتريات - اجمالي مبيعات البضاعة - اجمالي مشتريات البضاعة) . لذا نضبط
ColumnCount=4 .

وسنضبط باقي الخصائص وقت التصميم .

والآن نحتاج إلى أربع قيم نخزنها في مصفوفة **Total(3)** بالترتيب ، وسنضع هذه الأوامر في آخر زر التنفيذ . فالأولى هو العدد الكلي للمبيعات ، لذا ننفذ جملة استعلام على المدة المحددة ونخزن البيانات في العنصر الأول للمصفوفة :

وسنستعلم عن مجموع العمود الذي يحقق الشرط ، وعدد السجلات فيه لمنع حدوث خطأ **NULL** .

```

If Option1.Value = True Then
    SQL = "select sum(count) as sum1,count(product) as
cnt from tb_sel_bay where kind=0"
Else
    SQL = "select sum(count) as sum1,count(product) as
cnt from tb_sel_bay where kind=0 " & " and date
between#" & Format(MaskedTextBox1.Text, "MM/DD/YYYY")
& "# and #" & Format(MaskedTextBox2.Text, "MM/DD/YYYY ("
"#" &
End If
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4!cnt <> 0 Then
    Total(0) = T4!sum1
End If
T4.Close

```

لاحظ أننا تأكدنا من عدد السجلات مسبقاً لتلافي خطأ Null ، جرب ازالته ولاحظ الفرق .
والثاني العدد الكلي للمشتريات ، وهو مثل السابق مع اختلاف Kind في جملة الاستعلام :

```

If Option1.Value = True Then
    SQL = "select sum(count) as sum1,count(product) as
cnt from tb_sel_bay where kind=1"
Else
    SQL = "select sum(count) as sum1,count(product) as
cnt from tb_sel_bay where kind=1 " & " and date
between#" & Format(MaskedTextBox1.Text, "MM/DD/YYYY")
& "# and #" & Format(MaskedTextBox2.Text, "MM/DD/YYYY ("
"#" &
End If
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4!cnt <> 0 Then
    Total(1) = T4!sum1
End If
T4.Close

```


أما البيان الثالث فهو مجموع مبيعات البضاعة ، ونستطيع الحصول عليها من الجدول دون أن نضطر إلى جملة استعلام أخرى :

```
MSFlexGrid(0).Col = 3
For i = 1 To MSFlexGrid(0).Rows - 1
    MSFlexGrid(0).Row = i
    Total(2) = Total(2) + MSFlexGrid(0).Text
Next i
```

ونفس الأمر بالنسبة لمشتريات البضاعة :

```
MSFlexGrid(1).Col = 3
For i = 1 To MSFlexGrid(1).Rows - 1
    MSFlexGrid(1).Row = i
    Total(3) = Total(3) + MSFlexGrid(1).Text
Next i
```

والآن جاء دور وضع البيانات ، وبما أننا نريد الاستفادة من خاصية Index سنضيف النصوص التوضيحية في مصفوفة هي الأخرى :

```
Dim str(3) As String
str(0) = "اجمالي مبيعات"
str(1) = "اجمالي مشتريات"
str(2) = "اجمالي مبيعات بضاعة"
str(3) = "اجمالي مشتريات بضاعة"
```

ونضبط خصائص الجدول كما أسلفنا :

```
MSChart2.ColumnCount = 4
MSChart2.RowCount = 1
```

ونبدأ في وضع البيانات بحلقة تكرار ، مع مراعاة أننا نضع البيان بواسطة الخاصية Date وننتقل للعمود من خلال الخاصية Column ونضع النص التوضيحي من خلال ColumnLabel :

```
For i = 1 To 4
    MSChart2.Column = i
    MSChart2.Data = Total(i - 1)
    MSChart2.ColumnLabel = str(i - 1)
Next i
```

ملف مرفق :

قم بمطالعة الملف prog21.zip في المرفقات لمزيد من التفاصيل .

التقارير اليومية

في هذا الدرس سوف نتعرف على كيفية عمل نموذج مبسط للتقارير اليومية .
ما نحتاج إليه في هذه الشاشة هو أداة MaskedTextBox للتاريخ ، والذي سيكون افتراضياً تاريخ اليوم .

نحتاج أيضاً إلى زر للتنفيذ ، وإلى جدولين لعرض المبيعات والمشتريات ، وزر لغلاق الفورم .
أيضاً نحتاج إلى خمسة Labels على النحو التالي :
اليوم من الأسبوع : عدد عمليات المبيعات - عدد عمليات المشتريات - إجمالي سعر المبيعات - إجمالي سعر المشتريات .

وأخيراً نضبط خصائص الفورم ، فنجعل MDIChild=True ونزيل عناصر التحكم من أعلى الفورم .

والآن نبدأ في كتابة أكواد هذا الفورم .

فأولاً نضبط احدثيات الفورم كما اعتدنا مع كل فورم من هذا النوع :

```
Me.Width = 4800
Me.Height = 4905
Me.Left = ((MDIForm1.Width - Me.Width) / 2)
Me.Top = ((MDIForm1.Height - Me.Height) / 2) - 40
Me.Show
```

ثم نضبط تاريخ اليوم ونضعه في الأداة :

```
Dim Myday, Mymonth, Myyear
Myday = Day(Now)
If Len(Myday) = 1 Then Myday = 0 & Myday
Mymonth = Month(Now)
If Len(Mymonth) = 1 Then Mymonth = 0 & Mymonth
Myyear = Year(Now)
MaskedTextBox1.Text = Myday & "/" & Mymonth & "/" &
Myyear
```

وأخيراً نقوم بالضغط على الزر الخاص بالتنفيذ ليكون افتراضياً تاريخ اليوم :

```
Command2.Value = True
```

والآن سنقوم بعمل دالة Function نقوم بتمرير رقم اليوم لها وتعطينا اسم اليوم ، ونستخدم في ذلك الدالة WeekDay ، ونبدأ أولاً بتكوين الدالة :

```
Private Function dayname(x As Date) As String

End Function
```

والآن أصبح لدينا التاريخ في متغير X وسنستخدم معه الدالة WeekDay والتي نمرر لها بارميترين (التاريخ ، ويوم بداية الأسبوع) وتعيد رقم اليوم من الأسبوع :

```
x = Weekday(x, vbSaturday)
```

والآن سنحول هذا الرقم إلى نص بالشكل التالي :

```
Select Case x
Case 1
    dayname = "السبت"
Case 2
    dayname = "الأحد"
Case 3
    dayname = "الاثنين"
Case 4
    dayname = "الثلاثاء"
Case 5
    dayname = "الأربعاء"
Case 6
    dayname = "الخميس"
Case 7
    dayname = "الجمعة"
End Select
```

والآن نبدأ في تكوين زر **Command2** والخاص بـ (نفذ) .

نعرف أولاً أربع متغيرات لنخزن فيها معلومات ال Labels وذلك في قسم التصاريح Genral :

```
Dim a1 As Integer, a2 As Integer, a3 As Integer, a4 As Integer
```

وفي بداية أوامر هذا الزر نقوم بتصغيرها كلها :

```
a1 = 0: a2 = 0: a3 = 0: a4 = 0
```

ونستدعي الدالة التي قمنا بعملها ونضعها في مكانها المخصص :

```
x = dayname(MaskedTextBox1.Text)
Lbl_day.Caption & " = X اليوم من الأسبوع :
```

بعد ذلك نقوم بالاستعلام عن عمليات اليوم ، البيع ثم الشراء ، ونضع القيم المستقاه بواسطة دوال الاستعلام في المتغيرات التي قمنا بتعريفها :

```
SQL = "select sum(price) as prc,count(*) as cnt from
tb_sel_bay where kind=0 and date=#" &
Format(MaskedTextBox1.Text, "MM/DD/YYYY" & "/" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
If T4!cnt <> 0 Then
    a1 = T4!cnt
    a2 = T4!prc
End If
T4.Close
```

```
SQL = "select sum(price) as prc,count(*) as cnt from
tb_sel_bay where kind=1 and date=#" &
Format(MaskedTextBox1.Text, "MM/DD/YYYY" & "/" &
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset)
```

```
If T4!cnt <> 0 Then
    a3 = T4!cnt
    a4 = T4!prc
End If
T4.Close
```

```

lbl_cnt1.Caption & " = عدد مبيعات اليوم : a1
lbl_prc1.Caption & " = إجمالي سعر المبيعات : a2
lbl_cnt2.Caption & " = عدد مشتريات اليوم : a3
lbl_prc2.Caption & " = إجمالي سعر المشتريات : a4

```

والآن نقوم ببناء الجدولين ، ولاحظ أنه يمكننا الاستفادة من خاصية **Index** كما في الشاشات السابقة :

```

Flex1
Flex2

```

واللذان قمنا بعملهما سابقاً بالشكل التالي :

```

Private Sub Flex1()
MSFlexGrid1.Clear
MSFlexGrid1.Cols = 5
MSFlexGrid1.Rows = a1 + 1

MSFlexGrid1.Row = 0
MSFlexGrid1.Col = 0
MSFlexGrid1.Text"ت" =
MSFlexGrid1.Col = 1
MSFlexGrid1.Text"رقم" =
MSFlexGrid1.Col = 2
MSFlexGrid1.Text"البضاعة" =
MSFlexGrid1.Col = 3
MSFlexGrid1.Text"الكمية" =
MSFlexGrid1.Col = 4
MSFlexGrid1.Text"السعر" =

MSFlexGrid1.ColAlignment(0) = 5

```

```
MSFlexGrid1.ColAlignment(1) = 5
MSFlexGrid1.ColAlignment(2) = 5
MSFlexGrid1.ColAlignment(3) = 5
MSFlexGrid1.ColAlignment(4) = 5
```

```
MSFlexGrid1.ColWidth(0) = 500
MSFlexGrid1.ColWidth(1) = 500
MSFlexGrid1.ColWidth(2) = 1500
MSFlexGrid1.ColWidth(3) = 1200
End Sub
```

```
Private Sub Flex2()
MSFlexGrid2.Clear
MSFlexGrid2.Cols = 5
MSFlexGrid2.Rows = a3 + 1
```

```
MSFlexGrid2.Row = 0
MSFlexGrid2.Col = 0
MSFlexGrid2.Text"ت" =
MSFlexGrid2.Col = 1
MSFlexGrid2.Text"رقم" =
MSFlexGrid2.Col = 2
MSFlexGrid2.Text"البضاعة" =
MSFlexGrid2.Col = 3
MSFlexGrid2.Text"الكمية" =
MSFlexGrid2.Col = 4
MSFlexGrid2.Text"السعر" =
```

```
MSFlexGrid2.ColAlignment(0) = 5
MSFlexGrid2.ColAlignment(1) = 5
MSFlexGrid2.ColAlignment(2) = 5
MSFlexGrid2.ColAlignment(3) = 5
MSFlexGrid2.ColAlignment(4) = 5
```

```
MSFlexGrid2.ColWidth(0) = 500
MSFlexGrid2.ColWidth(1) = 500
MSFlexGrid2.ColWidth(2) = 1500
MSFlexGrid2.ColWidth(3) = 1200
End Sub
```

ثم نقوم بالاستعلام عن البيانات المطلوبة ، ووضعها في اماكنها المناسبة عبر استدعاء الحدثين
Put_Data1 و Put_Data2 :

```
SQL = "select * from tb_sel_bay where kind=0 and  
date=#" & Format(MaskedTextBox1.Text, "MM/DD/YYYY & ("  
"#"  
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,  
Put_Data1  
T4.Close  
  
SQL = "select * from tb_sel_bay where kind=1 and  
date=#" & Format(MaskedTextBox1.Text, "MM/DD/YYYY & ("  
"#"  
Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,  
Put_Data2  
T4.Close
```

في حدث Put_Data سنقوم بوضع البيانات في الجدول كما هي العادة ، ولكن المخزن لدينا في
الجدول هو رقم البضاعة وليس اسمها ، لذا سنقوم بالاستعلام عنه أولاً ، وسنضطر لتعرف T6 في
الموديول :

```
Public T6 As Recordset
```


بهذا يكون شكل الأمر

```
Private Sub Put_Data1()  
n = 0  
If T4.RecordCount <> 0 Then  
    T4.MoveLast  
    T4.MoveFirst  
End If  
With MSFlexGrid1  
For i = 0 To T4.RecordCount - 1  
    n = n + 1  
    . Row = n  
    . Col = 0  
    . Text = n  
    . Col = 1  
    . Text = T4!Number  
    . Col = 2  
    SQL = "select name from tb_product where number=" &  
T4!product  
    Set T6 = D1.OpenRecordset(SQL, dbOpenDynaset)  
    . Text = T6!Name  
    T6.Close  
    . Col = 3  
    . Text = T4!Count  
    . Col = 4  
    . Text = T4!price  
    T4.MoveNext  
Next i  
End With  
End Sub
```

وكذلك الثاني تماماً .

التقارير بين تاريخين .

خلال هذا الدرس سوف نقوم بتطبيق التقرير بين تاريخين ، ولأنه لا يختلف كثيراً عن التقرير اليومي لذلك سنضعهم معاً في نفس الشاشة ، وسوف نسر التغييرات التي سنقوم بها من جراء هذه العملية .

أولاً : تغييرات في التصميم :

سنضيف اداة MaskedTextBox2 لعرض التاريخ الثاني ، اضافة إلى Label2 يحتوي على كلمة (إلى) توضع بين التاريخين ، ونضبط Visible=False لهما ، فيما عدا ذلك لن نغير شيئاً حتى في أماكن الأزرار والخانات وغيرها .

ثانياً : تغييرات أوامر اظهار الشاشات :

سوف نخفي أولاً الشاشة Frm_Report4 لكي تعود إلى وضعها الطبيعي من حيث أماكن الأزرار وغيرها (وضع التقرير اليومي) .

بعد ذلك فيما لو كان النوع الثاني فإننا نظهر الاداتين السابقتين ، ونضبط احدثيات Command2 وهو زر التنفيذ من جديد ، وذلك بالشكل التالي :

Case 0

```
Unload Frm_Report4  
Frm_Report4.Show
```

Case 1

```
Unload Frm_Report4  
Frm_Report4.Show  
Frm_Report4.MaskedTextBox2.Visible = True  
Frm_Report4.Label2.Visible = True  
Frm_Report4.Command2.Left = 120  
Frm_Report4.Command2.Value = True
```

ثالثاً : تغييرات حدث Form_Load :

لن نضيف سوى أمر وضع التاريخ الحالي حتى في الخانة الثانية :

```
MaskedTextBox2.Text = MaskedTextBox1.Text
```

رابعاً : التغييرات في زر (نفذ) :

* لن نضع اليوم من الأسبوع في حالة التاريخين ، بل سنضع : مدة مفتوحة ، فيصبح الكود كالتالي :

```
If Label2.Visible = False Then
    x = dayname(MaskedTextBox1.Text)
    Lbl_day.Caption & " = X اليوم من الأسبوع : "
Else
    Lbl_day.Caption & " = اليوم من الأسبوع : " & " فترة مفتوحة "
End If
```

* لن نستعلم مباشرة بواسطة علامة المساواة بل سنستخدم العامل **Between** إذا تحقق الشرط :

```
If Label2.Visible = False Then
    SQL = "select sum(price) as prc,count(*) as cnt from
    tb_sel_bay where kind=0 and date=#" &
    Format(MaskedTextBox1.Text, "MM/DD/YYYY" & " #" & "
Else
    SQL = "select sum(price) as prc,count(*) as cnt from
    tb_sel_bay where kind=0 and date between #" &
    Format(MaskedTextBox1.Text, "MM/DD/YYYY") & " #" & " and #" &
    Format(MaskedTextBox2.Text, "MM/DD/YYYY" & " #" & "
End If
```

وكذلك مع جمل استعلام الشراء .

* في جملة الاستعلام الخاصة بوضع البيانات أيضاً سنؤكد من استخدام المساواة أو **Between** :

```
If Label2.Visible = False Then
```

```

SQL = "select * from tb_sel_bay where kind=0 and
date=#" & Format(MaskedTextBox1.Text, "MM/DD/YYYY & ("
"#")
Else
SQL = "select * from tb_sel_bay where kind=0 and date
between #" & Format(MaskedTextBox1.Text, "MM/DD/YYYY")
& "# and #" & Format(MaskedTextBox2.Text, "MM/DD/YYYY ("
"#") &
End If

```

وكذلك مع الجملة الثانية .

خامساً : التغييرات على مستوى كود Flex1 و Flex2 .

لن نغير شيئاً سوى أنه في حالة بين تاريخين سنضيف خانة لعرض التاريخ ، ولذلك نضيف الكود التالي قبل نهاية أمر Flex1 :

```

If Label2.Visible = True Then
MSFlexGrid1.Cols = 6
MSFlexGrid1.Col = 5
MSFlexGrid1.Text = "التاريخ" =
MSFlexGrid1.ColWidth(5) = 1200
MSFlexGrid1.ColAlignment(5) = 5
End If

```

وكذلك مع Flex2 .

سادساً : التغييرات على مستوى كود Put_Data1 و Put_Data2 :

```

For i = 0 To T4.RecordCount - 1
n = n + 1
Row = n

```

```

. Col = 0
. Text = n
. Col = 1
. Text = T4!Number
. Col = 2
  SQL = "select name from tb_product where number=" &
T4!product
  Set T6 = D1.OpenRecordset(SQL, dbOpenDynaset)
. Text = T6!Name
  T6.Close
. Col = 3
. Text = T4!Count
. Col = 4
. Text = T4!price
  If Label2.Visible = True Then
. Col = 5
. Text = T4!Date
  End If
  T4.MoveNext
Next i

```

وأيضاً مع Put_Data2 .

لن ننسى ايضاً الأوامر الموجودة في القائمة في الشاشة الرئيسية ، والتي ستصبح بالشكل التالي :

```

If T1.RecordCount = 0 Then
  MsgBox "لا توجد أي بضائع لعرضها ، لاضافة بضاعة جديدة اختر اضافة - اضافة vbExclamation + arabic"؟؟"عرض البضائع" ،
  Exit Sub
End If

```

Select Case Index

Case 0

Unload Frm_Report4

Frm_Report4.Show

Case 1

Unload Frm_Report4

Frm_Report4.Show

Frm_Report4.MaskedTextBox2.Visible = True

Frm_Report4.Label2.Visible = True

Frm_Report4.Command2.Left = 120

Frm_Report4.Command2.Value = True

Case 3

End Select

ملف مرفق :

قم بمطالعة الملف prog22.zip في المرفقات لمزيد من التفاصيل .

الطباعة باستخدام داتا ريبورت

ما هي الداتا ريبورت DataReport ؟

هي إحدى نماذج العرض المرفقة افتراضياً مع الفيجوال بيسك ، وهي مصممة لطباعة التقارير والجداول من قواعد البيانات ، وهي تستقبل جملة استعلام أو اسم جدول لكي تقوم بطباعته . وهناك نماذج أو أدوات أخرى في هذا المجال من أشهرها الكريستال ريبورت وهو أكثر احترافية من الداتا ريبورت ولكنه ليس موجود افتراضياً مع مكتبات فيجوال بيسك .

وسوف نتعرف في هذه الدروس على كيفية التعامل مع DataReport .

لنضيف DataReport فإننا ومن خلال النقر بزر الماوس الايمن على الفورم سوف تظهر لنا Add ، ونختار منها DataReport .

نلاحظ أن الصفحة التي ظهرت لنا مقسمة إلى خمسة اقسام :

Report Header : البيانات التي تظهر فقط مع أول صفحة في التقرير .

Page Header : البيانات التي تظهر على رأس كل صفحة .

Details : محتويات التقرير ، ويمكن ان تعرض على اكثر من صفحة .

Page Footer : ما يظهر مع كل آخر صفحة .

Report Footer : بيانات تظهر في آخر صفحة في التقرير فقط .

والآن سوف تجد على يمين الشاشة مربع أدوات تحكم Control Box غير المعتاد في شاشات

فيجوال بيسك التقليدية تحت تبويب Tab باسم DataReport ، وتحتوي على الأدوات

التالية - من اليسار - :

Pointer : وهو مؤشر السهم المعروف في الفيجوال بيسك . ويستخدم للتحديد وغيره مما لا يلزمنا

شرحه .

RptLabel : هو مجرد أداة لعرض النصوص التي لا تعتمد على قاعدة البيانات ، يمكن تغيير

محتواها فقط بواسطة الكود قبل العرض مثلاً ، وتستخدم لوضع العنوان مثلاً وما شابه .

RptText : هذا النوع مخصص للارتباط بحقل معين في قاعدة البيانات ، ويستخدم لوضع بيانات

العملاء على سبيل المثال ، وسوف نتعرف لاحقاً على كيفية القيام بذلك .

RptImage : لوضع صورة في التقرير . يمكن ضبطها أيضاً وقت التصميم باستخدام

LoadPicture المعروفة .

RptLine : خط لتنسيق طباعة التقرير .

RptShape : أيضاً مثل ال Line السابق .

RptFunction : تستخدم لعرض بعض العمليات التي قد نحتاج إليها لاحقاً ، وسنرى أمثلة

عن ذلك لاحقاً .

وللبدء في التعامل مع البيانات من خلال زر عرض التقرير .

أولاً : للتحكم في الخصائص الثابتة .

مثل بيانات Rptlable ، الألوان وغير ذلك .

قم أولاً بوصف اسم DataReport ، ثم اسم القسم ، ثم اسم الاداة بالشكل التالي :

```
DataReport2.Sections("Section2").Controls("l1").Caption =  
Lbl_name
```

ويمكن بنفس الطريقة التحكم بالالوان وما شابه ذلك .

القسم الثاني : كيفية ربط ال RptTextBox بقاعدة البيانات .

خلال مرحلة التصميم احرص على ضبط الخواص التالية :

DataMember و DataFields لتحديد اسم قاعدة البيانات (كائن قاعدة البيانات أو

اداتها) ، و اسم الحقل المطلوب عرضه في هذا المكان .

ويمكن ضبطهما وقت التصميم أو التنفيذ ، ففي وقت التصميم وفي حقل DataFields يمكنك

كتابة اسم الحقل مثل fullname ، في حين قم مثلاً بتنفيذ أمر تحديد مصدر البيانات وقت

التنفيذ بالشكل التالي :

```
Set DataReport1.DataSource = Rs
```

وهذا ما سنقوم بتوضيحه عند البدء في العمل مع الداتا ريبورت في برنامجنا .

القسم الثالث : كيفية استخدام RptLable لعرض بعض العمليات .

يتم ذلك بوضع القيمة المناسبة في خاصية Caption لهذه الأداة ، فمثلاً لو قمنا بكتابة D٪ فإن

ذلك يعطينا تاريخ اليوم ، T٪ تعطينا الوقت ، P٪ تعطينا عدد الصفحات الاجمالي ، أما p٪

فتعطينا رقم الصفحة الحالية .

القسم الرابع : كيفية استخدام RptFunction

من خلال الخاصية FucntionType نختار طبيعة العملية التي نود القيام بها ، وذلك بعض

ضبط خصائص قاعدة البيانات الخاصة بها كما اسلفنا في RptText ، ولن نستخدم هذه الاداة في

برنامجنا .

أخيراً نذكر بأن هذه التقارير تستقبل كمصدر لبياناتها كائن قاعدة بيانات Ado ، لذلك لا بد من تعريف هذه الكائنات أولاً وذلك بالشكل التالي :

١- من قائمة Project ثم Refrences قم باختيار مكتبة للأدو مثل :
Microsoft Access ActiveX Data Object 2.5 Library

٢ - في أمر الطباعة لا بد من تعريف نسخة من كائن قاعدة البيانات من Ado ونسخة من جدول ومن ثم القيام بعملية التكوين لكليهما بالشكل التالي :

```
Dim cn As ADODB.Connection  
Dim Rs As ADODB.Recordset  
Set cn = New ADODB.Connection  
Set Rs = New ADODB.Recordset
```

٣ - والآن قم بربط قاعدة البيانات بمصدر قاعدة البيانات من نوع أكسيس مثلاً على سبيل المثال من خلال Connection String بالشكل التالي :

```
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data  
Source=" & Folder & "db1.mdb;" & "Jet OLEDB:Database  
Password=" & PassWord"
```

يمكنك زيارة msdn للعثور على كيفية تكوين Connection String على هذا الرابط :
<http://msdn.microsoft.com/library/default.asp?url=/library/enus/vbcon98/html/vbconmanagingjetodbconnections.asp>

حيث يشرح القسم الذي يشير هذا الرابط إلى أحد أقسامه إلى كيفية بناء Connection String ، كما ويحيلك في آخره إلى أمثله بالكود ال Language Refernce

ولكن بصورة اجمالية فإنك تمرر في البارميتر الأول نوع قاعدة البيانات ، الثاني تحدد له مكان قاعدة البيانات ، ويمكنك تمرير المزيد من البارميترات لاحقاً مثل كلمة المرور ويتم التفريق بين هذه البارميترات من خلال استخدام الفاصلة المنقوطة ؛ .

٤ - الخطوة الأخيرة هي تحديد مصدر البيانات وربطه ب **rs** ومن ثم ربط **rs** مع **DataReport** ، ولا تنس ان مصدر البيانات قد يكون اسماً لجدول أو جملة استعلام :

```
sql = "select * from table1"  
Set Rs = cn.Execute(sql)  
Set DataReport1.DataSource = Rs
```

٥ - لا تنس أمر اظهار التقرير :

```
DataReport1.Show
```

٦ - كما يمكنك طباعة التقرير دون عرضه من خلال الأمر :

```
DataReport1.PrintReport
```

ويمكنك اضافة بعض البارميترات الاضافية مثل **True** في البارميتر الأول لعرض مربع حوار طباعة قبل الطباعة لتحديد عدد النسخ وما شابه :

```
DataReport2.PrintReport True
```

أو تحديد **Range** - مدى الصفحات التي ستقوم بطباعتها - ، إلى آخر ذلك من بارميترات هذه الوظيفة .

والآن سوف نطبق ما تعلمناه على طباعة شاشة عرض البضائع الرئيسي الموجودة في **Frm_Show** في البداية سوف نقوم بتكوين **DataReport** الخاص بنا ، وذلك بالشكل التالي - بعد اضافته من **Add** طبعاً - .

بالطبع ستكون عناوين الجدول في القسم **Page Header** لأننا نريد أن تظهر في بداية كل صفحة ، وسوف نقوم برسمها على هيئة **Rpt_Labels** ، ونضع المعلومات التالية كـ **Caption** لهذه العناصر .

رقم - اسم البضاعة - اسم المصنع - النوع - السعر - الكمية - وحدة/صندوق .

أي ٧ من RptLabel .

وهذا القسم ثابت تماماً ... أما القسم الآخر فهو الذي يعتمد على القراءة من قاعدة البيانات .
سنرسم فيه على نفس العدد RptTextBox ، لكننا لن نضبط خاصية Caption بل إننا
سنضبط خاصية Data Field ، وسنضع فيها الحقل مصدر البيانات ، وذلك على النحو التالي -
بالترتيب .

tb_product.number - tb_product.name - tb_factory.name -
tb_category.name - Price - Count - Box_Count

لاحظ أن الحقول التي لا تتشابه في الجداول توضع كما هي مباشرة ، أما المتشابهة فنسبقتها باسم
الجدول .
لاحقاً قم بضبط Font كما يحلو لك ، واجعل اللون للصف العلوي مثلاً مختلف عن الآخرين ، أو
كما تريد .


قم بإضافة (بسم الله الرحمن الرحيم) في أول صفحة ، ولا تنس إن أضفتها ألا تقوم بالقاء الورقة التي
تطبعها . بإمكانك أيضاً المحاذاة إلى اليمين أو اليسار ، أو توسيط النص في أي أداة .

قم بإضافة بعض RptLabel لتضع بها تاريخ اليوم وما شابه ذلك ، أيضاً مجموعة أخرى لنضع
فيها اسم المحل ورقم هاتفه على سبيل المثال .

ستلاحظ أن تحديداتك للون الخلفية ، أو ظهور الخط لن تظهر إلا بعد ضبط الخاصيتين
BorderStyle و BackStyle إلى ١ في الأدوات السابقتين .

حاول الوصول إلى الشكل التالي إن كان يناسبك ، وإلا فاختر الشكل الذي تود إضافته :

ليصائع

Report Header (Section4)																											
بسم الله الرحمن الرحيم																											
Page Header (Section2)																											
																											
بيان بالبيانات المضافة وكمياتها																											
اسم العميل هنا																											
رقم المالك هنا																											
<table border="1"> <tr> <th>رقم</th> <th>اسم البضاعة</th> <th>المصنع</th> <th>النوع</th> <th>الكمية</th> <th>السعر</th> <th>وحدة/صندوق</th> </tr> </table>																رقم	اسم البضاعة	المصنع	النوع	الكمية	السعر	وحدة/صندوق					
رقم	اسم البضاعة	المصنع	النوع	الكمية	السعر	وحدة/صندوق																					
Detail (Section1)																											
<table border="1"> <tr> <th>box_cou</th> <th>price</th> <th>count</th> <th>tb_catego</th> <th>tb_factory name</th> <th>tb_product name</th> <th>tb_pr</th> </tr> </table>																box_cou	price	count	tb_catego	tb_factory name	tb_product name	tb_pr					
box_cou	price	count	tb_catego	tb_factory name	tb_product name	tb_pr																					
Page Footer (Section3)																											
<table border="1"> <tr> <td>p%</td> <td>من</td> <td>p%</td> <td>صفحة</td> </tr> </table>																p%	من	p%	صفحة								
p%	من	p%	صفحة																								
Report Footer (Section5)																											
<table border="1"> <tr> <td colspan="6">تمت الطباعة في</td> </tr> <tr> <td>T%</td> <td>عدد</td> <td>D%</td> <td colspan="3"></td> </tr> </table>																تمت الطباعة في						T%	عدد	D%			
تمت الطباعة في																											
T%	عدد	D%																									

والآن إلى الجزء الخاص بكتابة الكود ، في البداية لا تنس ان تضيف المكتبات المطلوبة كما أوضحنا في جزء سابق من هذا الدرس .
الجزء الأول الخاص بتعريف كائنات من قاعدة البيانات والجداول :

```
Dim cn As ADODB.Connection
Dim Rs As ADODB.Recordset
Set cn = New ADODB.Connection
Set Rs = New ADODB.Recordset
```

تحديد مصدر البيانات لقاعدة البيانات ، ومصدر البيانات للتقرير :

```
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
Source=" & Folder & "db1.mdb;" & "Jet OLEDB:Database
Password=" & PassWord
```

```
SQL = "select tb_product.*,tb_category.*,tb_factory.* from  
tb_product,tb_factory,tb_category where  
tb_product.category=tb_category.number and  
tb_product.factory=tb_factory.number"
```

```
Set Rs = cn.Execute(SQL)
```

```
Set DataReport1.DataSource = Rs
```

وأخيراً ضبط بعض الخصائص على سبيل المثال اسم المحل ورقم الهاتف :

```
DataReport1.Sections("Section2").Controls("label3").Caption = Info(2
```

```
DataReport1.Sections("Section2").Controls("label4").Caption & " : هاتف" = Info(3
```

أخيراً اظهار التقرير ، وحذف الكائنات من الذاكرة - بعد انتهاء الحاجة إليها - :

```
DataReport1.Show
```

```
Set cn = Nothing
```

```
Set Rs = Nothing
```

تقرير البضائع .

سوف نصمم هذا التقرير ليعرض العمليات التي جرت على كل بضاعة ، وتنقسم إلى قسمين لا فرق بينهما سوى في خاصية Kind التي تحدد إما مبيعات أو مشتريات .

لذا سوف نضع الزرين في Frm_Report1 ، وسوف نرى الفرق الوحيد بينهم .

والآن لنر عناصر التصميم ، فهو مثل التقرير السابق تماماً ، ما عدا أن الحقول سوف تكون :

رقم - تاريخ - كمية - سعر .

وسوف تكون مصادرها من قاعدة البيانات :

Number - Date - Count - Price

سوف نضيف مربعات عناوين RptLabel من 1 إلى 15 لنعرض فيهم معلومات هذه البضاعة
هكذا نكون قد أنهينا التصميم ، وبعد رسم زري الطباعة للمبيعات والمشتريات تحت اسم
Command3(0 للمبيعات و Command3(1 للمشتريات ، فسوف نبدا الكود
بالطريقة المعتادة بالشكل التالي :

```
Dim cn As ADODB.Connection
Dim Rs As ADODB.Recordset
Set cn = New ADODB.Connection
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data
Source=" & Folder & "db1.mdb;" & "Jet OLEDB:Database
Password=" & PassWord
Set Rs = New ADODB.Recordset
```

نقطة الاختلاف هي تحديد مصدر البيانات ، وحيث أن Index يمثل لنا ٠ أو ١ مثل تلك التي
تمثلها الخاصية Kind في قاعدة البيانات للمبيعات والمشتريات ، فإننا نستطيع كتابة الكود التالي :

```
Set Rs = cn.Execute(sql_now & " and kind=" & Index(
Set DataReport2.DataSource = Rs
```

ولكن من هي sql_now ؟

لأننا لن نستطيع لاحقاً أن نعرف العملية الحالية لأن المستخدم قد يغير التاريخ دون ضغط زر (تنفيذ)
مما يؤدي لحدوث خطأ ، من أجل ذلك سنعرف هذا المتغير في قسم التصاريح Genral

```
Dim sql_now
```

وفي الجزء الأول من كود التنفيذ سنغير الشكل ليصبح كالتالي :

```

If Option1.Value = True Then
    sql = "select * from tb_sel_bay where product=" & num
    & " and kind=" & i
    sql_now = "select * from tb_sel_bay where product=" &
    num
Else
    sql = "select * from tb_sel_bay where product=" & num
    & " and kind=" & i & " and date between#" &
    Format(MaskedTextBox1.Text, "MM/DD/YYYY") & "# and #" &
    Format(MaskedTextBox2.Text, "MM/DD/YYYY" & "#" & "("
    sql_now = "select * from tb_sel_bay where product=" &
    num & " and date between#" & Format(MaskedTextBox1.Text,
    "MM/DD/YYYY") & "# and #" & Format(MaskedTextBox2.Text,
    "MM/DD/YYYY" & "#" & "("
End If

```

وهذا ما يجعل sql_now يحتوي على الوضع الحالي دون تفريق بين المبيعات والمشتريات ، وهذا ما سنقوم بعمله في كود الطباعة .

نعود لمواصلة الكود ، وسننقل البيانات نقلاً طبيعياً كالتالي ، ثم نقوم باظهار التقرير بصورة طبيعية ، وقد شرحت معظم هذه الطرق في التقرير السابق :

```

DataReport2.Sections("Section2").Controls("I1").Caption =
Lbl_name
DataReport2.Sections("Section2").Controls("I2").Caption =
lbl_cate
DataReport2.Sections("Section2").Controls("I3").Caption =
lbl_fact
DataReport2.Sections("Section2").Controls("I4").Caption =
lbl_prc
DataReport2.Sections("Section2").Controls("I5").Caption =
lbl_cnt
Select Case Index
Case 0

```

```
DataReport2.Sections("Section2").Controls("label1").Caption = "تقرير باجمالي مبيعات بضاعة"
```

Case 1

```
DataReport2.Sections("Section2").Controls("label1").Caption = "تقرير باجمالي مشتريات بضاعة"
```

End Select

```
DataReport2.Sections("Section2").Controls("label3").Caption = Info(2,
```

```
DataReport2.Sections("Section2").Controls("label4").Caption & " : هاتف" = Info(3,
```

DataReport2.Show

Set cn = Nothing

Set Rs = Nothing

طباعة التقرير بين تاريخين شاملاً التقرير اليومي .

بمثل طريقة الدرس السابق ، سوف نصمم الشاشة كما تعلمنا بحيث تحتوي على المعلومات التالية :

رقم العملية - اسم البضاعة - التاريخ - الكمية - السعر .

ومصادرهـم :

tb_sel_bay.number - name - date - tb_sel_bay.count -
tb_Sel_bay.price

وتكون مربعات العناوين الخمسة التي اضعناها في التقرير السابق مستمرة ولكنها سوف تعرض المعلومات التالية :

* اليوم من الأسبوع .

* عدد عمليات المبيعات .

* عدد عمليات المشتريات .

* اجمالي سعر المبيعات .

* اجمالي سعر المشتريات

وهي معلومات معروضة في الفورم .

مجدداً سوف نستخدم طريقة `sql_now` لمعرفة جملة الاستعلام المنفذة حالياً ، كما سنستخدم أيضاً `dt1` و `dt2` لمعرفة التاريخ ، وأيضاً `typ` لمعرفة هل الاستعلام بين تاريخين أم ليوم واحد فقط .

سوف نغير الأسطر التالية من كود التنفيذ :

```
If Label2.Visible = False Then
    SQL = "select * from tb_sel_bay where kind=0 and
date=#" & Format(MaskedTextBox1.Text, "MM/DD/YYYY & ("
"#")
Else
    SQL = "select * from tb_sel_bay where kind=0 and date
between #" & Format(MaskedTextBox1.Text, "MM/DD/YYYY")
& "# and #" & Format(MaskedTextBox2.Text, "MM/DD/YYYY ("
"#") &
End If
```

إلى :

```

If Label2.Visible = False Then
    SQL = "select * from tb_sel_bay where kind=0 and
date=#" & Format(MaskedTextBox1.Text, "MM/DD/YYYY & ("
"#"

    sql_now = "select tb_sel_bay.*,tb_product.* from
tb_sel_bay,tb_product where date=#" &
Format(MaskedTextBox1.Text, "MM/DD/YYYY") & "# and
tb_product.number = tb_sel_bay.product"
    dt1 = MaskedTextBox1.Text
    typ = 0
Else
    SQL = "select * from tb_sel_bay where kind=0 and date
between #" & Format(MaskedTextBox1.Text, "MM/DD/YYYY")
& "# and #" & Format(MaskedTextBox2.Text, "MM/DD/YYYY ("
"#" &

    sql_now = "select tb_sel_bay.*,tb_product.* from
tb_sel_bay,tb_product where date between #" &
Format(MaskedTextBox1.Text, "MM/DD/YYYY") & "# and #" &
Format(MaskedTextBox2.Text, "MM/DD/YYYY") & "# and
tb_product.number = tb_sel_bay.product"
    dt1 = MaskedTextBox1.Text
    dt2 = MaskedTextBox2.Text
    typ = 1
End If

```

وأخيراً كود الطباعة تماماً مثل المرة السابقة مع بعض التغييرات الطفيفة فقط :

```

Private Sub Command3_Click(Index As Integer)
Dim cn As ADODB.Connection
Dim Rs As ADODB.Recordset
Set cn = New ADODB.Connection

```

```
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data  
Source=" & Folder & "db1.mdb;" & "Jet OLEDB:Database  
Password=" & PassWord  
Set Rs = New ADODB.Recordset  
Set Rs = cn.Execute(sql_now & " and kind=" & Index_  
Set DataReport3.DataSource = Rs
```

```
DataReport3.Sections("Section2").Controls("l1").Caption =  
Lbl_day  
DataReport3.Sections("Section2").Controls("l2").Caption =  
lbl_cnt1  
DataReport3.Sections("Section2").Controls("l3").Caption =  
lbl_cnt2  
DataReport3.Sections("Section2").Controls("l4").Caption =  
lbl_prc1  
DataReport3.Sections("Section2").Controls("l5").Caption =  
lbl_prc2  
Select Case Index  
Case 0  
If typ = 0 Then
```

```
DataReport3.Sections("Section2").Controls("label1").Caption  
on & " تقرير المبيعات ليوم " = dt1  
Else
```

```
DataReport3.Sections("Section2").Controls("label1").Caption  
on & " تقرير المبيعات للمدة بين " & dt1 & " و " & dt2  
End If  
Case 1  
If typ = 0 Then
```

```

DataReport3.Sections("Section2").Controls("label1").Caption & " = تقرير المشتريات ليوم " & dt1
Else

DataReport3.Sections("Section2").Controls("label1").Caption & " = تقرير المشتريات للمدة بين " & dt1 & " و " & dt2
End If
End Select
DataReport3.Sections("Section2").Controls("label3").Caption = Info(2,
DataReport3.Sections("Section2").Controls("label4").Caption & " : هاتف " = Info(3,

DataReport3.Show

Set cn = Nothing
Set Rs = Nothing
End Sub

```

طباعة نتائج البحث .

لطباعة تقرير حول البحث ، فإننا لا نحتاج لأكثر من تصميم الشاشة الأولى تماماً ... كل ما سنغير فيه هو جملة الاستعلام .

نقوم بعمل شاشة شبيهة تماماً بالشاشة الأولى دون أي تغيير ونعود لشاشة البحث لكتابة كود الطباعة .

قبل كتابة كود الطباعة سوف نقوم بإضافة كل من **Sql_now** لعرض جملة الاستعلام الحالية كما تعلمنا سابقاً ، وكل من **a1,a2,a3** لتوضيح متغيرات عملية البحث الثلاثة : البحث بدلالة ، معامل البحث ، كلمة البحث .

ونكتب الكود ضمن عملية البحث وذلك في الدالة **Search** كما ترى :

```
sql_now = SQL  
a1 = Combo1.Text  
a2 = Combo2.Text  
a3 = Text1.Text
```

والآن ننتقل لكتابة الكود ، وفي البداية الاتصال بقاعدة البيانات :

```
If sql_now = "" Then Exit Sub  
Dim cn As ADODB.Connection  
Dim Rs As ADODB.Recordset  
Set cn = New ADODB.Connection  
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & "Data  
Source=" & Folder & "db1.mdb;" & "Jet OLEDB:Database  
Password=" & PassWord  
Set Rs = New ADODB.Recordset  
Set Rs = cn.Execute(sql_now)  
Set DataReport4.DataSource = Rs
```

ضبط المعلومات المعروضة ، وتغيير العنوان :

```
DataReport4.Sections("Section2").Controls("label3").Caption = Info(2)  
DataReport4.Sections("Section2").Controls("label4").Caption & " : هاتف" = Info(3)  
DataReport4.Sections("Section2").Controls("label1").Caption & " : نتائج البحث عن بضائع" = a1 & a2 & a3
```

استكمال عملية الطباعة :

DataReport4.Show

Set cn = Nothing

Set Rs = Nothing

ملف مرفق :

قم بمطالعة الملف prog24.zip في المرفقات لمزيد من التفاصيل .

هذا هو الدرس الأخير من هذا القسم ، في القسم القادم سوف نقوم بتحسين أداء البرنامج وإضافة اشرطة الأدوات بإذن الله .

القسم الثالث

عمل برنامج لإدارة

السوبر ماركت

باستخدام

Visual Basic 6.0

- المرحلة الثالثة -

في هذا الفصل :

- التعامل مع شريط الحالة وشريط الأدوات .
- نقاط سريعة في البرنامج .
- البحث بأكثر من معامل .
- طباعة الفواتير .
- تحديد العمليات السريعة .
- التحكم في التقارير .
- التحكم في ألوان البرنامج .
- تعددية المستخدمين وقواعد البيانات .
- كتابة تعليمات البرنامج .
- تحويل البرنامج إلى تطبيق exe
- عمل عملية Setup

التعامل مع شريط الحالة

سوف نحاول في هذه المرحلة انتهاء البرنامج واعداده للنشر ... وسنبدأ باضافة اللمسات الأخيرة على البرنامج .

التعامل مع شريط الحالة

سنعود الآن مجدداً إلى الفورم الرئيسي ، وسنقوم باضافة شريط الحالة StatusBar ووضع بعض المهام الرئيسية فيه .

من شريط أدوات التحكم قم باضافة StatusBar وستجده تلقائياً يظهر في أسفل الفورم ، من ضمن الخصائص داخل هذه الأداة اختر Custem واختر التبويب الثاني Panels .
تجد زرین Insert Panel و Remove Panel لاضافة وحذف أجزاء من هذا الشريط .

أيضاً تجد الخصائص التالية لكل جزء من هذا الشريط :

Text : لتحديد النص الذي يظهر في هذا الجزء .

Alignment : لتحديد محاذاة النص داخل هذا الجزء .

Style : افتراضياً على text حيث يوضع النص المحدد بالخاصية Text ، لكن يمكن اختيار عرض التاريخ أو الوقت الحالي وغيرها .

Beval : لتحديد بروز وغطس كل جزء من الشريط .

AutoSize : يقسم الحجم افتراضياً بين جميع اجزاء الشريط بالتساوي ، ولكن ولأن بعض الاجزاء تكون كبيرة نضبط هذه الخاصية لها بحيث تتناسب مع حجم النص .

سنضيف العناصر التالية داخل شريط الحالة :

* تحديث الشريط : بعض المعلومات في هذا الشريط تحتاج إلى تحديث مثل عدد البضائع وغيرها .
سيكون هذا الجزء بارزاً على هيئة الزر .

* الوقت الحالي : وسنضبطه من خلال خاصية Style .

* عدد البضائع .

* عدد البضائع غير المتوفرة .

وسنضبط قيمهما وقت التصميم ، وهما المعنيان بزر التحديث ، وقم بوضع قيم في خاصية Key

ليمكنك ضبطها لاحقاً .

* تشغيل الآلة الحاسبة .

* تشغيل التقرير اليومي .

* وأخيراً البريد الالكتروني (راسلنا) .

بعد ذلك سنبدأ في البرمجة ، ونقوم بذلك عبر اختبار النص المضغوط ، ولأن بعض النصوص متغيرة فلن نستطيع اختبار كامل النص ، لذا سنقتصر على اختبار أول خمسة أحرف فقط ، وبناء عليه ننفذ بعض الأوامر التي تعلمناها سابقاً :

```
Private Sub StatusBar1_PanelClick(ByVal Panel As
MSComctlLib.Panel)
Select Case Left(Panel, 5)
Case "تحديث"
Refresh_St
Case "عرض ا"
If T1.RecordCount = 0 Then
MsgBox "لا توجد أي بضائع لعرض تقارير عنها ، لاضافة بضاعة جديدة اختر اضافة -
vbExclamation + arabic" عرض البضائع "اضافة بضاعة"،
Exit Sub
End If
Unload Frm_Report4
Frm_Report4.Show
Case "تشغيل"
Frm_Calc.Width = 3105
Frm_Calc.Height = 2640
Frm_Calc.Top = 0
Frm_Calc.Left = 0
Frm_Calc.Show
```

Case "لديك"

If T1.RecordCount = 0 Then

MsgBox "لا توجد أي بضائع للبحث عنها ، لاضافة بضاعة جديدة اختر اضافة -
، "عرض البضائع"vbExclamation + arabicاضافة بضاعة"،

Exit Sub

End If

Frm_Find.Show

Frm_Find.Text1.Text = 0

Frm_Find.Combo1.ListIndex = 3

Frm_Find.Combo2.ListIndex = 2

Frm_Find.Command2.Value = True

Case "Hamma"

Shell "explorer mailto:Hammada2091@Hotmail.com"

End Select

End Sub

أما في أمر التحديث :

Private Sub Refresh_St()

SQL = "select count(*) as cnt from tb_product"

Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,

StatusBar1.Panels(3).Text & "بضاعة" = T4!cnt"عدد البضائع " &

T4.Close

SQL = "select count(*) as cnt from tb_product where
count=0"

Set T4 = D1.OpenRecordset(SQL, dbOpenDynaset,

StatusBar1.Panels(4).Text & "بضاعة غير متوفرة" = T4!cnt"لديك " &

T4.Close

End Sub

*** ملاحظة : نستطيع بنفس الطريقة الاعتماد على خاصية Index كإدليل لقراءة النص المعروض بالشكل التالي :

Select Case Panel.Index

ونعيد نفس الأكواد السابقة ، وهذه هي الطريقة الأمثل لهذه العملية .

عمل شريط الأدوات .

سنقوم بعمل شريط أدوات لبرنامجنا خلال هذا الدرس .. ولهذا الغرض سوف نستخدم أداتين :

* أداة **Toolbar** : لعرض الشريط والتحكم فيه .

* أداة **ImageList** : نستخدم هذه الأداة لعرض الصور في الشريط ، وهذه الأداة تستخدم مع أكثر من أداة أخرى لعرض الصور بداخلها .

والآن نقوم بتحديد المهام التي سنضع لها رمزا في شريط الأدوات لكي نبدأ في البحث عن أيقونات مناسبة لها ، وهي :

* عرض .

* تقرير البضائع .

* التقرير اليومي .

* شراء .

* بيع .

* بحث .

* آلة حاسبة .

* نسخ .

* كلمة السر .

* حول .

* خروج .

وبعد البحث نقوم بوضع الأيقونات في مسار البرنامج تمهيداً لاستخدامها .

يأتي الآن دور استخدام أداة Imagelist ، لذا قم بتحديدتها ، واضغط على Custom من خصائصها ، واختر التبويب Image ، والآن قم بشكل متكرر بما يلي :

* اضغط Insert Picture .

* اختر الصورة المطلوبة .

وهكذا حتى تنتهي من الصور ... وهكذا انتهى دور هذه الاداة .

الآن حدد الأداة الثانية Toolbar ، واختر Custom .

ستظهر لك شاشة الخصائص ، في الشاشة الأولى نستطيع التحكم بما يلي :

* Image List : منها نستطيع تحديد مصدر الصور وهي أدواتنا 1 Imagelist .

* Appearance و Style : تحدد المظهر من ثلاثية الأبعاد وغيرها .

* TextAligment : اتجاه النص مع الصورة .

وفي التبويب الثاني Buttons نحدد ما يلي :

نختار Inseret Button ، ومن خاصية Caption نحدد النص الذي سيظهر معها . ومن

ToolTipText نختار النص الذي يظهر عند مرور الماوس .

أخيراً خاصية Style تحدد شكل الزر .

فمثلاً الافتراضي ٠ زر عادي لكن ١ مثلاً يكون على شكل مربع اختيار ، أما ٢ فهو زر يظهر قائمة ،

و ٣ هو خط Separator نضعه عادة بين كل زرين .

أيضاً خاصية Image تحدد رقم الصورة التي تظهر مع كل زر ، ونعرف الرقم من أدواتنا السابقة

. ImageList

بهذا الشكل نقوم بضبط تصميم شريط الأدوات ، ويتبقى علينا الكود .

أما عن برمجة شريط الأدوات فيظهر الكود في البداية بالشكل التالي :

```
Private Sub Toolbar1_ButtonClick(ByVal Button As  
MSComctlLib.Button)
```

حيث يحتوي المتغير Button على النص الموجود في الزر المضغوط .
لذلك سوف نستخدم جملة Select بالشكل التالي :

Select Case Button Case "خروج"

وبهذا الشكل ، ومنعاً لتكرار الأكواد في بعض الحالات مثل تغيير كلمة السر قمت بإنشاء حدث
Chng_pass ووضعت فيه كود تغيير كلمة السر وقمت باستدعاء الحدث مباشرة سواء من
القائمة أو من شريط الأدوات .

اجمالي الكود سوف يكون بالشكل التالي :

```
Private Sub Toolbar1_ButtonClick(ByVal Button As  
MSComctlLib.Button)  
Select Case Button  
Case "خروج"  
    Dim sure  
    sure = MsgBox (vbExclamation + "هل تريد بالتأكيد انهاء البرنامج ؟" ،  
vbYesNo + arabic("تأكيد الخروج") ،  
    If sure = vbYes Then End  
Case "حول"  
    AboutMe.Show 1  
Case "كلمة السر"  
    Call Chng_pass  
Case "نسخ"  
    Copy_Me  
Case "آلة حاسبة"  
    Frm_Calc.Width = 3105  
    Frm_Calc.Height = 2640  
    Frm_Calc.Top = 0  
    Frm_Calc.Left = 0
```

```

Frm_Calc.Show
Case"بحث"
    If T1.RecordCount = 0 Then
        MsgBox – لا توجد أي بضائع للبحث عنها ، لاضافة بضاعة جديدة اختر اضافة –
        ، "عرض البضائع"vbExclamation + arabic"اضافة بضاعة"،
        Exit Sub
    End If
    Frm_Find.Show
Case"بيع"
    If T1.RecordCount = 0 Then
        MsgBox – لا توجد أي بضائع لعرضها ، لاضافة بضاعة جديدة اختر اضافة – اضافة
        ، "عرض البضائع"vbExclamation + arabic"اضافة بضاعة"،
        Exit Sub
    End If
    Frm_Sel_Bay.Lbl_name.Caption = "اضافة عملية بيع"
    Frm_Sel_Bay.Refresh_Me
    Frm_Sel_Bay.Show 1
Case"شراء"
    If T1.RecordCount = 0 Then
        MsgBox – لا توجد أي بضائع لعرضها ، لاضافة بضاعة جديدة اختر اضافة – اضافة
        ، "عرض البضائع"vbExclamation + arabic"اضافة بضاعة"،
        Exit Sub
    End If
    Frm_Sel_Bay.Lbl_name.Caption = "اضافة عملية شراء"
    Frm_Sel_Bay.Refresh_Me
    Frm_Sel_Bay.Show 1
Case"تقرير يومي"
    If T1.RecordCount = 0 Then
        MsgBox – لا توجد أي بضائع لعرض تقارير عنها ، لاضافة بضاعة جديدة اختر اضافة –
        ، "عرض البضائع"vbExclamation + arabic"اضافة بضاعة"،
        Exit Sub
    End If
    Unload Frm_Report4

```

```

Frm_Report4.Show
Case "تقرير بضائع"
    If T1.RecordCount = 0 Then
        MsgBox "لا توجد أي بضائع لعرضها ، لاضافة بضاعة جديدة اختر اضافة - اضافة vbExclamation + arabic" عرض البضائع ،
    Exit Sub
End If

Frm_Report1.Show
Case "عرض"
    If T1.RecordCount = 0 Then
        MsgBox "لا توجد أي بضائع لعرضها ، لاضافة بضاعة جديدة اختر اضافة - اضافة vbExclamation + arabic" عرض البضائع ،
    Exit Sub
End If

Frm_Show.Show
End Select
End Sub

```

*** بنفس الطريقة في شريط الحالة فإن بإمكاننا استخدام الخاصية **Index** بدلاً من القيمة المعروضة على الشريط .

بهذا الدرس نكون قد اتممنا بحمد الله البرنامج ، ما تبقى لنا الآن هو اعداد البرنامج وتجهيزه للعمل .
وحتى الدرس التالي فهذه هي آخر نسخة من البرنامج بها كل ما تم من عمليات منذ بدء البرنامج .

ملف مرفق :
قم بمطالعة الملف programme.zip في المرفقات لمزيد من التفاصيل

*** محتويات **Frm_Edit_SP** لم تكن موجودة قبل هذا الملف وقد تم شرح محتواها في الدرس ١٢ من هذه الدورة ، ولكنها لم تضاف واضيفت الآن .

نقاط سريعة في البرنامج

نقاط سريعة لم يتم تطبيقها في البرنامج - البحث بأكثر من معامل - .

لو نظرت أخي الكريم إلى شاشة البحث تجد أننا نبحث بمعامل واحد فقط ، وهذا وإن كان جيداً لسوبر ماركت متوسط كما أسلفنا ، إلا أننا وفي حالة التعامل مع قاعدة بيانات ضخمة لمؤسسة أو منفذ بيع كبير فإننا نحتاج للبحث بأكثر من خاصية ، فمثلاً البضائع التي كميتها > 10 ومن مؤسسة معينة عند وصول المورد الخاص بها على سبيل المثال .
لعمل ذلك هناك أكثر من طريقة ، منها ما يلي :

نضع بالجزء العلوي من شاشة البحث والخاص بالتحكم في البحث أزرار (إضافة - إنهاء) .
في هذه الشاشة توجد مربعات التحرير والسرد الموجود في الشاشة العادية .
إلا أنه عند النقر على زر إضافة فإنني اضع هذه الشروط في ثلاثة متغيرات على شكل مصفوفة :

$a(x), b(x), c(x)$

وقيمة X تبدأ ب ٠ ، ونحدثها مع كل عملية بحث بحيث تساوي $X + 1$
ونعرفها في قسم التصريحات Genral في الفورم :

Dim a(),b(),c()

Dim X

وفي Form_Load :

$X=0$

فمثلاً يكون كود زر الاضافة :

```

a(x) = Combo1.Text
If a(x) = "اسم البضاعة" Then a(x) = "product.name"
'
'
وهكذا لتحويل النص المعروض إلى اسم الحقل في قاعدة البيانات .
b(x) = Combo2.Text
c(x) = Text1.Text

X=X+1

```

وهكذا حتى ينتهي المستخدم من اضافة معلوماته ، وعند الضغط على زر انتهاء نضيف جملة Sql من المعلومات التي وضعها المستخدم ، وقد تكون بالشكل التالي :

```

SQL = "select tb_product.*,tb_category.*,tb_factory.* from
tb_product,tb_factory,tb_category where
tb_product.category=tb_category.number and
tb_product.factory=tb_factory.number"
For i = 0 To X-1
sql = sql & " and " & a(x) & b(x) & c(x)
Next i

```

وهكذا يتم عرض جميع نتائج البحث المطلوبة كما حددها هو .
تستطيع لاحقاً عرض المعاملات التي بحث عنها في قائمة لمزيد من الاحترافية ، والموضوع به الكثير من النقاط التي قد تكون علامات ايجابية في برنامجك .

طباعة الفواتير .

- * طباعة الفواتير أكثر من طريقة ، منها ما يلي :
- * اضافة العمليات المطلوب اظهارها في الفاتورة واحدة تلو الأخرى ، وتخزين ارقامها في متغيرات .
- * عرض بكافة العمليات خلال مدة واختيار بعض منها ليظهر في التقارير .

الطريقة الأولى :

في الحالة الأولى بإمكاننا الاعتماد على شاشة اضافة عملية بيع ، حيث نضيف عملية البيع الأولى ونخزن رقمها في متغير **a** ثم الثانية ونضيفها في **a** وهكذا ، بالشكل التالي مثلاً - عند الضغط على زر (اضافة) .

```
if a="" then  
a = Number  
else  
a = a & "," & Number  
end if
```

وذلك على اعتبار **Number** هو رقم العملية المضافة .
وبعد الضغط على زر (طباعة فاتورة) أقوم بتقسيم المتغير **a** بواسطة الدالة **Split** :

```
b = split(a,"،"،
```

ومن ثم الدوران على عناصر المصفوفة الجديدة الناتجة من التقسيم و اضافتها للتقرير :

```
for i = 0 to UPBound(b(
```

```
'  
'
```

في النهاية سيكون بحثك على الأرقام الموجودة في المصفوفة ، ومن ثم تقوم بعرضها في تقرير عادي جداً

الطريقة الثانية :

في هذه الطريقة سوف نعرض له شاشة فيها عمليات مبيعات اليوم أو بين تاريخين يحددهما المستخدم ، ويستطيع من خلال النقر على الجدول اضافة العمليات المطلوبة ، فمثلاً نقوم بوضع **List** نعرض فيها أرقام العمليات .

بالشكل التالي ، نضع الكود في حالة النقر على الجدول ضغطتين - أو أي عملية أخرى نخtarها
لاضافة العملية إلى التقرير - .

MsFlexrid1.Cols=1

'على اعتبار أنه صف رقم العملية

List1.AddItem MSFlexGrid1.Text

تستطيع لجعل التصميم أكثر احترافية ، اضافة اسم العملية ، وأن تتيح للمستخدم حذف العناصر أو
البضائع والتحكم في الليست في الترتيب وغيره .
لاحقاً قم بالاستعلام عن الأرقام المخزنة في القائمة ، وقم بوضعها في التقرير .

هناك **طريقة ثالثة** أفضل وأكثر استخداماً ، لن يكون لدينا جدول لعمليات البيع والشراء حيث أن كل
قطعة تمثل عملية بيع ، بل سيكون لدينا في الأول جدول لعمليات البيع أو الشراء بحيث يتم تسجيل
سجل واحد في كل عملية بيع كاملة ثم نقوم بعمل جدول فرعي بعلاقة (واحد إلى مجموعة Many
To One) بدلالة رقم العملية حيث نقوم باضافة القطع والكميات المضافة إلى هذه العملية .
لاحقاً في عملية الطباعة قم بطباعة كافة السجلات المعروضة بالنسبة لعملية بيع محددة .

ليست هذه هي الطرق فحسب ، بل هناك العديد من الطرق الأكثر دقة واحترافية ، ويبقى عليك
اختيار الطريقة المناسبة لمشروعك .

تحديد العمليات السريعة .

لو نظرت في شريط الحالة اسفل الشاشة الرئيسية لوجدت أننا اضعنا اختصارات لبعض العمليات
السريعة مثل (البضائع التي لها الكمية ٠) وهكذا .
على اعتبار أنها أكثر العمليات شيوعاً .

إلا أن المصمم المحترف للبرنامج يضع للمستخدم حرية تحديد عمليات سريعة خاصة به ، وقد يضع
لك قائمة أو يضيفها ضمن شريط المهام ، أو في شريط الأدوات على شكل قائمة أيضاً (مثل خاصية
Back للخلف الموجودة في مستكشف ويندوز) .

لو لاحظت أن هذه العمليات السريعة هي عبارة عن جمل استعمال كتلك التي نستخدمها في البحث ،
ولذلك قد نستطيع حفظ بعض عمليات البحث لاضافتها في قاعدة البيانات لاستخدامها سريعاً لاحقاً .
يمكننا اضافة جدول جديد نعرض فيه نصوص جمل الاستعلام Sql التي يرغب المستخدم في
استخدامها بسرعة ، وأسماء رمزية لها ، وعند فتحنا للبرنامج نضيف أمراً لاضافة قائمة جديدة
تحتوي على هذه العمليات .

أما عن الكود الذي سنقوم بكتابته ، فإننا سوف نبحث في جدول الاستعلامات السريعة عن الاسم
المضغوط ، ونقوم بتخزين جملة الاستعلام الخاصة به وتنفيذها في شاشة البحث .

※ ملاحظة : يمكنك بنفس طريقة الدرس السابق استخدام اكثر من شرط للبحث وتخزينه أيضاً بنفس
الطريقة .

التحكم في التقارير .

نستطيع أن نعطي المستخدم حرية اظهار ما يريد في التقرير ، فمثلاً قبل أن نظهر له شاشة التقرير
نسأله فيما إذا كان يرغب في اظهار الصورة أم لا ، وإذا كان يرغب في ظهور العنوان ، ونتيح له تغييره
، وقد يبدو الأمر أسهل في حالة عمل شاشة خاصة باعدادات طباعة التقارير ويتم تخزين الاعدادات في
قاعدة البيانات كما نخزن كلمة السر ومعلومات الملف ، فنخزن True,false لعرض الصورة ام لا
، ونخزن String لعرض العنوان لو كان يريد اظهاره ، وهكذا ...

التحكم في ألوان البرنامج .

إن ذوقك في اختيار الألوان والتصميم ليس بالضرورة أن يكون هو نفس ذوق مستخدم البرنامج ، بل قد
يرى أحد المستخدمين برنامجك خال من أي لمحات فنية لمجرد أنه لم يناسب ذوقه ، مع أنه لدى
الكثيرين تحفة في التصميم .

ولسوء الحظ ، قد يكون هذا الشخص هو مدير الشركة مثلاً ، وعليه يتوقف تحديد تطبيق برنامجك أم لا .

في السابق كان المبرمج ينزل إلى الشركة لتحليل البرنامج وكان أيضاً يحرص على سؤال المستخدمين عن أذواقهم اللونية والتصميمية في البرنامج ، أما الآن فأصبح من الأفضل دائماً إتاحة الفرصة لمستخدم البرنامج للتحكم بتصميمك كما يشاء .

فتجد برنامج مثل Word مثلاً يتيح لك اختيار أشرطة الأدوات المطلوبة ، واللون ، وفي برنامجنا هذا نستطيع إضافة شاشة بالتصميم نطلب منه فيها المعلومات التالية على سبيل المثال :

لون الخلفية - لون الأزرار - لون مربع النص - ظهور شريط الأدوات - ظهور شريط الحالة .

ويتم تخزين اعدادات المستخدم في جدول خاص به ، وتطبق فور تشغيل البرنامج وذلك في الاجراء . Start

تعددية المستخدمين .

إن موضوع تعددية المستخدمين له أكثر من طريقة ووجه ، ومنها :

تعدد المستخدمين على قاعدة بيانات واحدة .

في هذه الحالة يتم إضافة جدول باسماء المستخدمين ، وكلمات سر لهم ، ويتاح لهم الدخول في قاعدة البيانات ، حيث أنه في شاشة الدخول يختبر اسم المستخدم ويتأكد من كلمة المرور الخاصة به .

أحد التطبيقات الاحترافية لهذا الموضوع هو تحديد صلاحيات الأعضاء ، حيث نضع مجموعة من الحقول True-False أمام كل عضو مثل (امكانية تعديل - امكانية اضافة - امكانية حذف) ويتحكم بها المدير العام للمجموعة .

يمكننا لاحقاً تحديد العمليات التي يقوم بها العضو بإضافة رقم العضو أمام كل عملية ، فيمكننا معرفة من قام بإضافة البضاعة الأولى ، ومن قام بعملية الشراء الثانية .

تعدد المستخدمين وتعدد قواعد البيانات .

في هذه الطريقة يكون لكل مستخدم قاعدة بيانات خاصة به وربما أكثر ، وقبل الشروع في العمل يختار قاعدة البيانات المطلوبة ، ويعض كلمة السر الخاصة بها .

في هذه الحالة يحب أن تتاح فرصة اضافة قاعدة جديدة في الشاشة الخارجية من البرنامج وقبل تسجيل الدخول إلا في حالة كان المطلوب التحكم في الاضافة من المدير فقط .

وهناك العديد من الطرق الأكثر تعقيداً ، ولكن هذا ما سمح المجال بذكره .

كتابة تعليمات البرنامج .

إننا دائماً بحاجة إلى كتابة تعليمات برامجنا لمنح الفرصة لكافة الأشخاص لاستخدام برامجنا بكفاءة كاملة وفاعلية .

نحتاج إلى أن نشرح في التعليمات كافة النقاط المتعلقة بالبرنامج ، فنشرح له مثلاً قوائم البرنامج والمهام الأساسية فيه .

وسنشرح له أيضاً سير البرنامج ، هذا كله بعد موجز تفصيلي نشرح فيه فائدة البرنامج .

إن قيامك بكتابة التعليمات أيضاً يعني اعطاء بعض التلميحات ، فعلى سبيل المثال في برنامجنا سيكون من الجيد تعليمه كيفية استخدام البحث لعرض البضائع التي لم تتبق منها كمية كافية .

أيضاً اشرح له بعض المشاكل التي قد تحدث وكيف يتعامل معها ، وأخيراً اصف إليه معلومات عن كيفية الاتصال بك وعرض مشاكله أو اسداء الاقتراحات .

لا تنس في حال برمجتك لبرامج معينة أن تعرض امكانيات النظام التي يمكنك التعامل معها ، ومكونات الجهاز كالحدد الأدنى من سرعة المعالج وذاكرة الرام .

وملف الوورد مرفق وهو عبارة عن تعليمات الملف الخاص بنا .

ملف مرفق :

قم بمطالعة الملف help.zip في المرفقات لمزيد من التفاصيل .

ولا يشترط أسلوب كتابة التعليمات على الورد ، بل قد نستطيع استخدام برامج خاصة بكتابة ملفات المساعدة يمكنها أن تساعدنا كثيراً ...

تحويل البرنامج إلى تطبيق

إنك كمصمم للبرنامج لن تعطي كل من يريد استخدام برنامجك نسخة من الكود التطبيقي للبرنامج ، بل إنك ستعطيه ملفاً بامتداد **exe** يكفل له تشغيل البرنامج بمجرد الضغط عليه من الويندوز ، وهو ما سنتعلمه خلال هذا الدرس .

من قائمة **File** في الفيجوال بيسك قم باختيار **Make Project1.exe** ، وهنا سوف تظهر لك شاشة مستكشف ويندوز لتحديد مكان الحفظ ، وأربعة أزرار ما يهمنا منهم هو الزر (خيارات - Option) والذي سوف نقوم بالنقر عليه الآن .

في أعلى الشاشة سوف تظهر لك ثلاثة ارقام **Major** ، و **Minor** و **Rivision** ، وهذه هي الأرقام التي تحدد اصدارك .

في العادة يكون الرقم الأول هو الرقم المحدد للاصدار ككل ، وبتغير الرقم تحدث تغييرات كبيرة في البرنامج مثل الفرق بين **Visual Basic 5** و **Visual Basic 6** ، أما الرقم الثاني فهو يكون بعد التعديلات الطفيفة حيث نجد برنامج **Winzip 8** و **Winzip 8.1** مع فروق بسيطة .

أما الأخير فهو أسرعهم تغييراً مثل التحديثات وغير ذلك ، وفي برنامجنا سيكون افتراضياً ١,٠,٠ فإذا قمنا بتغيير كامل للبرنامج وازدادت كبراً جداً فإننا سنغير الرقم الأول ، أما إذا قمنا بإضافة تعدد الشروط للبحث وتقارير جديدة فنسغير الرقم الثاني ، أما إذا تداركنا مشكلة أو أضفنا إضافة بسيطة فسوف نغير الرقم الثالث .

بجانب هذه الشاشة سوف يطلب منك تحديد اسم المشروع ، واختيار أيقونته ، سوف نكتب (البائع الذكي) ونختار الأيقونة الخاصة بنا .

أسفل الشاشة سوف نحدد بعض معلومات البرنامج مثل اسم الشركة واسم المنتج ووصف المنتج وحقوق الطبع وغيرها .

بعد أن تكون قد أنهيت الخطوات في الدرس السابق ، سوف نقوم باختيار التبويب الثاني من شاشة Option والذي هو تحت مسمى Compile ، وسوف نرى خيارين رئيسيين هما :
Compile To P_Code و Compile To NativeCode .

Compile To P_Code

إذا قمت باختيار هذا العنصر فإنك تطلب من مترجم فيجوال بيسك خلال ترجمته لاسطر برنامجك بتحويلها إلى P_Code قبل تنفيذها ، ويتميز بأنه أصغر حجماً وأكثر توافقية مع الأخطاء .

Compile To NativeCode

برغم كبر حجم هذا النوع إلا أنه أسرع بكثير من النوع الأول ، ويحتوي على بعض الخيارات مثل Optimize for ast Code والذي يحاول قد المستطاع تسريع التنفيذ حتى ولو على حساب الحجم ، على عكس اللاحق Optimize for Small Code والذي سيصغر الحجم على حساب السرعة ... أما الخيار No Optimize فلا يقوم بتحديد شيء على حساب الآخر ، كما يمكنك وضع إشارة (صح) أما الاختيار (Favor Pentium Pro(tm) والذي يفيدك في تسريع العمليات الحسابية إذا كنت تعمل على بنتيوم بور ، أما الاختيار Create Sympolic Debug Info فإنه يعطي الملف فرصة لامكانية التنقيح لاحقاً باستخدام أي من البرامج المعدة لهذا الغرض .

: Advanced Optimizations

هذا الزر الموجود في أسفل الشاشة يقوم بعرض خيارات متقدمة لعملية Optimization ، ومنها :

Assume No Aliasing : وهو يقوم بتسريع البرنامج على افتراض عدم وجود عملية

استعارة Aliasing .

Remove Array Bound Check : ويقوم بعدم التحقق من أرقام Index للمصفوفة

مما يزيد التعامل معها .

Remove Floating Point Error Checks : حيث يزيل التأكد من أنواع الأرقام

ذات العلامة العائمة وهي التي تمثل بـ 2×10^5 أو 2×10^{-12} .

Allow Unrounded Floating Point Operations : وهي تبطل من سرعة

التنفيذ حيث تسمح بالحصول على دقة أكبر لهذا النوع من الأرقام .

Remove Safe Pentium(TM) FDIV Checks : ويقوم بإزالة التدقيق في

المعالجات التي تحتوي على مشكلة FDIV .

هل تريد معرفة المزيد عن مشكلة FDIV ، إن هذا الرابط ربما يكون مفيداً لك ، حيث تجد بعض

الأسئلة عن الموضوع :

<http://www.trnicely.net/pentbug/pentbug.html>

اجملاً في نهاية هذه الخيارات حاول عدم التعرض لها إلا إذا كنت متأكداً من أنك تحتاج إليها حتى لا تتسبب بأخطاء في برنامجك .

لن نعدل في هذه العناصر ، وبعد الانتهاء سوف نغلق الشاشة بالنقر على زر OK ثم نكتب اسم

البرنامج ، وننقر على Compile .

سوف يظهر شريط الترجمة ، حيث سيقوم بعدد من المهام منها التأكد من عدم وجود أخطاء أو

تعريفات خاطئة في الأوامر ، وبعد الانتهاء يتكون لدينا برنامج exe من البائع الذكي ، قم بوضع

قاعدة البيانات ، وملف التعليمات في نفس المسار وقم بتشغيله .

والآن قم بنقل البرنامج على جهاز لا يحتوي على Visual Basic وسترى أنه لا يعمل لأنه

تنقصه بعض الملفات ، إلا أنه سوف يعمل في حالة وجود مكتبات فيجوال بيسك ، ولكي يعمل

البرنامج على كل الأجهزة نقوم بعمل Setup للبرنامج ، وهو ما سنتعلمه في الدرس المقبل .

عمل عملية Setup

إذا قمت ببرمجة برنامجك باستخدام فيجوال بيسك ثم قمت بتحويله إلى تطبيق Exe ، ثم حاولت تشغيله على جهاز آخر فإنك لن تستطيع ، وإذا كان الجهاز يحتوي على نسخة من ملفات تشغيل الفيجوال بيسك فإنك ستواجه مشاكل فيما إذا كنت قد استخدمت أدوات غير افتراضية ، أو على الأقل ستعاني من نقل قاعدة البيانات مع البرنامج مع الملفات الملحقة كلما قررت تغيير مكان البرنامج .

أما التعليب – إن صحت الترجمة لكلمة Package – فإنه يمكنك من توفير ملف واحد يحتوي على جميع ملفات برنامجك ، وبمجرد تشغيله قوم تلقائياً بانزال كافة مكونات برنامجك على النظام ، ويضيف لك البرنامج في قائمة البرامج ، كما توفر برامج متقدمة لهذا الغرض توفير اختصار على سطح المكتب وبرمجة متقدمة للتحميل وإعادة تشغيل عند الطلب ، وكلمات مرور وتسجيل المنتج .

لن نتوسع في استخدام برنامج غير متوفر ، بل سنستخدم برنامجاً مرفقاً مع الفيجوال بيسك بصورة افتراضية ، وهو برنامج **Package & Deployment Wizard** والموجود ضمن قائمة الفيجوال بيسك تحت قسم Microsoft Visual Basic 6.0 .

والآن سنبدأ بالعمل على هذا البرنامج ، وبمجرد تشغيل البرنامج تظهر لنا شاشة تطلب منا في مقدمتها مسار البرنامج الذي نريد عمل Setup له ، وهناك نضغط على **Browse** ونختار البرنامج الخاص بنا . ثم نقوم بالضغط على **Package** .

يقوم البرنامج بفحص النسخة التي وضعناها ، فإذا لم يجد نسخة exe أو إذا وجد تعديلاً في النسخة عن ملف exe معها فإنه يطلب منك إرشاده إلى عمل **Compiling** من جديد أو الاحتفاظ بنسخة exe الحالية ، أو وضع مسار نسخة exe ترغب في استخدامها ، أما إذا لم تكن هناك فروق فإنه يستمر ، وينتقل للخطوة التالية .

في الخطوة التالية سوف يسألك فيما إذا كنت ترغب في استخدام خطوات سابقة ، أو الاختيار **None** ونستفيد من هذه الخاصية في حالة عمل **Setup** لنفس البرنامج أكثر من مرة ، حيث لا

تحتاج في كل مرة إلى اضافة الأدوات من جديد بل استخدم خطوات سابقة وسوف يضيفها لك تلقائياً .
مبدئياً سوف نختار None .

لاحقاً سوف نقوم بتحديد نوع Package الذي نرغب في القيام به ... وسوف نختار النوع الأول
Standerd Setup Package .

ثم يطلب منك البرنامج المكان الذي تود عمل ملف التنصيب فيه ، ضع أي مسار وقم لاحقاً بنقله كما
تحب ، كما تستطيع وضعه على شبكة Network . وفيما إذا كان المسار يحتوي على مجلد غير
موجود فسوف يخبرك بين انشاءه ، أو وضع مسار موجود أصلاً .

بعدها تظهر شاشة المكونات .

في هذه الشاشة سوف يقوم الفيچوال بيسك تلقائياً باضافة الملفات التي يتطلب برنامجك تشغيلها إلى
هذه القائمة ، وتستطيع الغاء ما لا تحتاجه منها عن طريق ازالة التحديد ، كما أنه سيكون عليك
اضافة العناصر التي تحتاجها وهي ليست موجودة في القائمة ، ويشمل ذلك ما يلي :

* ملفات قاعدة البيانات حيث نضغط زر Add ثم نختار All Files من أسفل الشاشة ونختار
ملف قاعدة البيانات .

* الخطوط غير الافتراضية ، حيث يتم تحميلها في جهاز المستخدم .

* الصور والأصوات وملفات الفيديو التي يحتاجها برنامجك .

* ملفات المساعدة والتعليمات مثل ملف Help.rtf مع برنامجنا .

* الأدوات ocx وملفات النظام dll والتي لا توجد في القائمة مثل استخدام أزرار أو قوائم خارجية .

* ملفات الريسحترى reg وغيرها .

وبعد اضافة ما نحتاجه وهو في برنامجنا فقط ملفي Help.rtf و db1.mdb نختار الانتقال
للخطوة التالية Next .

في الخطوة التالية يطلب منك تحديد طبيعة الملف وهل هو ملف واحد فقط بغض النظر عن حجمه أم
مقسم بحجم معين لوضعه على اقراص مرنة Disks مثلاً . وفي الخطوة اللاحقة سوف يطلب اليك
تحديد اسم لهذا التنصيب يظهر كعنوان له ، وسنختار "البائع الذكي" .

والآن ننتقل للخطوة التالية ، حيث سيطلب منك تحديد شكل ظهور البرنامج في قائمة البرامج ، حيث تستطيع اضافة عنصر جديد أو حتى مجموعة جديدة ، ولتجربة ذلك نقوم باضافة ملف التعليمات مع ملف البرنامج من خلال الضغط على زر **New Item** ، ونكتب في الخانة الأولى اسم الملف وليكن تعليمات البرنامج" ثم نختار الملف ومكان تشغيله من الخانتين اللاحقتين كما تستطيع تعديل خصائص أي عنصر من خلال الضغط على زر **Properties** .

في الشاشة التالية يحدد البرنامج أماكن وضع المكونات المحددة على جهاز المستخدم ، فتلاحظ وضع ملفات قاعدة البيانات والتعليمات في مسار البرنامج **AppPath** ، والخطوط في **Fonts** ، ومكونات النظام في **System** ، وتستطيع تغييرها إذا أردت .

بالضغط على **Next** سوف يخبرك البرنامج بأن هناك قائمة من المكونات يمكن استخدامها ك **Shared Files** وهي ملفات يمكن للبرامج الأخرى استخدامها مثل الخطوط وغيرها ، ونستفيد من ذلك التحديد أنه في حالة الحذف يتم التأكيد على حذف هذه المكونات أكثر من مرة نظراً لاحتمال وجود برامج أخرى تستخدمها . ولعلك لاحظت ذلك في خلال عمليات تشغيلك وحذفك لبرامج سابقة .

الخطوة الأخيرة هو تحديد اسم لهذه الخطوات التي قمت بها ، ومن خلال هذا الاسم تستطيع تجنب تكرار كل هذه الخطوات مع كل مرة ، حيث تكون موجودة افتراضياً مع امكانية التعديل فيها بالطبع أخيراً قم بالضغط على **Finish** وهكذا تكون قد انتهيت من عمل ملف **Setup** لبرنامجك .

في نهاية درسنا عن عمل **Setup** للبرنامج سأشير إلى برنامجين هما من أشهر البرامج في مجال عمل **Setup** لبرامج الفيجوال بيسك وغيرها ، هما برنامجا **Setup Factory** وبرنامج **Install Shild** .

يحتوي هذان البرنامجان على مزايا اضافية عديدة ، مثل تحديد الشاشات وعددها وأنواعها وتصميمها ، اختيار عمل اختصارات على سطح المكتب وخلافه ، تحكم كامل في الأدوات والملفات المرفقة ، التسجيل في الريجستري ووضع عناوين وإزالة مفاتيح وخلافه ، اضافة كلمات مرور وانواع **install** ، اتاحة كتابة تعليمات برمجية بسيطة حيث يحتوي كل منهما على **Engine** خاص به لتفسير بعض الأوامر ، حيث يمكنك انشاء متغيرات والتأكد من قيمها وخلافه .

مشكلة تحويل البرنامج إلى كود مرة أخرى .

تتوفر في الإنترنت برامج متنوعة تقوم بتحويل التطبيقات إلى ملفات بالكود من جديد ما يعني ضياع حق المبرمج بسهولة ، وعملياً لا يوجد حل كامل لهذه المشكلة ، لأن الطريقة العكسية لعمل ال Compiling هي التي تفك الضغط ، وبذا يعود البرنامج كما كان ، وتفاقم الامر مع الاصدار net. من الفيجوال بيسك نظراً للطريقة التي تعتمد عليها البرامج المصنوعة تحت بيئة التطوير هذه ، ولذا تقوم مايكروسوفت دورياً بوضع برامج تشفير - إن صح التعبير - تقوم بتعصيب هذه العملية ، وكما توفر بعض الشركات برامج وتقنيات للحفاظ على حقوق المبرمج ، ولكن العلاقة مستمرة في شد وجذب بصورة مستمرة .

هناك أيضاً اتفاقيات البرمجيات مفتوحة المصدر مثل GPL والتي تعني Genral Public Licence والتي تأتي مع أغلب برمجيات المفتوحة المصدر ، باختصار كل ما في الموضوع هو انك تقوم بتوفير برنامجك مع ال Source Code الخاص به للجميع ، ويمكنك أن تطلب ثمناً لذلك لكن بشرط اعطاء الكود مع البرنامج ، يمكن التعديل عليه واطافة مزايا أخرى مما يجعل تطوير البرنامج أسرع ، لا تقلق من ذلك فلن يضيع حقك بكل تأكيد ، ولن تجد شخصاً يحاول أن يكسر حماية برنامجك على أي حال . من نظم التشغيل التي تتبنى هذا الاتجاه عائلة Unix أيضاً من لغات البرمجة التي تسير في هذا المنهاج Java و PHP . ومن البرامج الشهيرة برامج مثل متصفحات Mozilla و Firefox سيرفرات Apache وغيرها .

تشغيل برنامج التنصيب تلقائياً بمجرد ادخال الاسطوانة .

سيكون هذا الموضوع الخفيف هو ختام دروسنا في هذه السلسلة إن شاء الله .

لعمل هذه الخطوة البسيطة لا يلزمنا سوى فتح مستند نصي فارغ . غير الاسم إلى :
AUTORUN.INF .

والآن قم بفتح هذا الملف بواسطة المفكرة NotePad ، وأكتب هذين السطرين بداخله :

[autorun]
open=Setup.exe

وانسخها وضعها مع البرنامج في نفس مسار الملف Setup.exe .
وهذه الطريقة تصلح مع جميع انواع الملفات وليس التطبيقات فحسب .

بهذا نكون بحمد الله قد اتممنا هذا الجزء من الدروة وإلى
اللقاء في الجزء القادم مع تطوير مواقع الانترنت .
تم نشره في منتديات فيجوال بيسك : ٢٠٠٣-٢٠٠٤ .