مقدمة في برمجة حواسيب الكف

الإصدار الأول
23/04/2008

بن العيد

بسم الله الرحمن الرحيم

قال رسول الله صلى الله عليه و سلم : "إذا مات الإنسان انقطع عنه عمله إلا من ثلاثة: إلا من صدقة جارية. أو علم ينتفع به. أو ولد صالح يدعو له" .

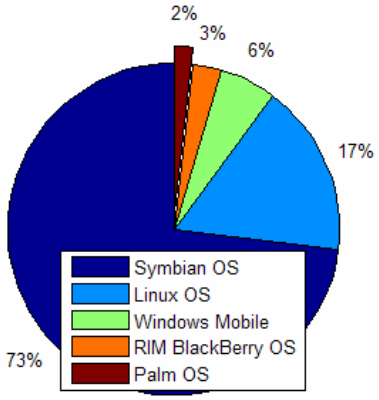
رواه مسلم.

الفهرس

الجزء الأول: مقدمة عامة

1. مقدمة

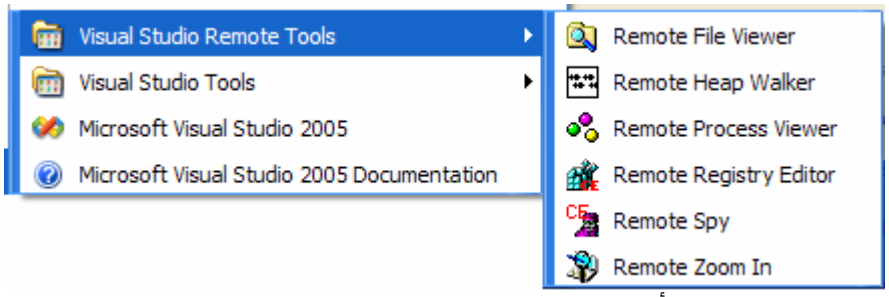
هناك عدة مقاييس لتعريف الهاتف الذكي وتتضمن الشاشة التي تعتمد اللمس، نظام التشغيل، وقدرات المودم كمقايس أساسية. خدمة البريد الإلكتروني يمكن أن تعتبرها أيضا من مقاييس التعريف. أغلب الهواتف الذكية توفر إمكانية تنصيب برامج إضافية، حتى من قبل مطورين/مبرمجين مستقلين ولكن المنتجين يصرون على التسمية "هواتف ذكية" حتى مع عدم توفير هذه الإمكانية. هناك سعي لإدماج خاصية الإبحار في شبكة الأنترنت، إستعمال البريد الإلكتروني، التصرف في قائمة الإتصال، وعدة برامج أخرى كقراءة ملفات مثلPDF و مايكروسوف أوفيس للهواتف الذكية. ليس كافة الهواتف الذكية و حواسيب الكف تعتمد نظام التشغيل الويندوز بل هي تمثل 5.6% فقط من الأجهزة المتوفرة في الأسواق. حيث نجد الهواتف المعتمدة على نظام التشغيل سيمبيان في الصدارة الترتيب كما هو مجسد في الرسم البياني التالي:



الصورة رقم 1: أبرز نظم التشغيل للهواتف الذكية حسب نسبة تقاسم السوق

2. الأدوات الفرعية للتعامل مع حواسيب الكف و الهواتف الذكية

توفر عدة البرمجة الخاصة بحواسيب الكف و الهواتف الذكية عدة أدوات تيسر تطوير التطبيقات في نظام التشغيل الويندوز موبايل. في هذا الجزء أقدم بعضا من هاته الأدوات و وضائفها علما أن في [5] توجد القائمة كاملة مع تقديم واف.

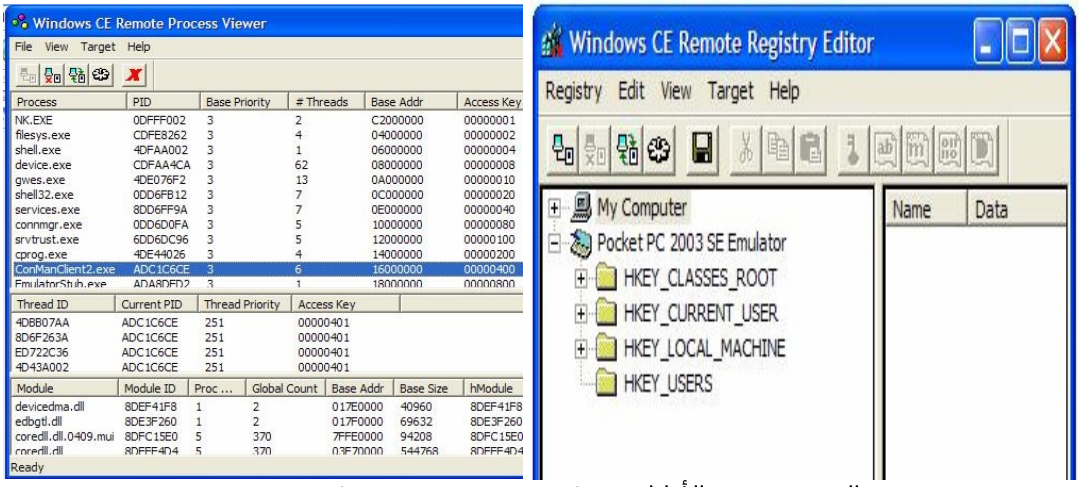


الصورة رقم 2: الأدوات الفرعية للتعامل مع حواسيب الكف و الهواتف الذكية

1.2. Remote Registry

هذه الأداة تماثل في باقي الحواسيب ولها نفس الوضائف وينبغي الحذر الشديد عند التعامل مع بيانات التي توفرها هذه الأداة كي لا يحدث خلل في عمل نظام التشغيل في حاسوب الكف أو الهاتف.

بما أنه يوجد دائما محبي المجازفة، فأنصح بحفظ نسخة من التعديل الأصلي لل Registry قبل إحداث أي تغير لكي تسهل العودة الى الوضع الأصلي وذلك عبر إختيار Registry -> Export Registry File...



الصورة رقم 3: الأداتان Remote Registry و Process viewer

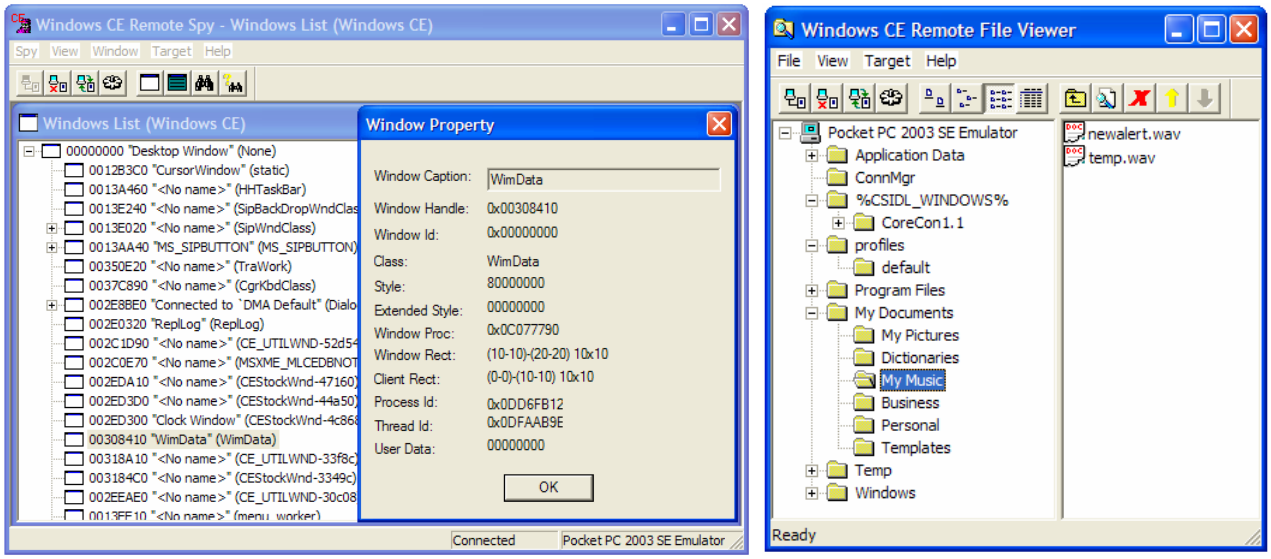
مفتاح الريجيستري	الوضيفة
HKEY_CLASSES_ROOT	ربط الملفات بالبرمجيات.
HKEY_CURRENT_USER	تحتوي معلومات حول المستخدم المتصل بالجهاز المعني.
HKEY_LOCAL_MACHINE	تحتوي معلومات حول إعدادات الجهاز المتصل.

2.2. Process viewer

يمكن من الإطلاع على ملفات التشغيل النشطة في حاسوب الكف أو المحاكي، مكان تواجده، المكتبات التي تدعمها ومزيد من التفاصيل الأخرى مع إمكانية إيقاف أحد هذه الملفات عند توفر الصلاحيات الكافية للقيام بذلك.

3.2. File viewer

تمكن من تصفح المجلدات الموجودة في حاسوب الكف أو المحاكي والإطلاع على الملفات مع إمكانية فسخها و تبادل الملفات مع حاسوب المكتب. لإستعمال هذه الأداة لابد من تأسيس الربط أولا بين حاسوب الكف أو المحاكي و حاسوب المكتب.

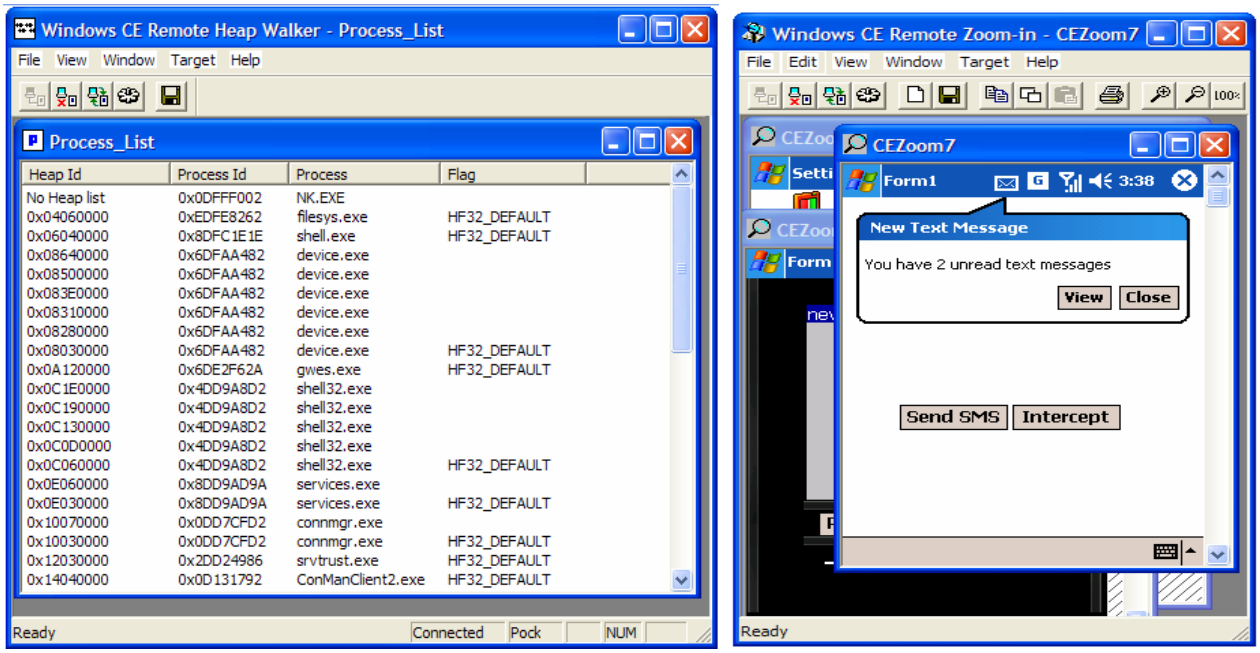


الصورة رقم 4: الأدتان File viewer و Remote Spy

إثر تحقيق الربط بين حاسوب الكف/المحاكي و حاسوب المكتب يكون هناك تبادل لرسائل بين الطرفين. يمكن Remote Spy من تتبع هذه الرسائل.

5.2. Remote Zoom In

تمكن هذه الأداة من أخذ أكثر من صورة من نوع bitmap لواجهة المحاكي/الجهاز شرط أن يكون هناك إتصال بين هذا الأخير و حاسوب المكتب. لمعرفة كيفية ربط الجهاز أو المحاكي راجع الجزء الثاني من هذا الكتاب.



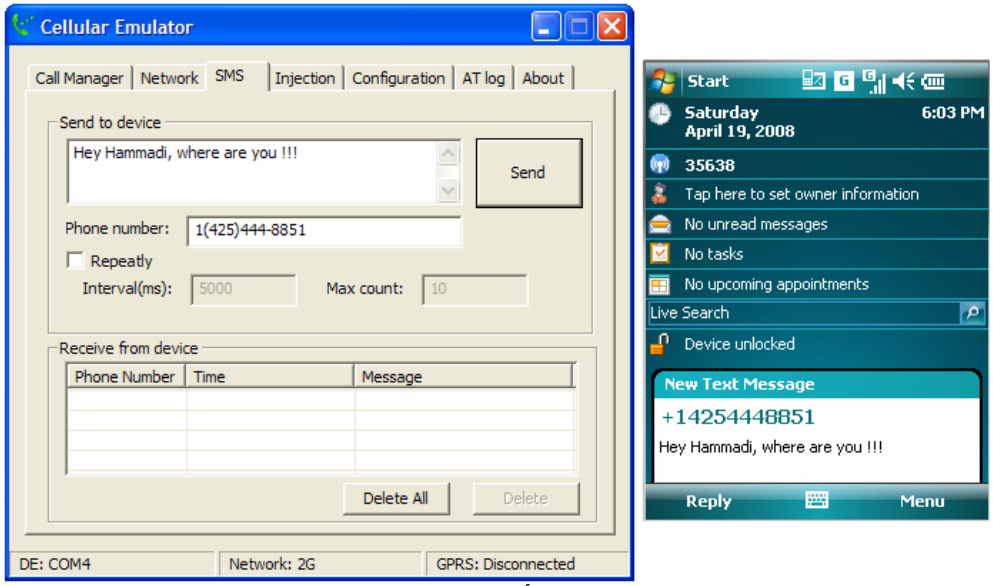
الصورة رقم 5: الأداة Remote Zoom In

6.2. Remote Heap Walker

لتوفير تفاصيل دقيقة حول ملفات التشغيل النشطة (Active Processes).

7.2. Cellular Emulator

هذه الأداة يوفرها حاسوب الكف من جيل 2006 و تمكن من محاكاة شبكة الهاتف الجوال و تيسر تطوير التطبيقات المتعلقة بإستقبال المكالمات و الرسائل القصيرة و الوضائف المتعلقة بال AT-Commands.



الصورة رقم 6: الأداة Cellular Emulator

3. عدة البرمجة و الأدوات اللازمة

كافة الأدوات و عدة البرمجة التي سنحتاجها في تجربة التطبيقات و المضي قدما في برمجة حواسيب الكف و الهواتف الذكية متوفرة في الجدول التالي:

	العدة/البرنامج	الرابط
1	Windows Mobile 5.0 SDK for Smartphone	http://www.microsoft.com/downloads/details.aspx?familyid=DC6C00CB-738A-4B97-8910-5CD29AB5F8D9&displaylang=en
2	Windows Mobile 5.0 SDK for Pocket PC	http://www.microsoft.com/downloads/details.aspx?FamilyID=83a52af2-f524-4ec5-9155-717cbe5d25ed&displaylang=en&Hash=ROflLpbIDX4DtputncKa1TrWH%2bqUwXpQhBH7J9v%2fqqEUgCO4hQ%2boRCnbGK6SCJHmbrq1d4vz3NI5aIZ89ZpXwA%3d%3d
3	Windows Mobile 6 Professional and Kits Standard Software Development Refresh	http://www.microsoft.com/downloads/details.aspx?familyid=06111a3a-a651-4745-88ef-3d48091a390b&displaylang=en
4	SQL Server 2000 Windows C Edition version 2.0	http://www.microsoft.com/downloads/details.aspx?FamilyID=b9b12312-fe57-4817-a4bc-69992802732d&DisplayLang=en
5	ActiveSync	http://www.microsoft.com/windowsmobile/activesync/default.mspx
6	windows mobile device center	http://www.microsoft.com/windowsmobile/devicecenter.mspx

4. خاتمة

إثر هذا الإستعراض الموجز لبعض الأدوات التي تيسر التعامل مع حاسوب الكف و الهواتف الذكية، يصبح للمطور المزيد من الحرية و الصلاحيات في تطوير تطبيقات متنوعة متجاوزة لحدود المحاكي. الجزء الموالي يهتم بتقديم بيئة التطوير و أنواع التطبيقات التي يمكن للمبرمج إنشائها، علما أنه سيتم التركيز على المحاكي عوض الجهاز الحقيقي لتوفير إمكانية المتابعة و التعلم لمن لا يملك حاسوب كف أيضا.

الجزء الثاني: الخطوات الأولى في تعلم برمجة حواسيب الكف

1. مقدمة

يخص هذا الجزء المراحل الأولى في إنشاء تطبيق للهواتف الذكية و حواسيب الكف، و قبل البدء ينبغي تحميل البرنامج ActiveSync إذا كان نظام تشغيل حاسوب المكتب Windows أما إذا كان Windows Vista فينبغي تحميل البرنامج windows mobile device center. رابطا تحميل هذين البرنامجين متوفران في الجدول رقم 1 في عدة البرمجة و الأدوات اللازمة.

2. إنشاء مشروع جديد

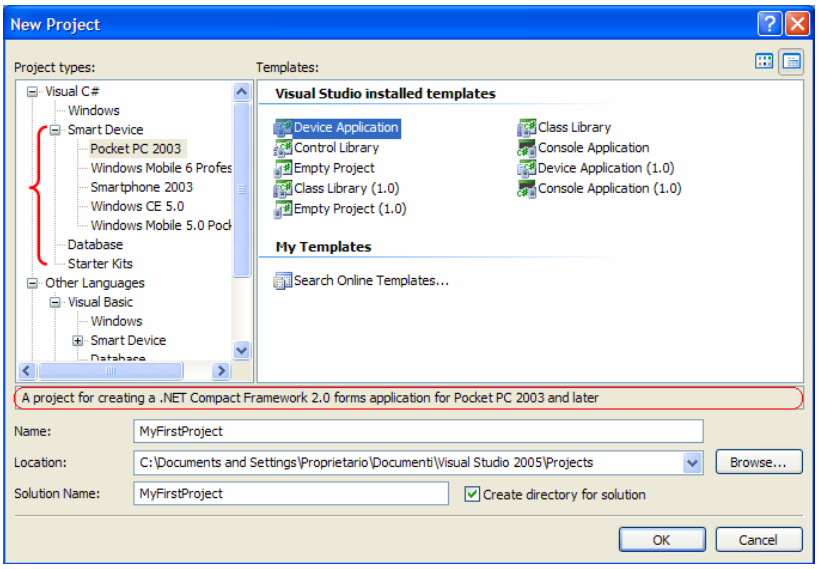
أعتمد على الفيزوال ستوديو إصدار 2005. و للأسف أن النسخة الخفيفة من إصدار 2008 لا توفر اللوازم لتطوير برمجيات موجهة لحواسيب الكف، لذلك ينغي الحصول على نسخة كاملة من هذا الإصدار أو ما سبقه.

إذا كانت البرمجيات التي تنوي تطويرها تخص حواسيب الكف و الهواتف الذكية من جيل 2003 فلن تحتاج لتنصيب عدة البرمجة الخاصة به لأنها متوفرة في برنامج التطوير لكن ستحتاج لتنصيب إحدى إصدارات منصة التطوير Microsoft .Net Framework.

- بيئة الدوت نت

رغم إختلاف لغات الدوت نت فإن مراحل إنشاء مشروع جديد متشابهة حسب نوع التطبيق. في المعقف الأحمر في الصورة رقم 1 أبرزت نظم التشغيل التي يمكن إعتمادها والتي تتطلب إضافة عدة تطوير البرمجيات1 الخاصة بجيل 2005 من حواسيب الكف والهواتف الذكية و 2006 لحواسيب الكف و في الإطار الأحمر معلومات حول نوع المشروع الذي تم إختياره.

1 SDK: Software Development Kit



الصورة رقم 1: إنشاء مشروع جديد لحاسوب الكف في بيئة الدوت نت

Device Application (1.0)

تهم التطبيقات ذات واجهة/واجهات رسومية حسب لغة الدوت نت المنتمي لها التطبيق المطور والإختلاف بينهما في المكتبات المعتمدة عليها وجيلها والتي تسمىNet compact FrameWork.

Console Application (1.0)

تهم التطبيقات التي تعمل في الخلفية أي دون وجود واجهات رسومية.

هذا النوع من التطبيقات يماثل تلك التي ترجع الشاشة السوداء في نظام التشغيل الويندوز العادي لكن في حواسيب الكف لاتوجد تلك الشاشة ويستحسن الإعتماد على الملفات لتتبع سير البرنامج.

والإختلاف بين هاتين التطبيقات هو جيل المكتبات المعتمد عليها.

Control Library

هذا النوع من التطبيقات يوفر مكتبة ديناميكية[2] تحتوي واجهة رسومية، مكونات[3] ومجالات[4] يمكن إستغلالها في تطبيقات أخرى

2 DLL: Dynamic Library Link
3 Components
4 NameSpaces

Class Library 1.0

هذا النوع من التطبيقات يوفر أيضا مكتبة ديناميكية تساعد في تعليب جزء من الكود وإخفاءه ليتم إستغلاله في تطبيقات أخرى والإختلاف بين هذين النوعين من التطبيقات هو جيل ال Net compact Framework.

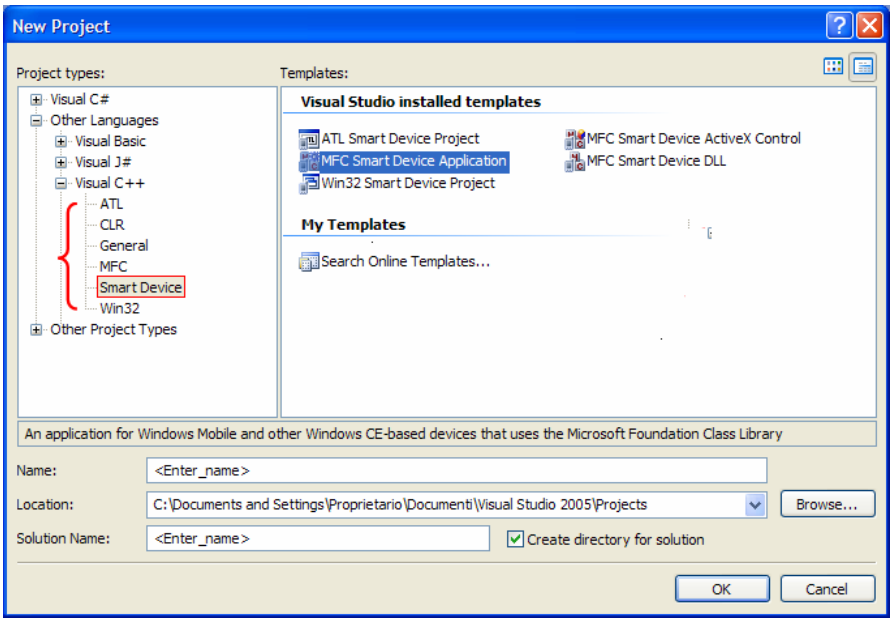
Empty Project

يوفر مشروع فارغ تكون نتيجته مرتبطة بنوعية الملفات المضافة وتنسيق الكود للحصول على تطبيق من إحدى التطبيقات السابقة أو مزج بينها.

لو أعدنا النظر الى محتويات المعقف الأحمر في الصورة رقم 2 ربما يخطر في بالنا سؤال وهو :مالفرق بين Windows Mobile و Windows CE؟ الإجابة أن الـWindows Mobile يعتمد على Windows CE وهو الجزء المركزي فيه. فال WM 2003 يرتكز على WM 5.0 , WinCE 4.x يرتكز على الإصدار الخامس لل Win CE مع إضافة بعض التطبيقات APIs و Standard shell.

- بيئة السي و السي ++

توفر هذه البيئة إمكانيات كبيرة في تطوير البرامج لحواسيب الكف والهواتف الذكية والكثير من الوضائف التي تعجز بيئة الدوت نت على توفيرها. ما يهمنا هو القسم Smart Device وفيه يمكن تطوير برامج ذات واجهة/واجهات رسومية MFC بإختيار MFC Smart Device Application أو تطبيقات من نوع "شاشة سوداء" و مكتبات ديناميكية عند إختيار Win32 Smart Device Project.



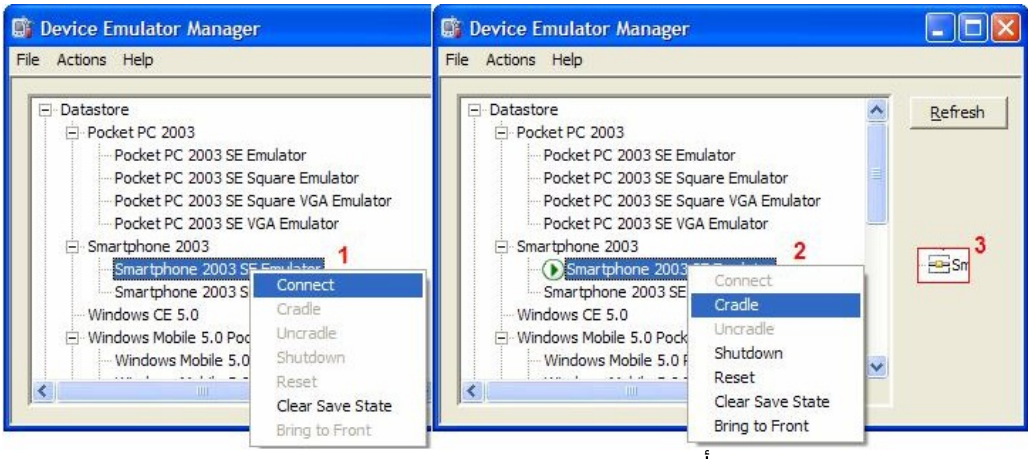
الصورة رقم 2: إنشاء مشروع جديد لحاسوب الكف في بيئة السي والسي++

ملاحظة: يستعمل الجافا أيضا لبرمجة حواسيب الكف والهواتف الذكية ويعتبر Eclipse من أبرز ال IDE الذي ييسر ذلك.

3. المتصرف في محاكي الجهاز

من جملة الأدوات التي يوفرها برنامج التطوير الفيزوال ستوديو نجد المتصرف في محاكي الجهاز[5] (حاسوب الكف أو الهاتف الذكي), يسر هذا الأخير تطوير البرمجيات حتى لو لم يكن متوفر للمبرمج جهاز حقيقي يطور فيه تطبيقاته. و للمحاكي عدة ميزات و وضائف تجعل الفرق بينه و بين الجهاز بسيطة و لا تظهر نقائصه إلا في بعض التطبيقات المتقدمة كإستخدام الكاميرا و شبكة الهاتف الجوال.

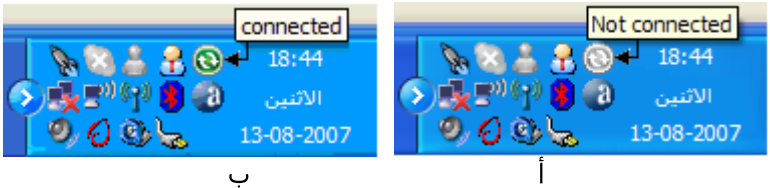
لتشغيل المحاكي، ينبغي تشغيل المتصرف في محاكي الجهاز أولا من قائمة الأدوات في الفيزوال ستوديو (Tools) ثم يتم إختيار المحاكي المعني و جيله و نوعه. في الصورة رقم 3 إخترت محاكي حاسوب الكف 2003. الإصدار الثاني[6].



ب أ

الصورة رقم 3: مراحل الربط

في نهاية الربط بين المحاكي و حاسوب المكتب تتغير أيقونة ال ActiveSync من الرمادي إلى الأخضر مم ييسر التأكد من حال الربط طوال فترة التطوير، لأنه ينقطع أحيانا الإتصال بين الطرفين.



ب أ

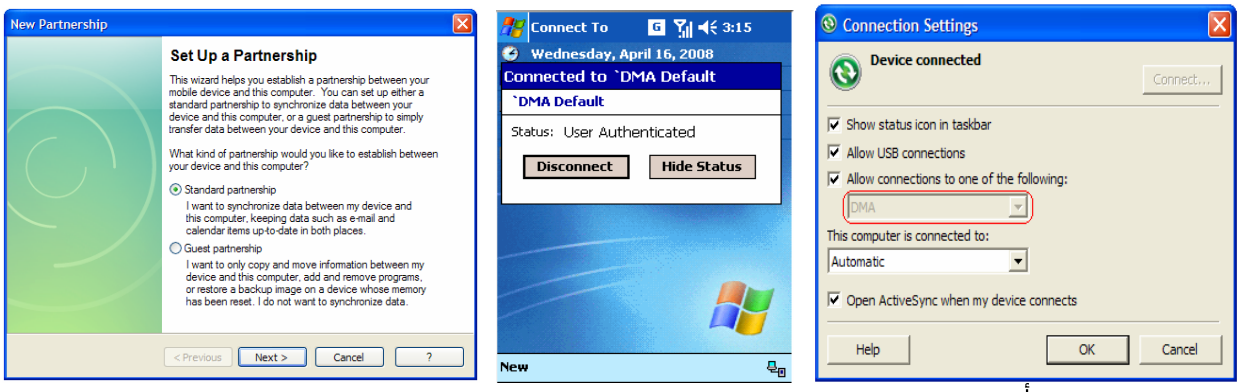
الصورة رقم 4: ActiveSync قبل و بعد الربط

[5] المتصرف في محاكي الجهاز: Device Emulator Manager
[6] الإصدار الثاني: SE: Second Editino

تحديد نوعية الربط مع حاسوب الكف يختلف حسب إحتياجات البرنامج الذي سيتم تطويره. إذا كان التطبيق لا يحتاج لنقل ملفات داخل حاسوب الكف أو المحاكي فإن الإختيار connect يكفي لتجربة البرنامج دوريا حتى إنهائه. أما إذا كانت هناك حاجة لنقل ملفات من حاسوب المكتب إلى حاسوب الكف أو المحاكي فينبغي تفعيل الإختيار connect ثم cradle حتى يكون هناك إتصال بين الطرفين. و ليس ضروري في كلتا الحالتين السابقتين إختيار نوع العلاقة Standard partnership، إلا إذا كانت هناك حاجة لمزامنة بين حاسوب المكتب و حاسوب الكف. هذه المزامنة تخص برامج معينة يحددها برنامج الربط ActiveSync لاحقا و هي Calendar, Contacts, Inbox .

إثر إختيار نوع الربط و تفعيل الزر يتم تأسيس الربط بين المحاكي و حاسوب الكف. هذا الربط يتم تلقائيا عند ربط الجهاز الحقيقي مع الحاسوب إذا تم تفعيل الإختيار

"Open ActiveSync when my device connects" في واجهة البرنامج كما هو في الصورة رقم 5.



ج ب أ

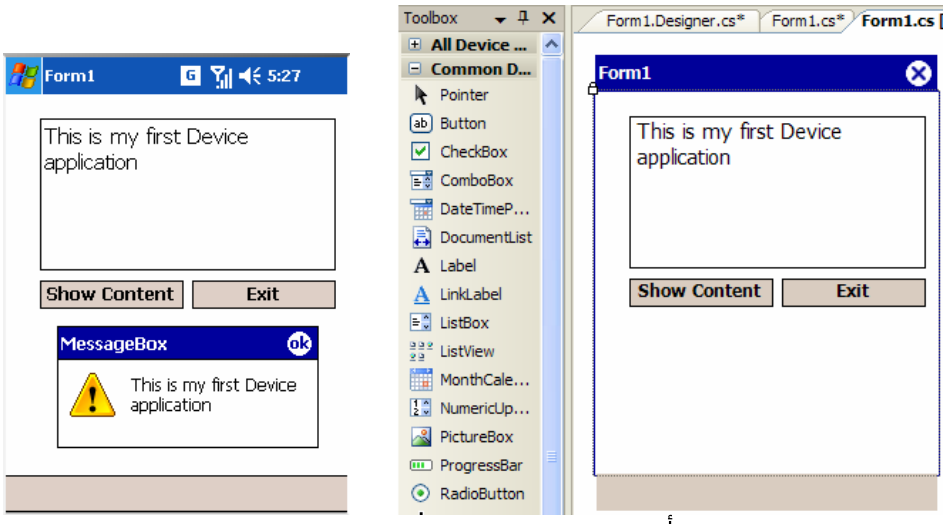
الصورة رقم 5: تأسيس الربط مع المحاكي

إثر إنشاء التطبيق الأول من نوع Device Application لحواسيب الكف من جيل 2003، سننشئ أول تطبيق لنا في نظام التشغل Windows Mobile.

من قائمة الأدوات[7] قم بإضافة زرين و مجال لكتابة نص كما هو في الصورة رقم 9 ثم غير إعدادات هاته المكونات كما هو مجسد في الجدول أسفله.

نوع المكون	الإسم	النص
Button	btn_ShowMsg	Show Content
Button	btn_Exit	Exit
TextBox	txb_InputText	This is my first Device application

[7] Toolbox



ب أ

الصورة رقم 6: الواجهة الرسومية قبل و بعد الترجمة

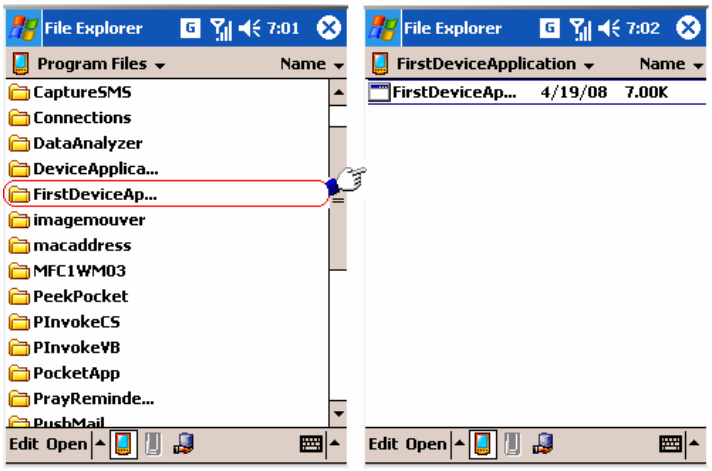
قمت بتعطيل خاصية إظهار الواجهة كاملة لحاسوب الكف و إكتفيت بالواجهة الرسومية أين أضع لوازم التطوير. للقيام بذلك عدل الخاصية Skin من قائمة خصائص الواجهة إلى false . إثر النقر مزدوجا على كل من الزرين قم بنسخ الشيفرة التالية, حيث أن وضيفة أحد الزرين نسخ محتويات النص المدخل و إظهاره في رسالة و الزر الثاني يغلق التطبيق.

```
private void btn_Exit_Click(object sender, EventArgs e)
  { Application.Exit();  }

private void btn_ShowMsg_Click(object sender, EventArgs e)
  {
    MessageBox.Show(txb_InputText.Text,
                    "MessageBox",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Exclamation,
                    MessageBoxDefaultButton.Button1);
  }
```

يمكن تقسيم الترجمة في حاسوب الكف إلى مرحلتين:

1. Build Solution: يتمثل في ترجمة التطبيق مع الأخذ بعين الإعتبار متطلبات نظام التشغيل و جيل عدة تطوير البرمجيات و يتم إنشاء ملف التشغيل في المجلد Bin\Debug في مسار إنشاء التطبيق. و هي مراحل مشابهة للتطبيقات التي تخص حاسوب المكتب.
2. Deploy Solution: يتمثل في إنشاء ملف بإسم التطبيق في المسار Program\ Filesلحاسوب الكف أو المحاكي ينسخ فيه ملف التشغيل الذي تم إنشائه في المرحلة السابقة (الصورة رقم 7).

أحيانا عند تطوير المشاريع الضخمة أو التي تخص حواسيب الكف من جيل 2006، تكون المرحلة الثانية بطيئة جدا و يصبح نقل ملف التشغيل يدويا أسرع.



الصورة رقم 7: إنشاء مجلد التطبيق في حاسوب الكف

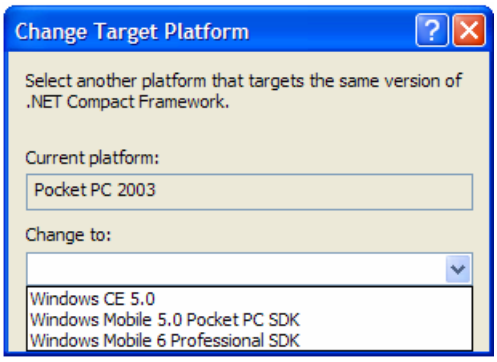
يمكن تغير منصة تطوير البرنامج إثر إنشاء مشروع و الشروع فيه عبر إختيار Change Target Platform في قائمة الإختيارات في الفيزوال ستوديو كما هو في الصورة رقم 8.

لا أنصح بهذا التحويل لأنه يولد أحيانا مشاكل خاصة عندما يكون بين التطبيقات الموجهة للهواتف الذكية و حواسيب الكف. فعدة البرمجة الموجهة (SDK: Software Development Kit) لكل منهما مختلفة إلا حد ما، إضافة إلى إختلاف كبير بين الدوال الأصلية. عامل آخر لتجنب هذا التحويل الآلي يتمثل في إختلاف جيل التطبيق، فعدة البرمجة الموجهة لحواسيب الكف من جيل 2005 و 2006 أكثر ثراء من سابقاتها و توفر عدة إختصارات في الشفرة إضافة إلى غياب بعض المكتبات.

مثال لو أردنا الإطلاع برمجيا ما إن كان هناك ربط بين حاسوب الكف أو المحاكي في تطبيق يخص جيل 2005 أو 2006 فهذا السطر كاف لإنجاز المهمة:

```
label_ActiveSyncStatus.Text=
string.Format(
            "ActiveSync Status: {0}.", SystemState.CradlePresent.ToString());
```

لكن المجال Microsoft.WindowsMobile.Status غير معروف في جيل 2003، لذلك فالتحويل الآلي لن يكون مجديا هنا و لابد من البحث على مقابل تلك الوضيفة مثل البحث ما إن كان rapimgr.exeنشط أم لا في حاسوب الكف أو المحاكي أثناء تشغيل التطبيق.



الصورة رقم 8: تغير منصة التطوير

4. خاتمة

وهكذا ينتهي هذا الدرس و قد أبرزت الخطوات الأولى لتعلم برمجة حواسيب الكف و الهواتف الذكية.
تم التركيز في هذا الدرس على المحاكي عوض الجهاز الحقيقي لكي تكون محتويات هذا الدرس في متناول الجميع حتى من لا يملك هاتف ذكي أو حاسوب كف.
في الأجزاء الاحقة من الكتاب مراوحة بين الجانب النظري و التطبيقي للبرمجة في حواسيب الكف.
اللغة الأساسية المعتمدة في التطبيقات هي السي شارب و بما أنها شبيهة إلى حد ما ب VB.NET فسأترك لكم عملية التحويل.

الجزء الثالث: التعامل مع المجلدات و الملفات و الذاكرة

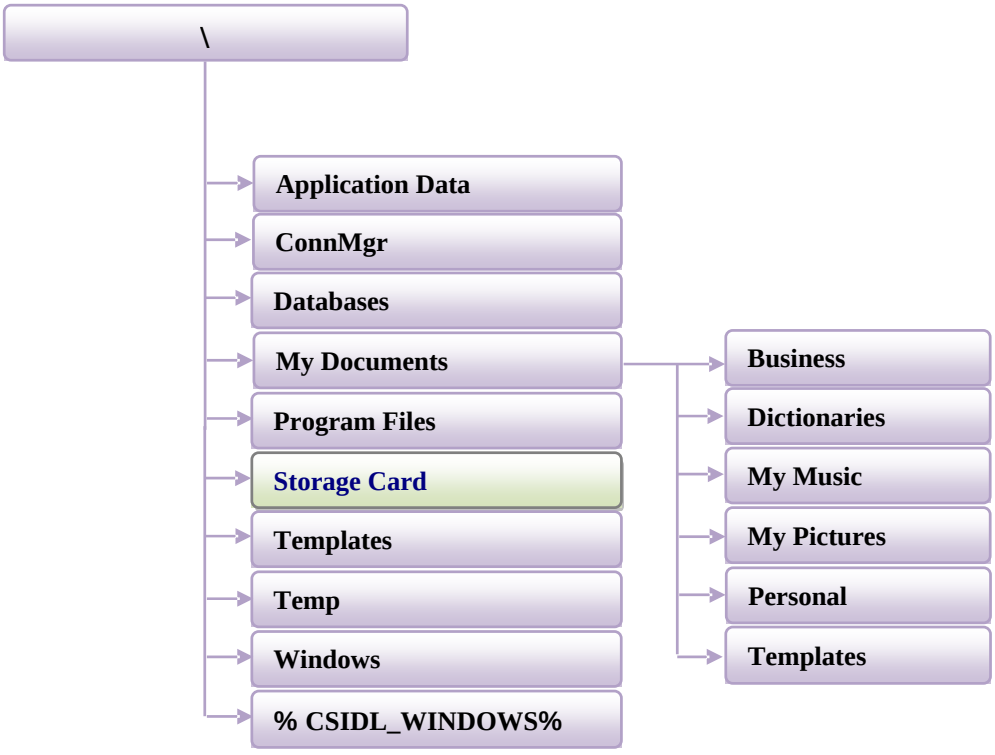
1. مقدمة

من الأسئلة الشائعة في البرمجة في حواسيب الكف و الهواتف الذكية، كيفية تحديد مسارات الملفات و المجلدات بما أنه لا يوجد تقسيمات للقرص الصلب بشكل مماثل لنظم التشغيل الويندوز الخاصة بحواسيب المكتب. لذلك أخصص هذا الجزء لشرح كيفية التعامل مع المجلدات و الملفات و الريجيستري لحاسوب الكف.

2. التعامل مع المجلدات و الملفات

لايوجد تقسيمات لحاسوب الكف كما هو في حاسوب المكتب حيث تبدأ تسمية المسار إنطلاقا من C:\\ مثلا, إنما تحديد المسارات يكون إنطلاقا من المجلد الرئيسي لحاسوب الكف وهو المجلد المرموز له ب \. يمكن تشبيه هذا المجلد بالمسار ROOT في نظام التشغيل يونيكس.

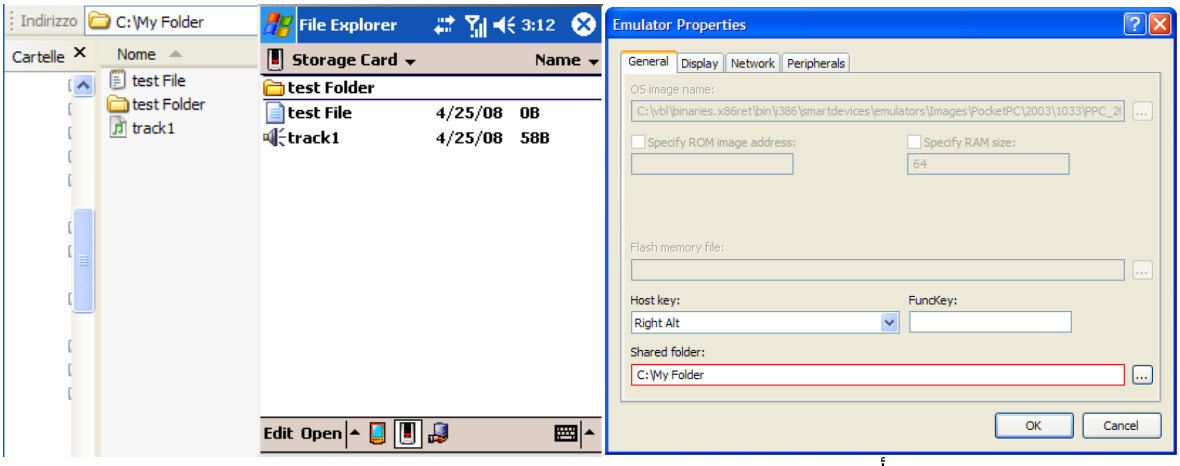
الرسم التالي يجسد تفرع المجلدات في حاسوب الكف إنطلاقا من المجلد الرئيسي مم ييسر على المطور تحديد المسارات في تطبيقاته.



الصورة رقم 1: معمارية المجلدات الرئيسية

قمت بتجسيد المجلد Storage Card بطريقة مغايرة لبقية المجلدات. هذا المجلد يخص بطاقة الذاكرة الخارجية التي يمكن إضافتها لحاسوب الكف لزيادة مساحة الذاكرة أو لأغراض أخرى. و لكن ماذا لو كان المطور يعتمد المحاكي في تطوير تطبيقاته، هل يمكن زيادة مساحة الذاكرة؟

أجل من الممكن الزيادة في مساحة ذاكرة المحاكي بأضعاف أضعافها و زد على ذلك ما شئت حتى تبلغ المساحة المتبقية من القرص الصلب لحاسوب المكتب. يكون ذلك عبر إستخدام مجلد في حاسوب المكتب كبطاقة ذاكرة خارجية للمحاكي من خلال تحديد المسار الكامل للمجلد في إعدادات المحاكي كما هو مبرز في الصورة رقم 2.



ب أ

الصورة رقم 2: إضافة بطاقة شحن للمحاكي

يمكن فتح الواجهة السابقة من قائمة الإختيارات للنافذة التي تحتوي المحاكي: File\Configure... هاته الواجهة تحتوي عدة إعدادات أخرى للمحاكي تساعد في تقريب وضائف المحاكي للجهاز الحقيقي مثل مستوى البطارية، الشبكة ...

المسار التلقائي لإنشاء الملفات و المجلدات في التطبيقات الخاصة بحاسوب المكتب و المطورة بالسي شارب هو المجلد bin\Debug التابع للمشروع أي نفس المجلد الذي ينشأ فيه ملف التشغيل. المسار التلقائي في حاسوب الكف هو متصفح الملفات[8]. للتعامل مع الملفات أو المجلدات المتواجدة في نفس مسار ملف التشغيل المتواجد في حاسوب الكف أو المحاكي فيمكن تعريف المسار بالكيفية التالية:

```
string directoryName = System.IO.Path.GetDirectoryName(
                       Assembly.GetExecutingAssembly().GetName().CodeBase);
MessageBox.Show(directoryName);
```

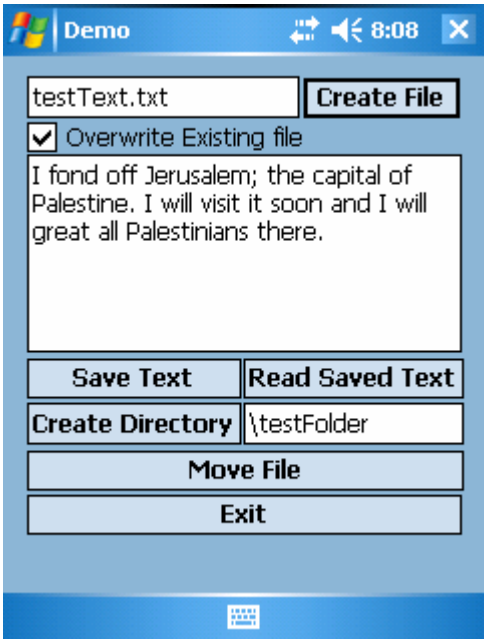
[8] File Explorer

3. التطبيق

الغاية من هذا التطبيق تيسير التعامل مع الملفات و المجلدات في حواسيب الكف لتسهيل تحديد المسارات. التطبيق مطور بلغة السي شارب و سأكتفي بوضع صورة للواجهة الرئيسية و الدوال الرئيسية دون إرفاق ملف الشفرة المصدرية للتطبيق.

أهداف التطبيق التعرف على كيفية:

- إنشاء ملف/مجلد جديد في مسار يحدده المستخدم.
- القراءة، الكتابة و نقل الملف المنشئ.



الصورة رقم 3 : الواجهة الرئيسية للتطبيق

لتيسير متابعة الشفرة و إستعمالها ألخص في هذا الجدول جملة المكونات، أسمائها و النص المسند لكل منها:

نوع المكون	الإسم	النص
Button	btn_CreateFile	Create File
Button	btn_SaveTXT	Save Text
Button	btn_ReadSavedTxt	Read Saved Text
Button	btn_createDir	Create Directory
Button	btn_move	Move File
Button	btn_Exit	Exit
TextBox	txt_CreateFile	testText.txt
TextBox	txt_Area	I fond off Jerusalem…
TextBox	txt_NewDir	\testFolder
CheckBox	chk_Overwrite	Overwrite Existing file

من نافذة الإختيارات يمكن تغير خصائص المكونات كلون الأزرار و خلفية الواجهة أو يمكن القيام بذلك برمجيا. مثال:

```
this.BackColor = System.Drawing.SystemColors.InactiveCaption;
```

الجدول التالي يحوصل أسماء الدوال التي أستعملتها في هذا التطبيق:

إسم الدالة	الوضيفة
CreateFile	إنشاء ملف جديد
SaveContent	حفض النص
ReadContent	قراءة محتويات الملف
Create_NewDir	إنشاء مجلد جديد
moveFile	نقل الملف

```
private void SaveContent()
{
    File_Full_Path = txt_CreateFile.Text;
    StreamWriter sw = null;

      if (File.Exists(File_Full_Path))
       {  sw = new StreamWriter(File_Full_Path);
          sw.Write(txt_Area.Text);
          sw.Close();
       }
      else MessageBox.Show("An error occured during saving file content",
                           "Saving Failed",
                           MessageBoxButtons.OK,
                           MessageBoxIcon.Hand,
                           MessageBoxDefaultButton.Button1);
}
```

```
private void CreateFile()
{ File_Full_Path=txt_CreateFile.Text;
  try
   {  if(!File.Exists(File_Full_Path) || chk_Overwrite.Checked)
           File.Create(File_Full_Path);
   }
  catch (IOException ioexp)
   {MessageBox.Show(ioexp.Message); }
}
```

```
private void ReadContent()
 {
  File_Full_Path = txt_CreateFile.Text;
  StreamReader sr = null;
  if (File.Exists(File_Full_Path))
   { sr = new StreamReader(File_Full_Path);
     MessageBox.Show(sr.ReadToEnd(),
                     "Saved Text",
                     MessageBoxButtons.OK,
                     MessageBoxIcon.None,
                     MessageBoxDefaultButton.Button1);
     sr.Close();
   }
  else MessageBox.Show("An error occured during reading target file",
                       "Saving Failed",
                       MessageBoxButtons.OK,
                       MessageBoxIcon.Hand,
                       MessageBoxDefaultButton.Button1);
}
```

```
private void Create_NewDir()
 {
  New_Directory = txt_NewDir.Text;
   if (!Directory.Exists(New_Directory))
    Directory.CreateDirectory(New_Directory);
 }
```

```
private void moveFile()
 {
  try
   {
    File.Move(txt_CreateFile.Text,
              string.Format("{0}\\{1}", txt_NewDir.Text,
                                        txt_CreateFile.Text));
   }
  catch (IOException ioexp)
   { MessageBox.Show(ioexp.Message); }
}
```

الآن و قد تم تعريف الدوال الأساسية للتطبيق، سنقوم بإسنادها للأزرار التي تم إدراجها سابقا في الواجهة الرسومية، و لكن قبل ذلك ينبغي إدراج المجال الخاص بالمدخلات و المخرجات بشكل عام و الملفات و المجلدات بشكل خاص و هو المجال System.IO.

قمت بتعريف متغيرين بشكل عام و هما File_Full_Path و New_Directory. المتغير الأول لخزن المسار كامل للملف المنشأ، المخزن و المقروء، أي مشترك بين ثلاثة دوال. المتغير الثاني يمكن من تحديد المسار الكامل للمجلد الجديد الذي تم إنشاءه، و تم تعريفه بشكل عام لأننا سنحتاج القيمة المسندة له لتحديد وجهة الملف المنقول.

```
private string File_Full_Path = string.Empty;
private string New_Directory = string.Empty;

private void btn_ReadSavedTxt_Click(object sender, EventArgs e)
 { ReadContent();      }

private void btn_CreateFile_Click_1(object sender, EventArgs e)
 { CreateFile();      }

private void btn_createDir_Click(object sender, EventArgs e)
 { Create_NewDir();    }

private void btn_SaveTXT_Click(object sender, EventArgs e)
 { SaveContent();      }

private void btn_move_Click(object sender, EventArgs e)
 { moveFile();         }

private void btn_Exit_Click(object sender, EventArgs e)
 { Application.Exit(); }
```

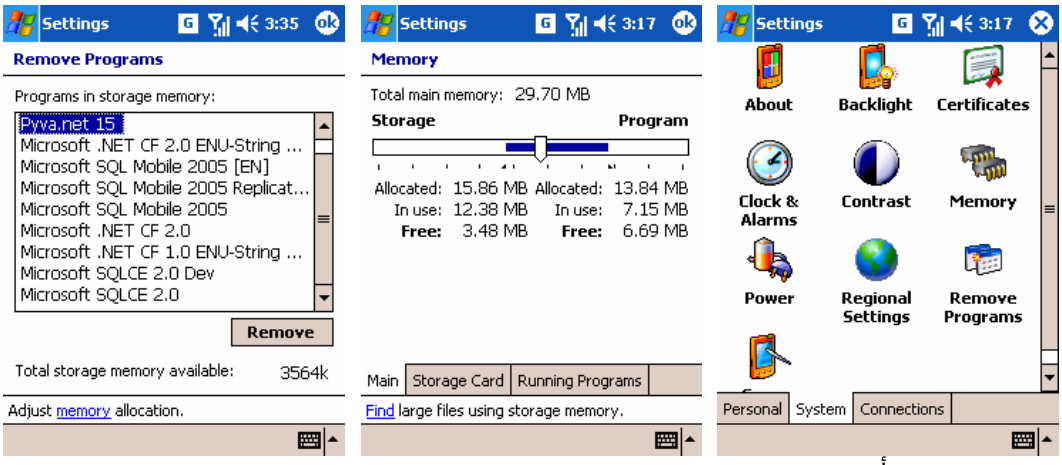
ملاحظة: من الممكن تغير خصائص الملف أو المجلد كجعله مخفي، مضغوط، للقراءة فقط... بالإعتماد على القسم FileInfo مثال:

```
System.IO.FileInfo fileInfo = new
System.IO.FileInfo("FileFullPath_WithExtension");
fileInfo.Attributes = System.IO.FileAttributes.ReadOnly;
System.IO.DirectoryInfo DirInfo = new System.IO.DirectoryInfo("MyDirPath");
DirInfo.Attributes = System.IO.FileAttributes.Hidden;
```

وهكذا ينتهي هذا التطبيق آملا أن أكون قد وفقت في بلوغ الأهداف المرجوة منه.

4. التعامل مع ذاكرة حاسوب الكف

تعتبر محدودية ذاكرة حاسوب الكف من أبرز العراقيل التي تعترض المطور أثناء تطوير البرمجيات الضخمة. لذلك أخصص هذا الجزء لإبراز كيفية التعامل مع الموارد المتاحة.



ج ب أ

الصورة رقم 4 : زيادة ذاكرة حاسوب الكف

نتيجة الإستعمال المكثف لحاسوب الكف خاصة في الإبحار في شبكة الأنترنت تتكاثر الملفات و المجلدات الغير مرغوب فيها أساسا في المجلدات التابعة لمتصفح الأنترنت. هذه بعض المراحل لصيانة الحاسوب و يحبذ القيام بها دوريا:

- إفتح متصفح الملفات و إفسخ الملفات التي لا تحتاجها.
- وبما أنك في ذلك المجلد إفتح المجلد Temp وقم بفسخ كافة محتوياته.
- إفسخ الملفات التي لا تحتاجها من المجلد

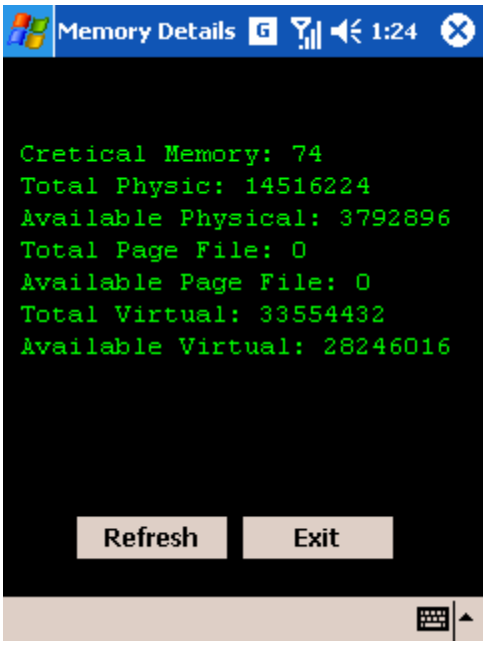
Windows\Profiles\guest\Temporary Internet Files\Content.IE5 ما عدى الملف index الذي هو بطبعه محمي.

يساهم حذف البرمجيات الغير مرغوب فيها من حاسوب الكف بشكل كبير في التخفيض من المساحة المحجوزة. و قد جسدت في الصورة رقم 4 (أ ثم ج) المراحل للقيام بذلك.

طريقة أخرى و ربما الأنجع للحصول على أكبر مساحة فارغة في حاسوب الكف تتمثل القيام بعملية Hard Reset. هذه العملية تعيد الجهاز إلى الحالة الأولى التي يوزع فيها، حيث يتم إلغاء كافة الإعدادت التي غيرها صاحب الجهاز مع حذف كافة البرمجيات المضافة و الملفات الخاصة. لذلك فإن كان المستخدم بحاجة لملفات ما أو إعدادات معينة فعليه الإحتفاظ بنسخة منها مسبقا.في قائمة الأدوات[9] في واجهة ال ActiveSync يوجد الإختيار ...Backup/Restore. هذا الإختيار يوفر واجهتين إحداهما للإحتفاض بنسخة إحطياطية من إعدادات الجهاز و الملفات و الثاني لإعادتها لحاسوب الكف.

5. التطبيق

الغاية من هذا التطبيق إبراز كيفية الإطلاع على خصائص ذاكرة حاسوب الكف بمختلف أنواعها.



الصورة رقم 5 : الواجهة الرئيسية للتطبيق

لتيسير متابعة الشيفرة و إستعماله ألخص في هذا الجدول جملة المكونات، أسمائها و النص المسند لها:

نوع المكون	الإسم	النص
Label	lbl_memoryDetails	
Button	btn_Exit	Exit

ترتكز الشفرة الأساسية للتطبيق على الدالتين الأصليتين GlobalMemoryStatus و GetSystemMemoryDivision المتوفرتان في المكتبة Coredll.dl. لذلك يمكن إدراج المجال System.Runtime.InteropServices عند تعريف الدالتين.

[9] قائمة الأدوات : Tools

```
public struct MEMORYSTATUS
{
  public uint Length;
  public uint MemoryLoad;
  public uint TotalPhys;
  public uint AvailPhys;
  public uint TotalPageFile;
  public uint AvailPageFile;
  public uint TotalVirtual;
  public uint AvailVirtual;
}
[System.Runtime.InteropServices.DllImport("CoreDll.Dll")]
 public static extern int GlobalMemoryStatus(ref MEMORYSTATUS ms);
[System.Runtime.InteropServices.DllImport("CoreDll.Dll")]
 public static extern int GetSystemMemoryDivision(ref uint lpdwStorePages,
                                                   ref uint ldpwRamPages,
                                                   ref uint ldpwPageSize);
```

عرفت الشفرة الرئيسية للتطبيق في دالة مستقلة و هي MemoryDetails. توفر هذه الأخيرة المعلومات اللازمة حول الذاكرة و قد عرفتها بشكل مستقل ليتم إستدعائها مرتين. الأولى عند تحميل الواجهة الرسومية و الثانية عند تحديث المعلومات المتحصل عليها.

```
private string MemoryDetails()
{
 string Details = string.Empty;
 uint storePages = 0;
 uint ramPages = 0;
 uint pageSize = 0;

 int res = GetSystemMemoryDivision(ref storePages,
                                   ref ramPages,
                                   ref pageSize);
 MEMORYSTATUS MM_Status = new MEMORYSTATUS();
 GlobalMemoryStatus(ref M_Status);
 Details = string.Format(
"Cretical Memory: {0}\nTotal Physic: {1}\nAvailable Physical: {2}\nTotal"+
 " Page File: {3}\nAvailable Page File: {4}\nTotal Virtual: {5}\nAvailable Virtual: {6}",
                           M_Status.MemoryLoad.ToString(),
                           M_Status.TotalPhys.ToString(),
                           M_Status.AvailPhys.ToString(),
                           M_Status.TotalPageFile.ToString(),
                           M_Status.AvailPageFile.ToString(),
                           M_Status.TotalVirtual.ToString(),
                           M_Status.AvailVirtual.ToString());
  return Details;
}
```

ينبغي تفعيل الخاصية Multiline لمجال إظهار المعلومات (lbl_memoryDetails) حول الذاكرة في عدة أسطر كما هو الصورة رقم.

يكفي االآن إستدعاء الدالة عند تحميل الواجهة الرسومية Form1_Load و عند النقر على زر التحديث btn_refresh_Click.

```
private void Form1_Load(object sender, EventArgs e)
 { lbl_memoryDetails.Text = MemoryDetails(); }

private void btn_refresh_Click(object sender, EventArgs e)
 { lbl_memoryDetails.Text = MemoryDetails(); }

private void btn_Exit_Click(object sender, EventArgs e)
 {  Application.Exit(); }
```

6. خاتمة

وهكذا ينتهي هذا التطبيق آملا أن أكون قد وفقت في بلوغ الأهداف المرجوة منه.

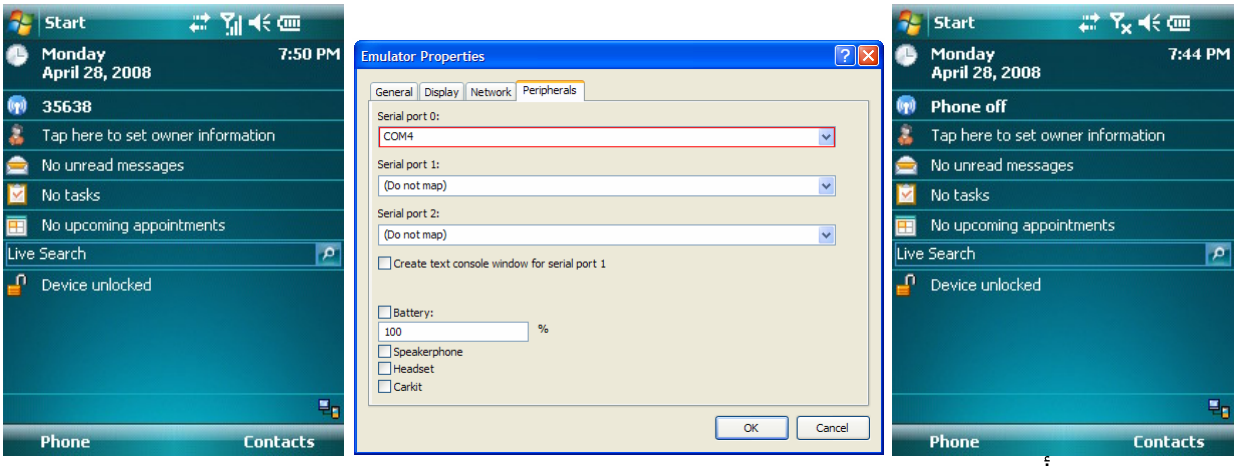
الجزء الرابع: الرسائل القصيرة

1. مقدمة

لا شك أن الرسائل القصيرة و الإتصال هي الخدمات الرئيسية لأي هاتف، و بدون إحداهما يفقد الجهاز قيمته. هاتان الخدمتان متوفرتان في حواسيب الكف و الهواتف الذكية مع توفير عدة صلاحيات للمستخدم.

2. الرسائل القصيرة

و للأسف أن التعامل مع الرسائل القصيرة و الإتصال في حواسيب الكف من جيلي 2003 و 2005 عسير إلى أن قدم جيل 2006 بالأداة Cellular Emulator. توفر هاته الأداة عدة وظائف من أبرزها إرسال رسالة قصيرة و الإتصال بالمحاكي، شرط تغير منفذ الإتصال في قائمة إعدادات المحاكي. الصورة التالية تجسد المراحل اللازمة لذلك.

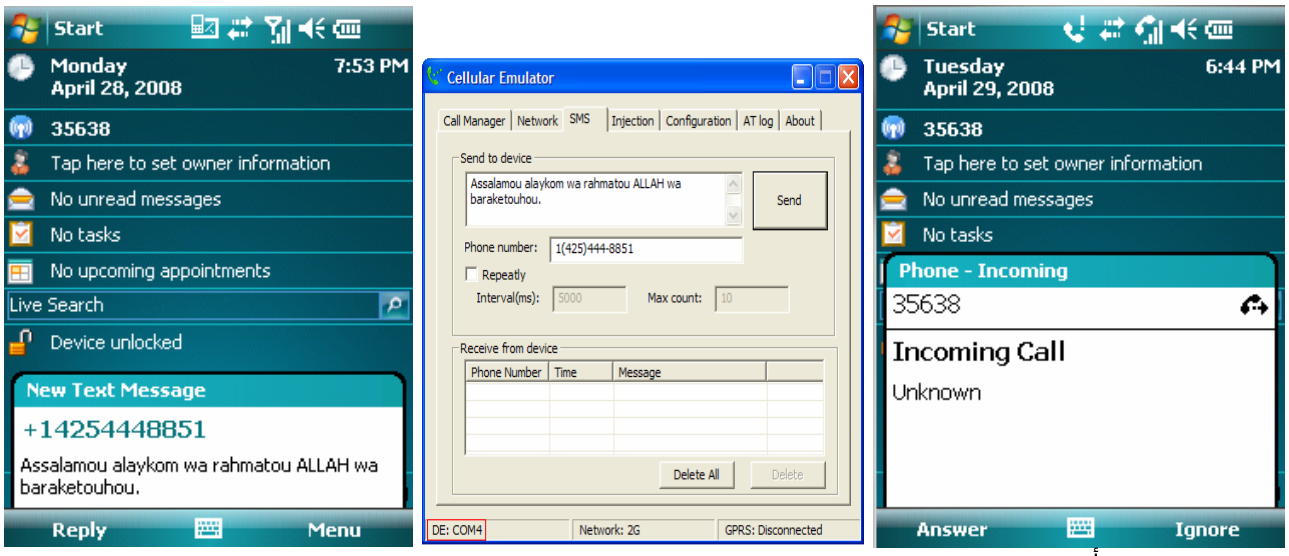


ج ب أ

الصورة رقم 1 : الواجهة الرئيسية للتطبيق

في الصورة أ نلاحظ أنه لا يوجد إتصال بين المحاكي و شبكة الهاتف الجوال. المرحلة الأولى من تغير الإعدادات تتمثل في إسناد القيمة COM4 للخانة "Serial port 0:" ثم القيام ب Soft Reset لمحاكي حاسوب الكف. وعند تشغيل المحاكي يكون هناك إتصال كما تجسده أيقونة شبكة الهاتف الجوال. هذا لا يعني أنه من الممكن الإتصال بهاتف أخر أو إرسال رسالة قصيرة لهاتف ما، إنما لمحاكاة الجهاز الحقيقي و تطوير التطبيقات المتعلقة بشبكة الهاتف الجوال.

لتجربة الإتصال بالمحاكي أو رسال رسالة قصيرة له أو منه، يكفي كتابة نص الرسالة في الجدول SMS و إرسالها و إستخدام الجدول Call Manager للإتصال بالمحاكي أو منه. الصورة الموالية تجسد عملية الإتصال بالمحاكي و إرسال رسالة قصيرة له.



ج ب أ

الصورة رقم 2 : الواجهة الرئيسية للتطبيق

أذكر أنه لايمكن تجربة المثال السابق إلا إثر تنصيب عدة البرمجة الخاصة بحواسيب الكف من جيل 2006 حيث تتواجد الأداة Cellular Emulator .

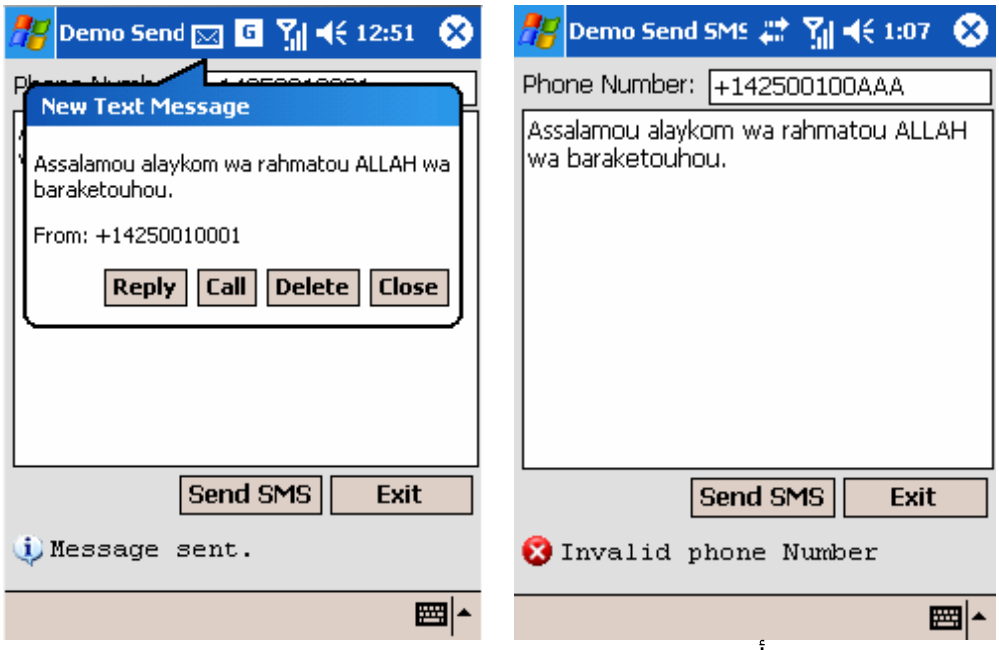
3. التطبيق

هذا التطبيق يخص هواتف الكف من جيل 2003 و لذلك سنعتمد على دوال أصلية[10] تتمثل في:

1. SmsOpen: تستعمل هذه الدالة لفتح العنصر المكلف بخدمة الرسائل القصيرة للإرسال و/أو للإستقبال. إذا لم يوفر مزود الخدمة، إمكانية إرسال و إستقبال الرسائل القصيرة ترجع هذه الدالة رسالة خطأ. مزيد من التفاصيل حول هذه الدالة متواجدة في الرابط [2].
2. SmsSendMessage: تمكن هذه الدالة من إنشاء و إرسال الرسائل القصيرة. مزيد من التفاصيل حول هذه الدالة متواجدة في الرابط [3].
3. SmsClose: تمكن هذه الدالة من غلق خدمة الرسائل القصيرة المتواجدة. مزيد من التفاصيل حول هذه الدالة متواجدة في الرابط [4].

توجد عدة دوال أخرى تخص خدمة الرسائل القصيرة كالقراءة، و التصفية و غير ذلك. كل هذه الدوال متواجدة في الرابط [1].

[10] APIs



ب أ

الصورة رقم 3 : الواجهة الرئيسية للتطبيق

حصلت على الأيقونتين المستعملتين في التطبيق من الملف Microsoft Visual Studio 8\Common7\VS2005ImageLibrary\VS2005ImageLibrary.zip علما أني أستعمل الإصدار 2005 للفيزوال ستوديو.

لتيسير متابعة الشيفرة و إستعماله ألخص في هذا الجدول جملة المكونات، أسمائها و النص المسند لها:

نوع المكون	الإسم	النص
Label	lbl_PhoneNbr	Lbl_PhoneNbr:
Label	lbl_ExceptionMsg	
TextBox	txt_PhoneNbr	+14250010001
Button	txt_SMSBody	Assalamou alaykom…
Button	btn_SendSMS	Send SMS
Button	btn_Exit	Exit
PictureBox	Pict_SendIcon	

ملاحظة: من المحبذ تسمية المكونات بإسم يوح لنوعها لتيسير متابعة الشفرة من قبل مطورها و غيره.

النص المسند لخانة كتابة رقم الهاتف هو "+14250010001" و هذا الرقم يستعمل لإرسال الرسائل من المحاكي و إليه في الآن ذاته.

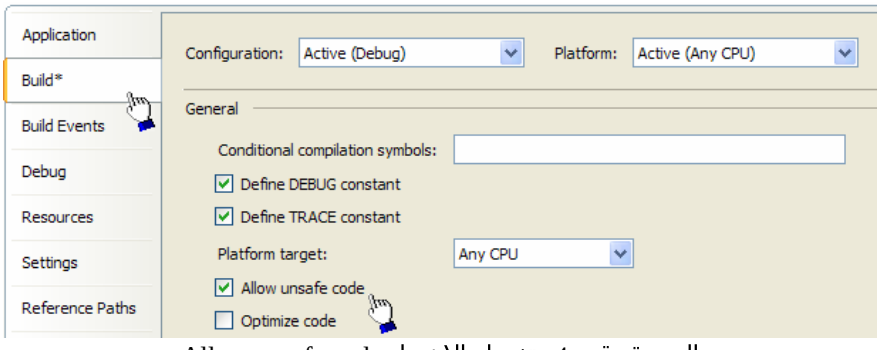
```
[DllImport("sms.dll")]
 private static extern IntPtr SmsClose(IntPtr smshHandle);

[DllImport("sms.dll")]
 private static extern IntPtr SmsOpen(string ptsMessageProtocol,
                                      IntPtr dwMessageModes,
                                      ref IntPtr psmshHandle,
                                      IntPtr phMessageAvailableEvent);
[DllImport("sms.dll")]
 private static extern IntPtr SmsSendMessage(
                                          IntPtr smshHandle,
                                          IntPtr psmsaSMSCAddress,
                                          IntPtr psmsaDestinationAddress,
                                          IntPtr pstValidityPeriod,
                                          byte[] pbData,
                                          IntPtr dwDataSize,
                                          byte[] pbProviderSpecificData,
                                          IntPtr dwProviderSpecificDataSize,
                                          SMS_DATA_ENCODING smsdeDataEncoding,
                                          IntPtr dwOptions,
                                          IntPtr psmsmidMessageID);
```

حيث يتم تعريف النوع SMS_DATA_ENCODING بالكيفية التالية:

```
private enum SMS_DATA_ENCODING
{ SMSDE_OPTIMAL,
  SMSDE_GSM,
  SMSDE_UCS2
}
```

في التطبيق الخاص بذاكرة حاسوب الكف أدرجت المجال System.Runtime.InteropServices ضمنيا مع إستدعاء الدالتين الأصليتين. لا يمكننا تكرار ذلك في هذا التطبيق، لأننا سنحتاج المجال المعني ليس فقط لتعريف الدوال الأصلية إنما لتعريف أقسام أخرى مثل Marshal

تم تعريف الدالة على أنها من نوع نظرا لإستعمال المؤشرات فيها بشكل يمكن أن يحدث مشاكل في التعامل مع الذاكرة و الترجمة. لإدراج النوع في الشفرة لابد من تغير إعدادات المشروع و تفعيل الإختيار Allow unsafe code كما هو مجسد في الصورة رقم 4.



الصورة رقم 4: تفعيل الإختيار Allow unsafe code

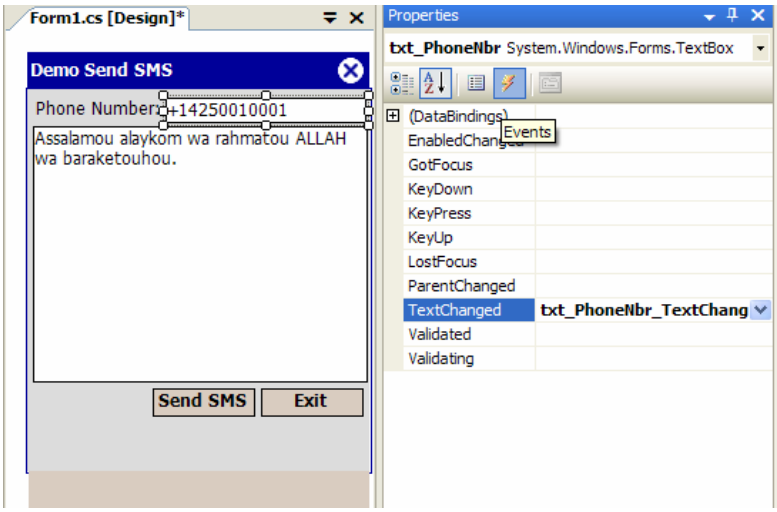
```
public static unsafe void SendMessage(string sPhoneNumber, string sMessage)
{  IntPtr _pzero = IntPtr.Zero;
   try
    { if (SmsOpen(SMS_MSGTYPE_TEXT,
                  (IntPtr)SMS_MODE_SEND,
                  ref _pzero, IntPtr.Zero) != IntPtr.Zero)
                    { SendingState = "Could not open SMS.";
                    // or we can: throw new Exception("Could not open SMS.");
                    }

       byte[] buffer = new byte[0x204];
         try
          {  fixed (byte* byte_DestNrb = buffer)
              {
               byte* byte_numPtr = byte_DestNrb;
               Marshal.WriteInt32((IntPtr)byte_numPtr, 0);
               byte_numPtr += 4;

               foreach (byte num in Encoding.Unicode.GetBytes(sPhoneNumber))
                { Marshal.WriteByte((IntPtr)byte_numPtr, num);
                  byte_numPtr++; }

              byte[] pbProviderSpecificData = new byte[12];
              byte[] bytes = Encoding.Unicode.GetBytes(sMessage);
              int length = bytes.Length;

              if (SmsSendMessage(_pzero,
                                 IntPtr.Zero,
                                 (IntPtr)byte_DestNrb,
                                 IntPtr.Zero,
                                 bytes,
                                 (IntPtr)length,
                                 pbProviderSpecificData,
                                 (IntPtr)pbProviderSpecificData.Length,
                                 SMS_DATA_ENCODING.SMSDE_OPTIMAL,
                                 (IntPtr)SMS_OPTION_DELIVERY_NONE,
                                IntPtr.Zero) != IntPtr.Zero)
                      { SendingState = "Could not open SMS.";
                      // or we can: throw new Exception("SMS send failed.");
                      }
               else SendingState = "Message sent.";
              }
          }
         finally
         { //byte_DestNrb = null;
         }
    }
  finally
    { if (_pzero != IntPtr.Zero)
      { SmsClose(_pzero); }
    }
}
```

توفر بعض لغات البرمجة إمكانية كتابة أرقام فقط في خانة النص و للأسف أن السي شارب لا يوفر ذلك سوى كان التطبيق موجه لحواسيب المكتب أو لحواسيب الكف. لذلك قمت بكتابة دالة تقوم بتفقد محتوى نص معين إن كان يحتوي أرقام فقط أو أكثر من ذلك لإجبار المستخدم على كتابة أرقام فقط في الخانة المخصصة لتحديد رقم الهاتف.

من الممكن أن يكتب المستخدم رقم الهاتف مسبوقا بعلامة الجمع '+' عندما يريد أن يرسل الرسالة القصيرة إلا بلد آخر، لذلك أضفت عملية إختبار أولية تقوم بحذف هذا الرمز من أول سلسلة الأرقام المكتوبة و مراجعة محتواها مجددا في مرحلة لاحقة.

```
private bool IsNumeric(string InputString)
 { string dico = "0123456789";

     if (InputString.StartsWith("+"))
      InputString = InputString.Replace("+", "");

           foreach (char c in InputString)
               if (dico.IndexOf(c) == -1)
                   return false;
           return true;
}
```

ممكن إستعمال الدالة السابقة قبل إرسال الرسالة القصيرة أو عند الكتابة بالإعتماد على الدالة الحدث TextChanged لمجال الكتابة.



الصورة رقم 5 : الواجهة الرئيسية للتطبيق

لتيسير كتابة نص الرسالة القصيرة أو رقم الهاتف، أدرجت دالة فتح و إغلاق لوحة المفاتيح و هي الدالة SipShowIM:

```
[DllImport("coredll.dll")]
  public extern static void SipShowIM(uint dwFlag);
```

هاته الدالة تأخذ قيمة 1 لفتح لوحة المفاتيح و 0 لغلقها و يمكن إدراجها داخل الدالة الحدث txt_PhoneNbr_GotFocus، أي ما إن ينقر المستخدم في خانة كتابة رقم الهاتف تفتح له لوحة المفاتيح. و يمكن غلقها إذا نقر خرجها بالإعتماد على الدالة الحدث txt_PhoneNbr_LostFocus. يمكن القيام بذلك أيضا لخانة كتابة نص الرسالة. الصورة رقم 5 تجسد كيفية القيام بذلك.

ليس إرسال الرسائل القصيرة الوضيفة الوحيدة التي يمكن القيام بها في هذا المجال إنما من الممكن أيضا قراءة الرسائل الواردة و التصنة على الرسائل الواردة مع إمكانية حذف بعضا منها مباشرة

و آليا وفق مواصفات محددة. لقراءة الرسائل ممكن الإعتماد على الدالة SmsReadMessage و للأسف أن هذه الدالة تعتمد عدة مدخلات بعضها غير واضح و لا توجد معلومات وافية في صفحات ال MSDN حول هاته الدالة.

عدة البرمجة 2005 و 2006 الخاصة بحواسيب الكف جائت بالكثير من التيسير و أبرزها المجال Microsoft.WindowsMobile.Status الذي ييسر الإطلاع على حال الجهاز كمستوى البطارية، الشبكات المتصل بها و غير ذلك. المجال الثاني الهام هو Microsoft.WindowsMobile.PocketOutlook الذي يمكن من تطوير البرمجيات الخاصة بالرسائل القصيرة و البريد الإكترونية.

سنستغل هذا الأخير لإبراز كيفية التنصة على الرسائل القصيرة الوردة على حاسوب الكف أو الهاتف الذكي. ينبغي أن يكون جيل التطبيق من 2005 و ما فوق و سنحتاج لإدراج المجال Microsoft.WindowsMobile.PocketOutlook.MessageInterception

```
using Microsoft.WindowsMobile.PocketOutlook.MessageInterception;
//…
MessageInterceptor SMS;

private void Form1_Load(object sender, EventArgs e)
 {
  SMS = new MessageInterceptor();
  MessageCondition MyCondtion = new MessageCondition(MessageProperty.Body,
MessageComparisonType.StartsWith, "Work" );
  InterceptionAction InterceptAction= InterceptionAction.NotifyAndDelete;
  MessageReceived += new MessageReceivedEventHandler( sms_MessageReceived );
 }

void sms_MessageReceived(object sender, MessageReceivedEventArgs e)
 {
  SmsMessage message = (SmsMessage)(e.Message);
 }
```

الصنف MessageCondition سيقوم بالبحث في نص الرسالة على نص محدد ما إن وجد في أوله، داخله أو آخره. نفس هذا البحث ممكن أيضا في رقم هاتف المرسل للرسالة القصيرة.

4. خاتمة

وهكذا ينتهي هذا التطبيق آملا أن أكون قد وفقت في بلوغ الأهداف المرجوة منه.

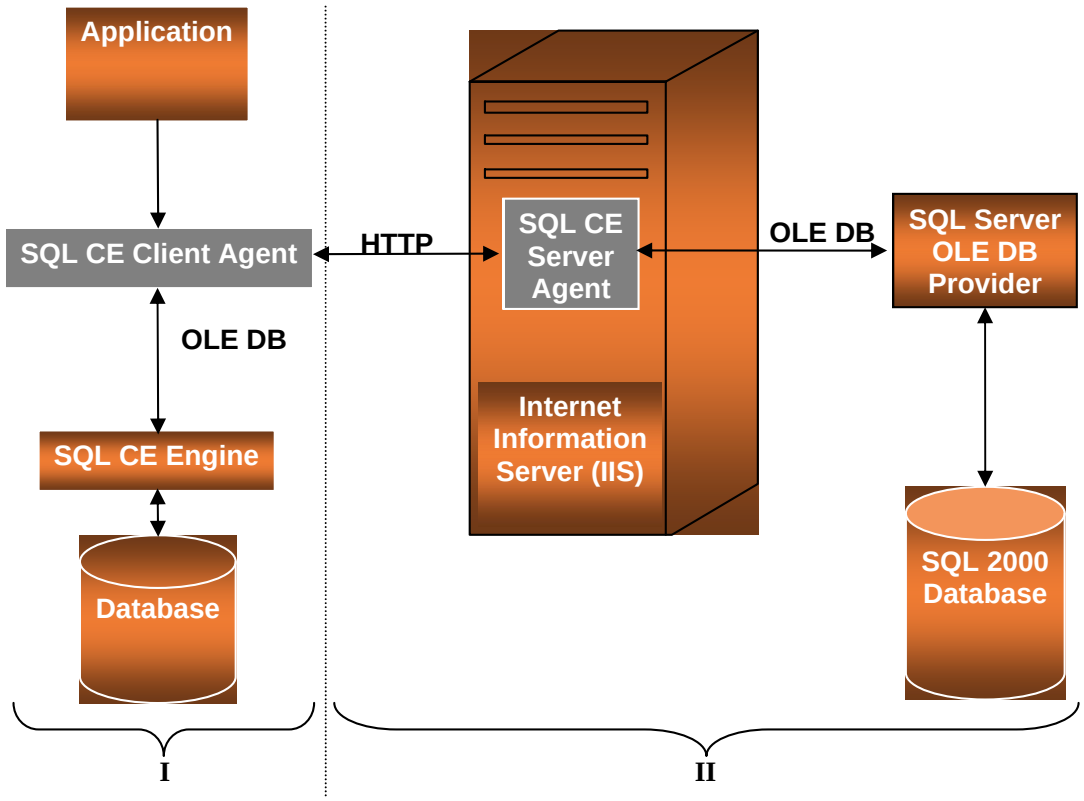
الجزء الخامس: حواسيب الكف و قواعد البيانات

1. مقدمة

الحديث عن البرمجة في منصة دوت دون الحديث عن قواعد البيانات SQL يشوبه خلل ما. لذلك سنتحدث في هذا الجزء عن تطوير هذا النوع من قواعد البيانات و إدراجها في تطبيق سي شارب مع التعريج على بعض الوضائف التي تفرض نفسها كصيانة القاعدة و ضغطها.

2. حواسيب الكف و قواعد البيانات

نحتاج لإستخدام قواعد البيانات SQL أساسا في تطوير المواقع و المشاريع من نوع Webservices و بنسبة أقل في التطبيقات المتعلقة بما يسمى GUI[11].



في الصورة رقم 1 تجسيد لإستخدام قاعدة البيانات في كلتا الحالتين؛ الحال الأولى حيث يحتاج المطور لإستخدام قاعدة البيانات مباشرة دون إدراج الواب و يكون هنا ال SQL CE Engine بمثابة الوسيط بين البرنامج و القاعدة. هذا يخص كل العمليات التي يمكن القيام بها كقراءة و تحديث في ما هو مخزن دون الحاجة للوسيط SQL CE Client Agent الذي يفرض نفسه كوسيط لمستخدم الواب.

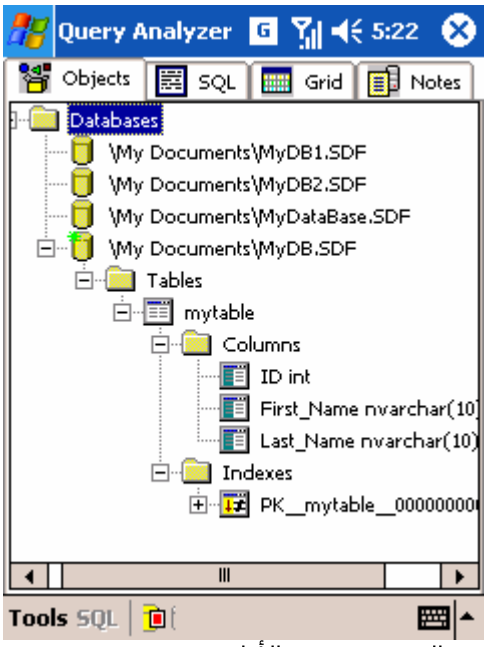
[11] GUI: Graphic User Interface.

المنهج HTTP هو إحدى طرق التواصل بين العميل و المستخدم. هذا المنهج يعتمد أساسا على الصنفين HttpWebRequest و HttpWebResponse و يفترض ذلك التشفير UrlEncoding الذي للأسف غير متوفر في عدة البرمجة الخاصة بحواسيب الكف و الهواتف الذكية. في ما يلي جدول يحتوي جملة من الرموز و الشفرة الخاصة مع ضرورة إضافة الرمز % لكل شفرة. لذلك كتبت الدالة التي تقوم بعملية التشفير و لكن كان ذلك لإستعمالي الخاص؛ أي أن معمارية البيانات التي كنت مطالب لإرسالها (XML) بين المستخدم و العميل تحتوي فقط هذه الرموز ولك تعديلها و فق الحاجة.

```
public string urlencoder(String str)
{
  str = str.Replace("%", "%25");
  str = str.Replace("/", "%2F");
  str = str.Replace(":", "%3a");
  str = str.Replace("#", "%23");
  str = str.Replace("&", "%24");
  str = str.Replace("=", "%3d");
  str = str.Replace("+", "%2b");
  str = str.Replace("$", "%26");
  str = str.Replace(",", "%2c");
  str = str.Replace(" ", "+"); //or  %20
  str = str.Replace("<", "%3c");
  str = str.Replace(">", "%3e");
  str = str.Replace("~", "%7e");
  str = str.Replace("\"", "%22");
  str = str.Replace("?", "%3f");
  str = str.Replace(";", "%3b");
  str = str.Replace("\t", "%09");


  return str;
}
```

الرمز	الشفرة
Tab	09
Space	20
"	22
)	28
(	29
+	B2
،	C2
.	E2
؛	B3
:	A3
>	C3
<	E3
@	40
]	B5
\	C5
[	D5
^	E5
'	60
}	B7
|	C7
{	D7
~	E7

إثر إنشاء قاعدة البيانات و الحاجة لمطالعة جداولها يمكن إستخدام الأداة Query Analyzer المتوفرة في عدة برمجة قواعد البيانات. يمكن تحميل العدة اللازمة من عدة البرمجة و الأدوات اللازمة.

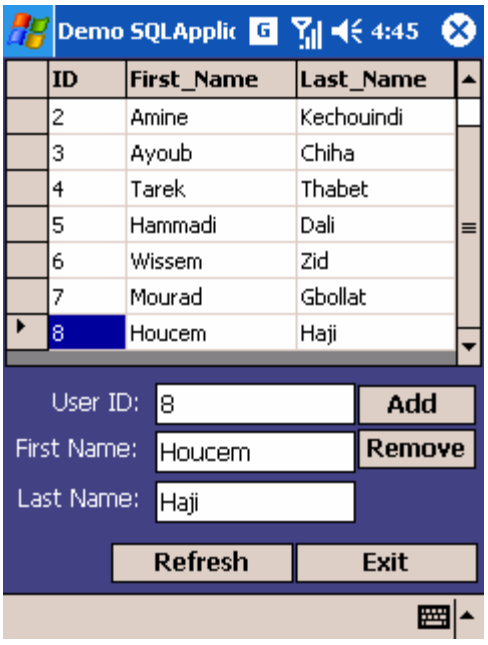


الصورة رقم 2 : الأداة Query Analyzer

3. التطبيق

هذا التطبيق بمثابة مقدمة للتعامل مع قواعد البيانات التي تعد الأكثر إستعمال في منصة الدوت نت ككل و الأكثر طلب في سوق الشغل أيضا.

كأول خطوة في هذا التطبيق، ينبغي إضافة المرجع[12] System.Data.SqlServerCe للمشروع ثم إضافته لقائمة المجالات التابعة للمشروع لكي نتمكن من إستعمال أقسام ال SQL في التطبيق.



الصورة رقم 2 : الواجهة الرئيسية للتطبيق

[12] إضافة مرجع :Add Reference

يحوصل الجدول التالي المكونات المضافة للواجهة الرسومية، أسمائها و نصوصها.

نوع المكون	الإسم	النص
DataGrid	dgr_Container	
Button	btn_Add	Add
Button	btn_Remove	Remove
Button	btn_Refresh	Refresh
Button	btn_Exit	Exit
TextBox	txb_UserID	0
TextBox	txb_UserFName	
TextBox	txb_UserLName	
Label	lbl_ID	User ID:
Label	lbl_FName	First Name:
Label	lbl_LName	Last Name:

كما هو الحال في التطبيقات السابقة تمت كتابة الشفرة رئيسية للتطبيق في دوال مستقلة لتيسير فهمها و إستعمالها. الجدول التالي يحوصل مجموعة الدوال و الوضائف المسندة لها.

إسم الدالة	الوضيفة
CreateDatabase	إنشاء قاعدة البيانات
LaodDatabase	تحميل قاعدة البيانات
AddNewUser	إضافة عنصر جديد
IsNumeric	التحقق من قيمة المعرف
moveFile	نقل الملف

```
public void CreateDatabase()
 {
  if (!File.Exists(@"\My Documents\MyDB.sdf"))
   {
    SQLengine = new SqlCeEngine(connStr);
    SqlCeConnection sqConn = new SqlCeConnection(connStr);
    SQLengine.CreateDatabase();
    sqConn.Open();

    SqlCeCommand SQLCmd = new SqlCeCommand(
     "Create Table mytable (ID int primary key, First_Name nvarchar(10),
Last_Name nvarchar(10))", sqConn);

    SQLCmd.ExecuteNonQuery();
  SQLCmd.CommandText = "insert into mytable values (1, 'Mourad', 'Baghdadi')";
                       SQLCmd.ExecuteNonQuery();
  SQLCmd.CommandText = "insert into mytable values (2, 'Amine', 'Kechwindi')";
                        SQLCmd.ExecuteNonQuery();
  SQLCmd.CommandText = "insert into mytable values (3, 'Ayoub', 'Chiha')";
                       SQLCmd.ExecuteNonQuery();
  SQLCmd.CommandText = "insert into mytable values (4, 'Tarek', 'Thabet')";
                       SQLCmd.ExecuteNonQuery();
  SQLCmd.CommandText = "insert into mytable values (5, 'Hammadi', 'Dali')";
                       SQLCmd.ExecuteNonQuery();
  SQLCmd.CommandText = "insert into mytable values (6, 'Wissem', 'Zid')";
                       SQLCmd.ExecuteNonQuery();
  SQLCmd.CommandText = "insert into mytable values (7, 'Mourad', 'Nehdi')";
                       SQLCmd.ExecuteNonQuery();
   }
 }
```

```
private bool LaodDatabase()
{
  bool bRet = false;
  try
  {   // Specify the data source
     SqlCeConnection oCon =
                 new SqlCeConnection("Data Source=\\my documents\\MYDB.sdf");
     SqlCeDataAdapter DA =
                 new SqlCeDataAdapter("select * from mytable", oCon);
     DataSet DS = new DataSet("mytable");
     // populate the dataset
     DA.Fill(DS);
     // Write the dataset contents to the xml document
     DS.WriteXml("\\my documents\\mytable.xml");
     dgr_Container.DataSource = DS.Tables[0].DefaultView;
     bRet = true;
 }
 catch (Exception exp) { MessageBox.Show(exp.Message); }
 return bRet;
}
```

```
private bool AddNewUser(int ID, string UserFName, string UserLName)
{
 bool bRet = false;
 SqlCeEngine sqEng = new SqlCeEngine(connStr);
 SqlCeConnection sqConn = new SqlCeConnection(connStr);
 try
  {
   sqConn.Open();
   SqlCeCommand sqCmd = new SqlCeCommand("Create Table mytable (ID int primary
key, First_Name nvarchar(10), Last_Name nvarchar(10))", sqConn);

   sqCmd.CommandText = "insert into mytable values (" +
                                        ID.ToString() +
                                   ", '" + UserFName +
                                " ', '" + UserLName + "')";
   sqCmd.ExecuteNonQuery();
   bRet = true;
  }
 finally { sqConn.Dispose();
           MessageBox.Show("New user was added to your database.",
                           "Adding New User");
         }
 return bRet;
```

```
private bool IsNumeric(string InputString)
{
 string dico = "0123456789";
 foreach (char c in InputString)
  if (dico.IndexOf(c) == -1)
   return false;
 return true;
}
```

أحيانا نحتاج لإنشاء قواعد بيانات كبيرة من ناحية التداخل بين جداولها و كثرة المعطيات المخزنة فيها مما يضيع الكثير من ذاكرة حاسوب الكف بالإضافة إلى بطأ البحث فيها و جعلها عرضة للتعطيل. لذلك

يحبذ ضغطها لإسترجاع المساحة المحجوزة دون فائدة عبر إستعمال الدالة Compact التابعة للقسم SqlCeEngine.

```
SqlCeEngine SQLEngine = new SqlCeEngine(@"\My Documents\MyDB.sdf");
SQLEngine.Compact(null);
// or
SQLEngine.Compact("Data Source=; Password =Arabteam2000;");
```

في التعليمة الأولى يتم ضغط قاعدة البيانات و إنشاء أخرى تعوض الموجودة دون إنشاء قاعدة أخرى لذلك يتم إستعمال المدخل . الدالة Compact لا توفر فقط إمكانية ضغط قاعدة البيانات إنما تشفيرها و إضافة كلمة سر لها و تغير جملة الربط.

إثر كل تحديث، من الأفضل أخذ نسخة إحطياطية لقاعدة البيانات لتجاوز أي مشاكل يمكن أن تصيبها بعطب. أبرز أسباب إصابة قاعدة البيانات بعطب تتمثل في إنقطاع التيار (البطارية) أو الربط أثناء القراءة منها أو تحديثها.

إذا حصل العطب و ليس لديك نسخة إحطياطية من القاعدة فجرب الدالة Repair التابعة بدورها للقسم SqlCeEngine، فربما تساعد في حل الإشكال.

```
SqlCeEngine SQLEngine = new SqlCeEngine(@"\My Documents\MyDB.sdf");
SQLEngine.Repair(null, RepairOption.RecoverCorruptedRows);
// or
SQLEngine.Repair(null, RepairOption.DeleteCorruptedRows);
```

هاته الدالة توفر إختيارين و هما إصلاح الأسطر المسببة للعطب و حذفها. و من البديهي تجربة إصلاحها و إن تواصل العطب ننتقل إلى الحل الثاني.

4. خاتمة

وهكذا ينتهي هذا التطبيق بشرح مبسط و سريع لما يتعلق بإستخدام قواعد البيانات SQL في حواسيب الكف و لا يزال الكثير لتعلمه في هذا المجال.

الجزء السادس: تصميم الواجهات الرسومية

1. مقدمة

من متطلبات تطوير برمجيات حرفية و خاصة التجارية منها لحواسيب الكف أو غيره تصميم واجهات رسومية مشجعة على إستخدامها من ناحية البساطة، الجمالية و النجاعة.

2. تصميم الواجهات الرسومية

تخصيص جزء كامل للخوض في تصميم الواجهات الرسومية جاء نتيجة للفرق الكبير بين ما توفره عدة البرمجة الخاصة بحاسوب المكتب و الخاصة بحواسيب الكف. العديد من المكونات مفقودة في بيئة حواسيب الكف مما يقيد حرية المطور و يدفعه للتعرف على التفاصيل للإستفادة منها بالإضافة لما توفره عدة البرمجة المطورة من فريق ال OpenNETCF.

3. التطبيق

التطبيق هذه المرة مختلف عن سابقيه من حيث الموضوع و لغة البرمجة أيضا. موضوع هذا التطبيق هو تصميم واجهات رسومية لتجاوز إمكانيات عدة البرمجة كتصميم InputBox و نافذة PopUp مما ييسر تصميم واجهات أخرى كالواجهات الإفتتاحية[13]. أما لغة البرمجة فهي ال VB.Net.

ج ب أ

الصورة رقم 1 : الواجهة الرئيسية للتطبيق

[13] واجهة إفتتاحية: Splash Screen

1.3. تصميم واجهة رسومية InputBox

فهذا النوع من التطبيقات لا يحتاج المطور إلى كتابة خوارزمية معقدة أو طويلة إنما يكفي أن يجيد تعديل إعدادات الواجهة التي سيعتمدها ك في مشروعه. لتيسير عملية التصميم سننتقل بمراحل:

1. غير الخاصية FormBorderStyle للواجهة الرسومية من FixedSingle إلى None لكي نتمكن من إظهار الواجهة الصغيرة التي نريدها دون إخفاء الخلفية التي كانت موجودة. كما أن بهذه العملية سيصبح الحجم الفعلي للواجهة متحكم فيه و ما إن نغير طول الواجهة يتغير مكان شريط الأدوات ليحذي أسفلها.
2. عدل الآن حجم الواجهة بالمقايس التي تريدها.
3. يبقى الأن تحديد مكان ظهور الواجهة عبر تعديل الإحداثيتين الأفقية و العمودية في الخاصية Location.
4. إذا قمت بترجمة التطبيق ستلاحظ أن في شريط الأدوات يوجد إسم الواجهة و زر الإغلاق. لحذفهما يمكن حذف نص الإسم في الخاصية Text للواجهة و تغير قيمة الخاصية ControlBox إلى False لحذف زر الإغلاق.
5. يمكننا الآن إضافة PictureBox للواجهة و إسناد صورة لها باللون و الخطوط التي نريدها. ثم تعديل حجمها بنفس مقايس الواجهة. و ينبغي أن تكون إحداثياتها (0,0) لأنها ستكون خلفية للواجهة الرسومية.
6. الآن سنقوم بإضافة الأزرار و مجال النص و نص المساعدة و هي مراحل سبق و أبرزتها في التطبيقات السابقة.

تمت حوصلت نوع، أسماء و نصوص المكونات المضافة للواجهة الرسومية في الجدول التالي:

نوع المكون	الإسم	النص
PictureBox	Pict_Background	
Button	btn_OK	OK
Button	btn_Abort	Abort
TextBox	txb_InputTxt	
Label	lbl_helpText	Please set …

تقوم الخوارزمية الرئيسية بمقارنة كلمة السر المدخلة من قبل المستخدم بنص مخزن مسبقا. و لحجب النص المكتوب و جعله في شكل نجوم, ينبغي إسناد الحرف الذي نريد إظهاره في الخانة PasswordChar عوض الحروف المكونة لكلمة السر.

```
Public Class Form1
  Private MyPassword As String = "Hammadi2007"
  Private Sub btn_abort_Click(ByVal sender As System.Object, _
                              ByVal e As System.EventArgs) _
                              Handles btn_abort.Click
   Application.Exit()
  End Sub

  Private Sub btn_Ok_Click(ByVal sender As System.Object, _
                           ByVal e As System.EventArgs) _
                           Handles btn_Ok.Click

   If Equals(txb_InputTxt.Text, MyPassword) Then
     Me.Hide()
     MainForm.Show()
   Else
    MessageBox.Show("Please set the correct passowrd or close the device.", _
                    "Wrong Passowd", _
                    MessageBoxButtons.OK, _
                    MessageBoxIcon.Hand, _
                    MessageBoxDefaultButton.Button1)
   End If
 End Sub
End Class
```

الصورة رقم 1.أ تجسد الواجهة الرسومية المتحصل عليها علما أن خلفية الواجهة تتمثل في صورة تحتوي الإطارات و اللون الرمادي.

2.3. تصميم واجهة رسومية PopUp

يعتمد تصميم واجهة رسومية PopUp نفس المراحل التي تم إبرازها في التطبيق السابق لكن هذه المرة سنضيف مؤقت لفتح الواجهة الرسومية بتدرج. أي سيتم تغير إحداثيات الواجهة لمدة زمنية معينة.

عند تحميل الواجهة الرسومية تم تحديد المكان المبدئي لها و فق حجم شاشة حاسوب الكف. الإحداثيات الأولية تساوي عرض الشاشة ناقص عرض الواجة مقابل طول الشاشة زائد طول الواجهة. بذلك تكون الواجهة الرسومية في أول التشغيل مخفية ثم ما إن يتم تشغيل المؤقت تبدأ في الظهور موازية الجانب الأيمن لشاشة حاسوب الكف.

```
Public Class Form1
    Private X_Location As Integer = 0
    Private Y_Location As Integer = 0

    Private Sub Form1_Load(ByVal sender As System.Object, _
                           ByVal e As System.EventArgs) Handles MyBase.Load
        X_Location = Screen.PrimaryScreen.Bounds.Width + 1
        Y_Location = Screen.PrimaryScreen.Bounds.Height - 14
        Me.Location = New Point(X_Location - Me.Width,
                                Y_Location + Me.Height)
        tmr1.Enabled = True
    End Sub

    Private Sub tmr1_Tick(ByVal sender As System.Object, _
                          ByVal e As System.EventArgs) Handles tmr1.Tick
        If (Me.Location.Y > Y_Location - Me.Height) Then
            Me.Location = New Point(X_Location - Me.Width, Me.Location.Y - 8)
        Else : tmr1.Enabled = False
        End If

    End Sub

End Class
```

داخل دالة المؤقت يتم التفقد دوريا ما إن كانت الواجهة الرسومية قد برزت بشكل كامل أو يواصل في تغير إحداثياتها.

3.3. إدراج المكون NotifyIcon

هذا المكون مستعمل بكثرة في التطبيقات الموجهة لحواسيب المكتب لكنه غير متوفر في لائحة المكونات الموجهة لحواسيب الكف. فريق ال OpenNETCF قام بالمطلوب و أضاف هذا المكون في عدة البرمجة التي يوفرها و الموجهة لحواسيب الكف. لذلك سنحتاج كمرحلة أولى لإدراج المجال OpenNETCF.Windows.Forms لقائمة المراجع المضافة للمشروع.

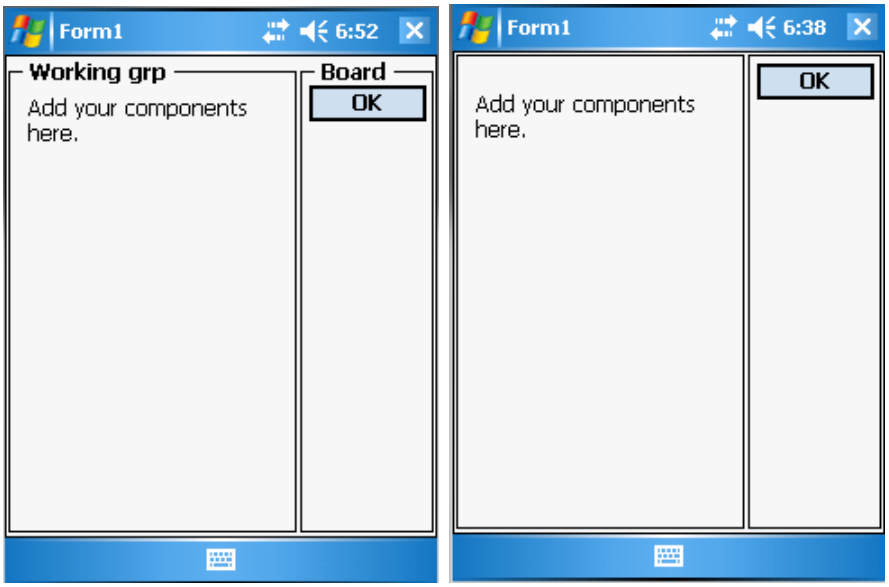
ثانيا سنضيف الأيقونة التي نريد إظهارها إلى موارد المشروع مثلا كما يلي:

Project Properties\Resources\Add Resource\Add Existing File ثم نقوم بالتقديم للمكون الجديد الذي علينا تعريفه بمفردنا بما أنه "دخيل":

```
private OpenNETCF.Windows.Forms.NotifyIcon MynotifyIcon;
public Form1()
{
  InitializeComponent();

  this.MynotifyIcon = new OpenNETCF.Windows.Forms.NotifyIcon();
  this.MynotifyIcon.Icon = Resources.Msg; // Msg is the name of the icon
  this.MynotifyIcon.Visible = true;
  this.MynotifyIcon.Click += new EventHandler(this.MynotifyIcon_Click);
}

private void MynotifyIcon_Click(object sender, EventArgs e)
{ this.Show();}
```

4.3. إدراج المكون GroupBox

أثناء تنظيمي للكود السابق إكتشفت وجود عدة مكونات هامة معرفة في المجال OpenNETCF.Windows.Forms و تساعد في تصميم الواجهات الرسومية بأكثر حرفية. أحد هذه المكونات هو ال GroupBox. في هذا المثال أبرزت كيفية إستغلال هذا المكون عبر إدراج ثلاثة منها. أحدهم يحتضن إثنين و وكل منهما يحتضن بدوره مكونات أخرى.



الصورة رقم 2 : إدراج المكون GroupBox

عادة عند إضافة مكون ما للواجهة الرسومية و تعديل إعداداته يتم إنشاء الشفرة المقابلة له تلقائيا. لكن في هذا الحال ينبغي تعريف كل ما يتعلق بالمكون من تصميم، وضيفة و غير ذلك. الآن بالإمكان العودة للتطبيق الأول في هذا الجزء و تعويض الصورة بمكونات GroupBox و مغيرين لون الخلفية.

```
private OpenNETCF.Windows.Forms.GroupBox MainGroupBox;
private OpenNETCF.Windows.Forms.GroupBox WorkingGroupBox;
private OpenNETCF.Windows.Forms.GroupBox BoardBox;

private void Form1_Load(object sender, EventArgs e)
{//=== Main GroupBox
  MainGroupBox = new OpenNETCF.Windows.Forms.GroupBox();
  MainGroupBox.Location = new Point(0, 0);
  MainGroupBox.Size = new Size(240, 268);
  MainGroupBox.BackColor = Color.WhiteSmoke;
  this.Controls.Add(MainGroupBox);

  //=== Working GroupBox
  WorkingGroupBox = new OpenNETCF.Windows.Forms.GroupBox();
  WorkingGroupBox.BackColor = Color.WhiteSmoke;
  WorkingGroupBox.Location = new Point(2, 2);
  WorkingGroupBox.Size = new Size(160, 263);
  WorkingGroupBox.Text = "Working grp";
    /*Adding a label to working GroupBox*/
    Label lbl_Help = new Label();
    lbl_Help.Text = "Add your components here.";
    lbl_Help.Size = new Size(140, 100);
    lbl_Help.Location = new Point(10, 20);
    WorkingGroupBox.Controls.Add(lbl_Help);
  MainGroupBox.Controls.Add(WorkingGroupBox);

  //=== Board GroupBox
  BoardBox = new OpenNETCF.Windows.Forms.GroupBox();
  BoardBox.Location = new Point(164, 2);
  BoardBox.Size = new Size(74, 263);
  BoardBox.BackColor = Color.WhiteSmoke;
  BoardBox.Text = "Board";
    /* Adding a Button to BoardGroupBox*/
    Button btn_OK = new Button();
    btn_OK.Text = "OK";
    btn_OK.Location = new Point(5, 15);
    btn_OK.Size = new System.Drawing.Size(65, 20);
    BoardBox.Controls.Add(btn_OK);
  MainGroupBox.Controls.Add(BoardBox);
}
```

4. خاتمة

لا يمكن إنكار أهمية الصور و الأيقونات في تصميم الواجهات الرسومية بطريقة حرفية كبيرة مع ضرورة الإنتباه إلى ذاكرة حاسوب الكف و حجم البرنامج المطور. المجال لا يزال يخبأ الكثير من المكونات و الصلاحيات و لم أذكر منها هنا إلا اليسير.

الجزء السابع : خاتمة عامة

إثر هذا التجوال السريع في كيفية تطوير تطبيقات متنوعة و خاصة بحواسيب الكف يصبح للقارئ أبجديات هذه التقنية علما أنها لا تزال تخفي العديد من الثراء و الإمكانيات خاصة إذا كانت لغة البرمجة المعتمدة السي أو السي++.

آمل أن يوفقني الله سبحانه و تعالى في إصدار نسخة ثانية من هذا الكتاب تكون أكثر ثراء و تشمل مجالات أخرى كالتعامل مع الشبكات اللاسلكية، برمجة الألعاب، ما يتعلق بالإتصال و غير ذلك.

ع

أعلم أنه هناك مشاكل أكبر من الكتابة في البرمجة في حواسيب الكف و الهواتف الذكية و لكن ... جاد الفقير بما عنده

المراجع

[1] http://msdn.microsoft.com/en-us/library/aa454259.aspx

[2] http://msdn.microsoft.com/en-us/library/aa455065.aspx

[3] http://msdn.microsoft.com/en-us/library/aa455068.aspx

[4] http://msdn.microsoft.com/en-us/library/aa455057.aspx

[5] http://msdn.microsoft.com/en-us/library/ms894600.aspx

لكل الإستفسارات، الإقتراحات و التصويب المتعلق بمحتويات هذا الكتاب بالإمكان الإتصال بي عبر البريد الإلكتروني التالي:

Daly.Hammadi@laposte.net