

عنوان	توسعه نرم افزار برای سیستم عامل ویندوز موبایل، بخش سوم: توسعه ی نرم افزارهای ویندوز فرم برای دستگاه های مختلف ویندوز موبایل
عنوان اصلی	<b>Windows Mobile App Development Part 3: Basic WinForm App Development for Windows Mobile Devices</b>
کلمات کلیدی	C#, Win Mobile, WinForm, .NET, Dev
مؤلف	<b>mstruys, dougturn</b>
مرجع	<a href="http://www.codeproject.com">http://www.codeproject.com</a>
سطح	مبتدی
مترجم	مهدی عبداللهی ( <a href="http://m0911.wordpress.com">http://m0911.wordpress.com</a> )
تاریخ انتشار	۲ فروردین ۱۳۸۹
تعداد صفحه	۱۲
مطالب مرتبط	توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش اول: ایجاد نخستین برنامه توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش دوم: شبیه ساز دستگاه و مدیریت شبیه ساز توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش چهارم: افزودن کنترل سفارشی و استفاده از GPS توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش پنجم: مقدمه ای بر SQL Server CE توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش ششم: امنیت دستگاه و نصب نرم افزار توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش هفتم: توسعه برای وب موبایل
فایل های ضمیمه	

توسعه ی نرم افزار ویندوز موبایل شباهت زیادی به توسعه ی نرم افزار در دسکتاپ دارد به ویژه زمانی که یکی از دو زبان ویژوال بیسیک یا ویژوال سی شارپ دات نت را استفاده می کنید. شما همان ابزارهای توسعه ی برنامه های ویندوز دسکتاپ را برای ویندوز موبایل هم استفاده می کنید لیکن تفاوت هایی نیز بین این دو محیط هست. دستگاه های ویندوز موبایل صفحه ی نمایش کوچک تر و منابع محدود تر دارند و همچنین قابل حمل بوده، اغلب باتری استفاده می کنند.

این مقاله اطلاعاتی را در باره ی چگونگی نرم افزارهای ویندوز فرم برای دستگاه های ویندوز موبایل به شما خواهد داد و در نهایت شما یاد می گیرید که چطور نرم افزارها را برای استفاده از امکانات دستگاه های ویندوز موبایل توسعه دهید و در عین حال حواس تان به شارژ باتری هم باشد و صفحه نمایش برنامه تان بدون توجه به جهت صفحه ی نمایش دستگاه، درست عمل کند.

### دستگاه های ویندوز موبایل مدل های متنوعی دارند

دستگاه های ویندوز موبایل در بازار بسیار متنوع هستند و به علاوه نگارش های مختلفی از ویندوز موبایل را استفاده می کنند. در این مقاله ما با نگارش ۶ ویندوز موبایل (هر دو ویرایش استاندارد و حرفه ای) کار خواهیم کرد.

### دستگاه های ویندوز موبایل استاندارد

این دستگاه ها اغلب به نام اسمارت فون (تلفن هوشمند) در بازار هستند. همه ی این دستگاه ها امکانات تلفن را دارند لیکن صفحه ی لمسی (Touch Screen) ندارند. معمولا برای کار با آن ها به یک دست نیاز دارید (one-hand operation). برای توسعه نرم افزار هایی که با این دستگاه ها کار کنند شما ابزار توسعه ی ویندوز موبایل ۶ استاندارد (Windows Mobile 6 Standard SDK) را همراه با ویژوال استودیو ۲۰۰۵ یا بالاتر نیاز دارید.

### دستگاه های ویندوز موبایل ۶ حرفه ای

این دستگاه ها عمدتا به نام پاکت پی سی (Pocket PC=PPC) شناخته می شوند. این دستگاه ها بعضا امکانات تلفن را نیز دارند ولی همه ی آن ها به صفحه ی لمسی مجهز هستند. معمولا برای کار با این دستگاه ها به دو دست همراه با یک کنترل لمسی نیاز دارید. (برای توسعه نرم افزار هایی که با این دستگاه ها کار کنند شما ابزار توسعه ی ویندوز موبایل ۶ حرفه ای (Windows Mobile 6 Professional SDK) را همراه با ویژوال استودیو ۲۰۰۵ یا بالاتر نیاز دارید.

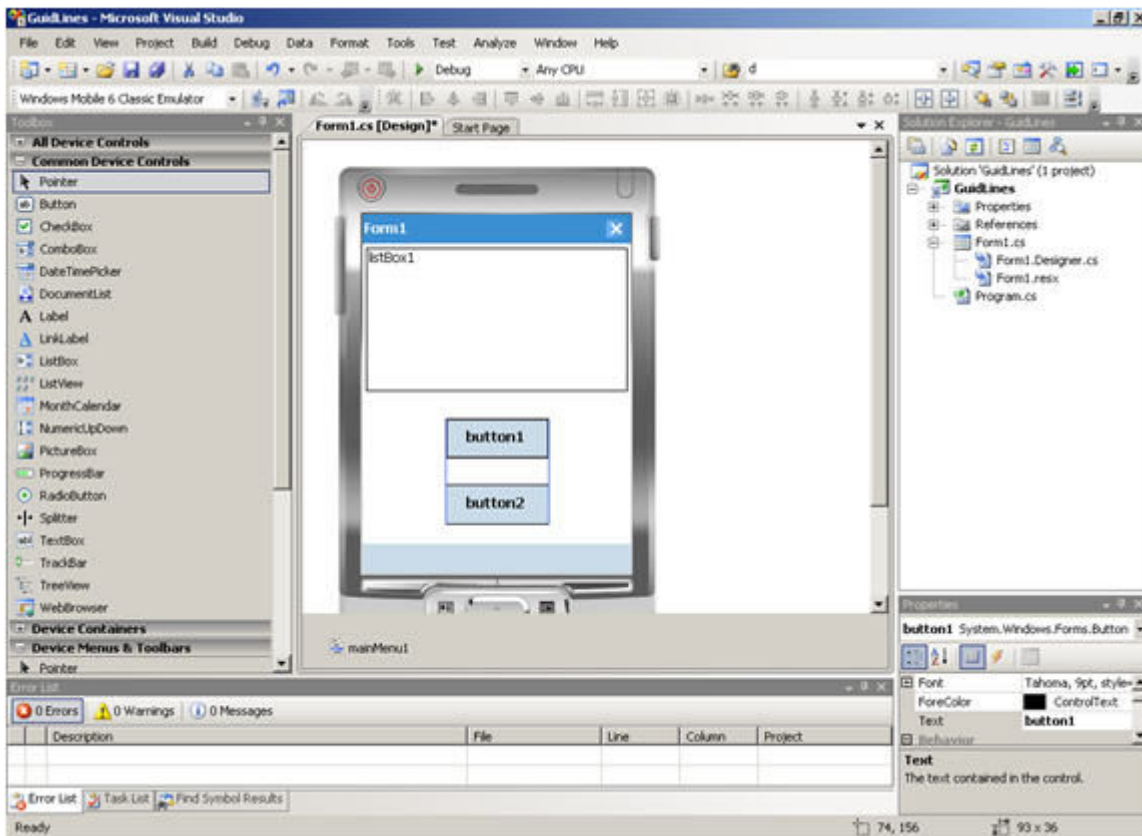
### مهارت هایی که توسعه دهنده باید داشته باشد

روش های توسعه ی نرم افزار های ویندوز موبایل و دسکتاپ در محیط ویژوال استودیو ۲۰۰۸ تفاوت چندانی با هم ندارند. پس از ورود به محیط ویژوال استودیو ۲۰۰۸ شما تصمیم می گیرید که چه نوع پروژه ای را و در چه پلاتفرمی آغاز کنید. سپس ویژوال استودیو یک پروژه ی اولیه با یک فرم خالی نمایش می دهد که در آن فرم، واسط کاربر را طراحی کنید. عمده ترین تفاوت به هنگام توسعه ی نرم افزار ویندوز موبایل، صفحه نمایش کوچک و در عین حال متنوع مورد استفاده در دستگاه ها است. تفاوت بعدی را زمانی متوجه می شوید که به بخواهید کنترل های واسط کاربر را از جعبه ابزار ویژوال استودیو ۲۰۰۸ بکشید و روی فرم قرار دهید. تعداد محدودی کنترل در مقایسه با پلاتفرم دسکتاپ در دسترس شما خواهد بود. این محدودیت در ویرایش استاندارد بیشتر از ویرایش حرفه ای ویندوز موبایل است. از آن جا که دستگاه های ویندوز موبایل استاندارد صفحه ی لمسی ندارند، برخی کنترل های ویرایش حرفه ای ویندوز موبایل در ویرایش استاندارد کارآیی ندارند.

ایجاد واسط کاربری برای برنامه یعنی همان کشیدن کنترل ها از جعبه ابزار ویژوال استودیو و رها کردن و قرار دادن آن روی یک فرم یا داخل یک کنترل کانتینر (Container) - که برای گروه بندی کنترل ها روی فرم قرار داده اید- و از طرفی می توانید چند فرم در برنامه تان داشته باشید، خودتان کنترلی درست کنید که از یک کنترل موجود مشتق شده باشد یا این که از ابتدا یک کنترل را کاملا خودتان ایجاد کنید. علاوه بر این کنترل های ساده تعدادی پنجره ی مکالمه هم هست که شما را کمک خواهد کرد تا برنامه تان به برنامه های عمومی و استاندارد ویندوز موبایل شبیه باشد.

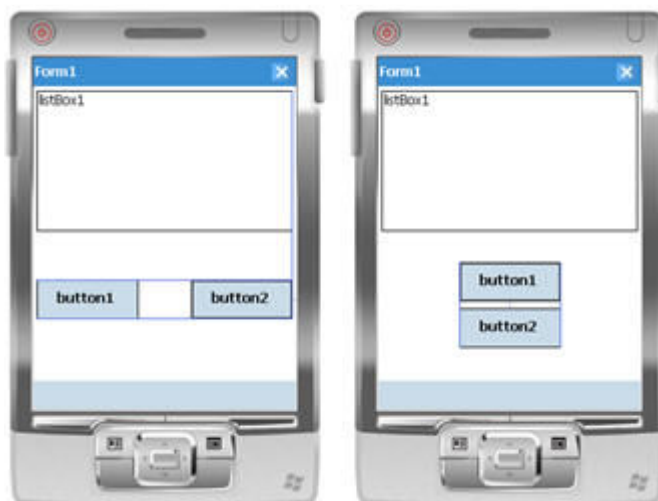
### طراحی یک واسط کاربر ساده در ویندوز موبایل ۶ حرفه ای

یک پروژه ی جدید در محیط ویژوال استودیو ۲۰۰۸ ایجاد کنید. نوع پروژه باید Smart Device project در یکی از زبان های C# یا ویژوال بیسیک دات نت باشد. پلاتفرم مقصد را Windows Mobile 6 Professional SDK انتخاب نمایید. ویژوال استودیو ۲۰۰۸ یک فرم خالی با ابعاد متناسب با پلاتفرم مقصد به شما نمایش خواهد داد که به شکل یک دستگاه ویندوز موبایل است تا به حد امکان شبیه تر باشد به آن چه در واقعیت استفاده می گردد. با استفاده از جعبه ابزار Device Controls (شکل ۱) می توانید کنترل های مورد نظر تان را روی فرم قرار دهید.



شکل ۱: ویژوال استودیو ۲۰۰۸ در نمای طراحی فرم

خب! بیکار نمانید و کنترل های مختلفی را که می توانید در طراحی واسط کاربر استفاده کنید، نگاهی بیندازید. اگر برخی از آن ها به چشم تان آشنا نیستند بهترین کار این است که آن ها روی فرم قرار دهید و مشخصه های آن ها و کارکرد هر کدام را مطالعه کنید و صد البته راهنمای آنلاین را هم به یاد داشته باشید. یکی از مهم ترین چیزهایی که به هنگام طراحی واسط کاربر به چشم خواهد آمد، نحوه ی کار طراح واسط کاربر است. دقت کنید که چطور کمک تان خواهد کرد محل قرار گیری کنترل ها را تراز کنید و کمترین فاصله ی ممکن را برای آن ها در نظر بگیرید. خطوط راهنما (Guide Lines) کمک زیادی خواهند کرد تا در اندک زمانی، واسط کاربر را خیلی خوش دست طراحی کنید. در شکل ۲ قرار گیری دکمه های button1 و button2 را در دو حالت افقی و عمودی نسبت به هم می بینید که به کمک خطوط باریک آبی رنگ، لبه های آن ها نسبت به هم تراز شده اند.



شکل ۲: خطوط راهنما بهترین دستیار شما در ترکیب بندی واسط کاربر

#### ایجاد نرم افزاری که بتواند با جهت های مختلف صفحه نمایش سازگار باشد

تنوع زیاد دستگاه ها و مدل های آن ها باعث شده است که توسعه ی یک نرم افزار که بتواند در تعداد متنوعی دستگاه کار کند اندکی پیچیده باشد. ویژگی استودیو به شما امکان بندکشی و ثابت کردن کنترل های واسط کاربر را فراهم می کند. با ابزار توسعه ی ویندوز موبایل ۶ می توانید اطلاعاتی در باره ی قابلیت های دستگاه و نگارش های سیستم عامل دستگاه به دست آورید. لیکن در این مقاله ما روی چند مورد خاص و محدود کار می کنیم:

- توجه به میزان منابع آزاد سیستم و باتری
- اجرا در دستگاه های ویندوز موبایل ۶

- نمایش درست در هر دو حالت افقی، عمودی و همچنین دستگاه هایی با صفحه نمایش مربع شکل

با وجودی که (در زمان نگارش این مقاله) دستگاه های ویندوز موبایل استاندارد جهت های مختلف صفحه ی نمایش را پشتیبانی نمی کنند ولی ممکن است در آینده این امکان فراهم شود. در برخی موارد نمایش اطلاعات در صفحه ی افقی باعث راحتی بیشتر کاربر خواهد شد. ویژگی استودیو به هنگام طراحی واسط کاربر یک برنامه ی ویندوز فرم، کمک تان خواهد کرد تا کنترل ها در یک موقعیت خاص ثابت کرده، یا بندکشی کنید.

#### بندکشی یا ثابت کردن کنترل ها

برای مطمئن شدن از این که یک فرم در هر دو حالت افقی و عمودی به درستی نمایش داده خواهد شد، می توانید کنترل ها را در محل خاصی داخل یک کنترل مافوق (Parent) مانند فرم یا پانل (Panel Control) ثابت یا بندکشی نمایید.

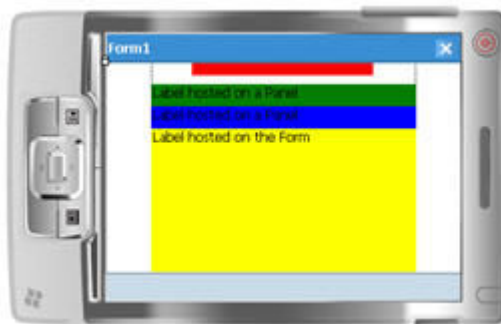
نکته ی مهم تفاوت بین بندکشی (Dock) و تثبیت (Anchor) کنترل ها است. زمانی که شما یک کنترل را نسبت به یک یا چند لبه ی یک کانتینر ثابت می کنید موجب خواهد شد که مختصات نسب به لبه های ثابت شده با تغییر جهت نمایش ثابت بماند. هنگام بندکشی یک کنترل شما لبه ای از کانتینر را که کنترل باید در آن موقعیت قرار بگیرد، تعیین می کنید. اگر کنترل را به هر چهار لبه بندکشی کنید در این صورت آن کنترل تمام سطح کنترل مافوقش (فرم یا کانتینر) را خواهد پوشاند.

برای فهمیدن بهتر تفاوت بندکشی و تثبیت کنترل ها و کارکرد آن ها در جهت های مختلف صفحه ی نمایش یک فرم ساده ایجاد می کنیم و تعدادی کنترل برچسب را (Label) - که هر کدام رنگ پس زمینه ی متفاوت با بقیه داشته باشد - روی آن قرار می دهیم. البته این واسط کاربری شاید به نظر بی مصرف باشد لیکن مهم ترین استفاده ی آن در حال حاضر همین فهمیدن بندکشی و تثبیت کنترل ها و تغییر وضعیت آن ها به هنگام تغییر جهت بین افقی و عمودی است. (شکل ۳)



شکل ۳: ویندوز موبایل ۶ حرفه ای در حالت عمودی، کنترل ها نه ثابت شده اند و نه بندکشی

سه برچسب بالایی روی یک پانل قرار گرفته اند و دو برچسب پایینی روی خود فرم. برای تمامی کنترل هایی که روی فرم قرار می دهید باید مشخصه های تثبیت آن را حذف کنید چرا که ویژوال استودیو به صورت پیش فرض همه ی کنترل ها را نسبت به لبه های بالا و سمت چپ کانتینر شان ثابت می کند. واسط کاربر در حالت عمودی خوب به نظر می آید ولی به محض تغییر جهت از عمودی به افقی مشکلاتی پیش می آید. کافی است به شکل ۴ توجه کنید.



شکل ۴: ویندوز موبایل ۶ حرفه ای در حالت افقی، کنترل ها نه ثابت شده اند و نه بندکشی

بلافاصله خواهید دید که تمامی کنترل های برچسب که در نمایش عمودی قابل دیدن بودند در نمایش افقی دیده نمی شوند. بخشی از پانلی که شامل برچسب بالایی است، از صفحه بیرون رفته است. به همین حالت برچسب پایینی کاملاً محو شده است و سمت راست و چپ صفحه ی نمایش در حالت عمودی خالی مانده اند. این یعنی حالت افقی صرفاً بخشی از صفحه نمایش حالت عمودی واسط کاربر را نشان می دهد.

برای رفع این مشکل بدون نیاز به کدنویسی کافی است از بندکشی و تثبیت کنرل ها به درستی استفاده نمایید. با استفاده از مشخصه ی Anchor می توانید یک کنترل را نسبت به یک یا چند لبه ی صفحه ی نمایش، ثابت نگه دارید. یعنی مختصات آن نسبت به لبه های تعیین شده در هر حالتی از جهت نمایش ثابت می ماند. با استفاده از مشخصه ی Dock نیز تعیین می کنید که یک کنترل چگونه نسبت به لبه های کنترل مافوق خودش تراز شده، یا بخشی از صفحه ی نمایش را پر کند. با بندکشی چند کنترل به یک لبه کنترل مافوق شان، باعث خواهد شد

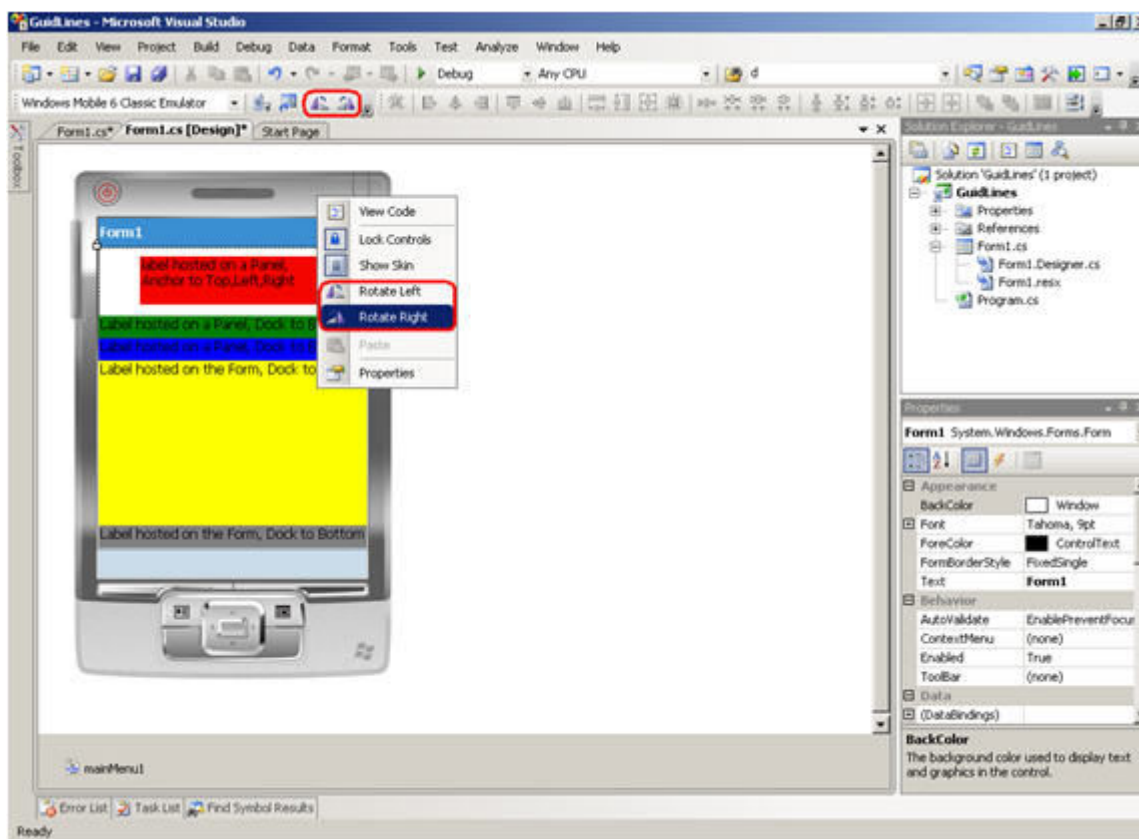
کنترل های مورد نظر به صورت پشت سر هم قرار گیرند. در این مثال سه برچسب بالایی داخل یک پانل قرار گرفته اند. بالاترین برچسب بندکشی نشده است و نسبت به لبه های چپ، راست و بالای پانل ثابت شده است. به عبارتی می توان گفت که به سمت بالای فرم بندکشی شده است. در هر دو حالت عمودی و افقی فاصله ی برچسب با لبه های بالا، چپ و راست تغییر نخواهد کرد. به طور مشخص در حالت نمایش افقی اندازه برچسب بزرگ تر می شود تا فاصله های مذکور تغییر نکنند. دو برچسب دیگر داخل پانل – که روی هم نمایش داده می شوند- به سمت پایین پانل بندکشی شده اند. شماره ی ترتیب (tab order) مشخص می کند که کدام یک از برچسب ها نمایش داده شوند. اگر جهت نمایش را به افقی تغییر دهید، همه ی کنترل های روی فرم همچنان قابل مشاهده خواهند بود و همه ی سطح فرم توسط کنترل ها پر خواهد شد. (شکل ۵)



شکل ۵: ویندوز موبایل ۶ حرفه ای در حالت افقی، با کنترل های ثابت و بندکشی شده

واسط کاربری که در شکل ۵ می بینید، به مراتب بهتر از شکل ۴ است، چون کنترل ها برای نمایش افقی تغییر اندازه داده شده اند. برای این که دست تان بیاید چطور با استفاده از تثبیت یا بندکشی، کنترل ها را جا نمایی کنید کافی است کمی بیشتر وقت گذاشته، حالت های مختلف را آزمایش کنید.

در محیط طراحی ویژوال استودیو ۲۰۰۸ به راحتی هر دو حالت عمودی و افقی را می‌توانید آزمایش کنید. روی شکل دستگاه راست کلیک و گزینه ی Rotate Left یا Rotate Right را انتخاب نمایید. (شکل ۶) راه دیگر آن است که فرم را انتخاب (کلیک) کرده، دکمه های روی نوار ابزار را برای تغییر جهت نمایش استفاده کنید.



شکل ۶: تغییر جهت نمایش در حالت طراحی

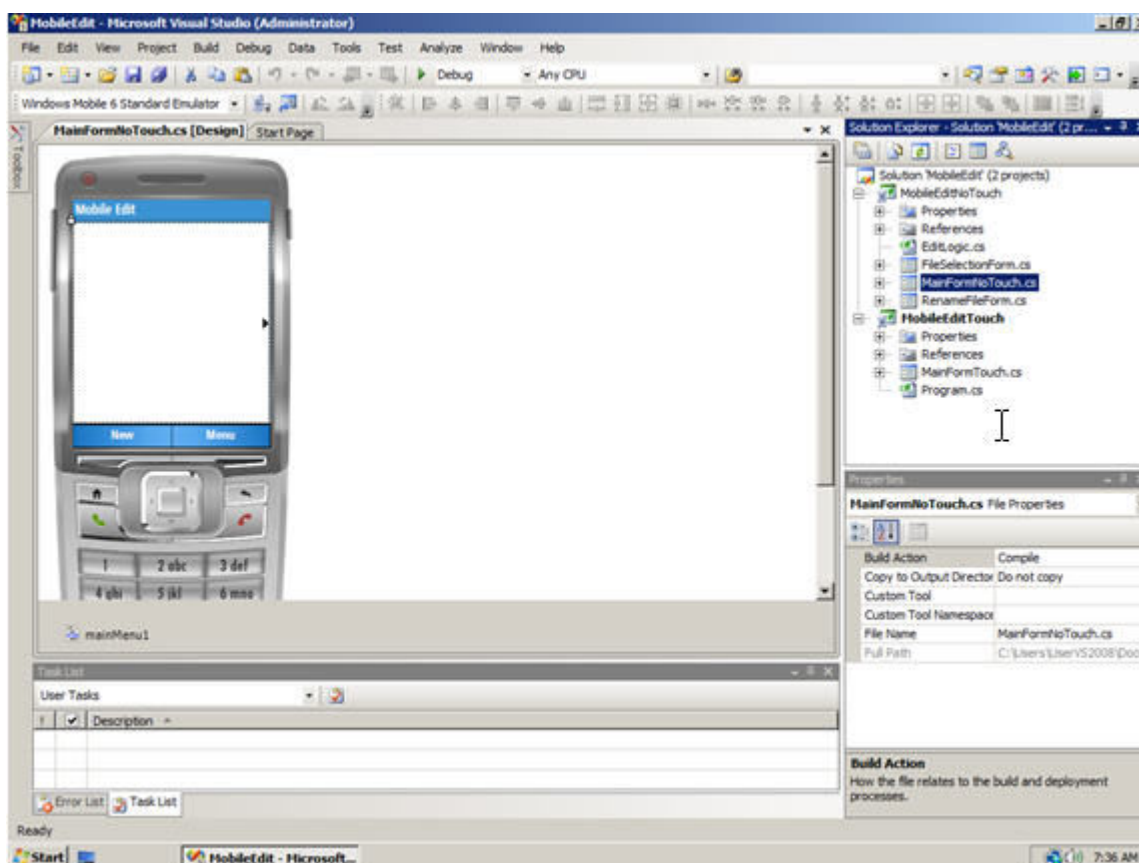
### طراحی برنامه ای که روی دستگاه های مختلف کار کند

به هنگام ایجاد نرم افزاری که روی یکی از پلتفرم های مقصد ابزار توسعه ی ویندوز موبایل ۶ (اعم از استاندارد یا حرفه ای) کار کند، می‌توانید پلتفرم فعلی را به پلتفرم مقصد در ابزار توسعه ی دیگری تغییر دهید. در این حالت نرم افزار به پلتفرم دیگری تبدیل می‌شود یعنی ویژوال استودیو ۲۰۰۸ برنامه ی شما را تغییر می‌دهد. اگر بخواهید برنامه ای را که در ویندوز موبایل ۶ حرفه ای ساخته اید، به پلتفرم دیگری تبدیل کنید مشکلاتی را پیش روی خود خواهید داشت. از آن جا که ویرایش استاندارد ویندوز موبایل ۶ به نسبت ویرایش حرفه ای آن کنترل های کمتری را در واسط کاربر پشتیبانی می‌کند، کنترل های پشتیبانی نشده را باید به کنترل هایی تغییر دهید که در پلتفرم جدید کامپایل و اجرا شوند. ویژوال استودیو ۲۰۰۸ کار تبدیل را به راحتی انجام خواهد داد. از دیدگاه توسعه ی نرم افزار ایده ی برتر آن است که برای ساده تر شدن بحث سازگاری پلتفرم ها، لایه ی بیزینس لاجیک\* را از لایه ی واسط کاربر جدا کنیم. کنترل های واسط کاربر که در پلتفرم جدید پشتیبانی نمی‌شوند، توسط ویژوال استودیو ۲۰۰۸ علامت گذاری می‌شوند و در عین حال ویژوال استودیو ۲۰۰۸ به شما پیشنهاد می‌دهد که کنترل های مذکور را به نزدیک ترین حالت منطبق با پلتفرم جدید تبدیل کنید. اگر نخواستید که پروژه تان را به پلتفرم مقصد دیگری تبدیل کنید می‌توانید از امکانات ارث بری فرم ها (Visual Form Inheritance) استفاده کنید تا با یک دفعه کدنویسی بتوانید برنامه تان را در پلتفرم های مختلف اجرا نمایید.

\* Business Logic این اصطلاح را ترجمه نکردم. اگر اهل فن هستید تصدیق می‌کنید که نباید این چنین عبارتی را ترجمه کرد. اگر هم این اصطلاح برای تان آشنا نیست، «به گیرنده های خود دست نزنید!» کافی است مطالبی در باره ی معماری لایه به لایه ی نرم افزار را مطالعه کنید. (مترجم)



ویژوال استودیو ۲۰۰۸ امکانات کامل طراحی را در زمان ارث بری یک فرم از فرم دیگر، به شما می دهد. برای فهمیدن بهتر، با یک فرم پایه شروع می کنیم که نقطه ی آغازی خواهد بود برای تولید نرم افزاری که در پلاتفرم های مختلف ویندوز موبایل کار کند. با وجود این که ممکن است این کار در نگاه اول غیر ضروری به چشم بیاید، لیکن تجربه ی ایجاد یک فایل کتابخانه ای کلاس (class library) - که فرم پایه را ایجاد می کند- بسیار ارزش مند خواهد بود. این موضوع زمانی که شما یاد بگیرید کدام کنترل ها در فرم های ارث بری شده قابل تغییر هستند و کدام شان نیستند، اهمیت بیشتری دارد. برای این که بتوانید کنترل ها یا کارکرد آن ها را تغییر دهید باید مشخصه ی Access Modifier کنترل مورد نظر را به مقداری غیر از Private تغییر دهید. در شکل ۷ یک فرم پایه می بینید که به عنوان مرحله ی آغاز برنامه تان استفاده می شود.



شکل ۷: فرم پایه ی برنامه برای دستگاه ویندوز موبایل استاندارد

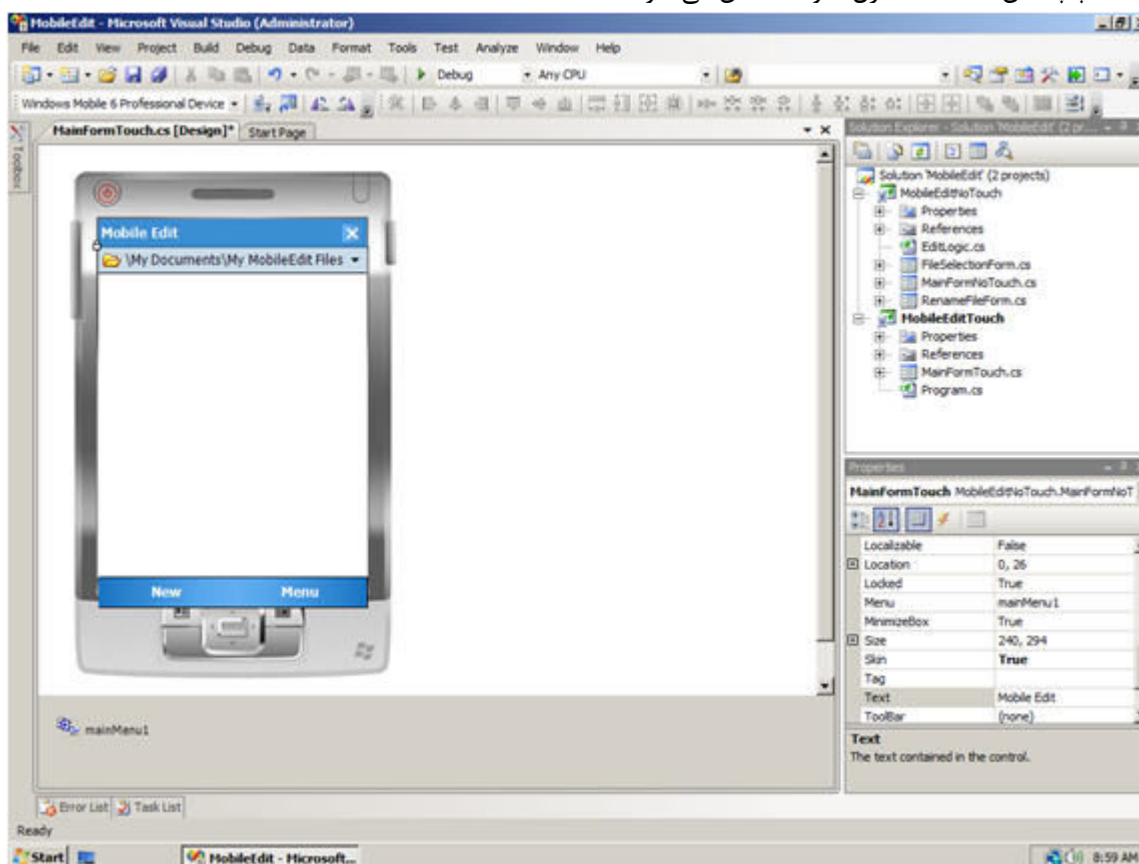
فرم پایه حاوی یک کنترل ویرایشگر (Edit) - که همه ی صفحه را پر کرده- و یک منو است. این برنامه یک ویرایشگر متن ساده است. این برنامه فایل متنی ایجاد می کند، محتوای فایل متنی موجود را تغییر می دهد و همچنین امکان تغییر نام یا حذف فایل متنی را فراهم می کند. با این که فایل های ایجاد شده توسط این برنامه همان فایل های متنی عادی هستند، لیکن برای مدیریت راحت آن ها و فیلتر کردن شان، پسوند فایل را مخصوص این برنامه قرار داده ایم. فرم پایه برای ویندوز موبایل استاندارد طراحی شده است. دلیلش هم روشن است. ویندوز موبایل استاندارد نسبت به حرفه ای کنترل های کمتری دارد. برای مدیریت فایل ها هم دو فرم دیگر در نظر گرفته ایم که یکی فهرست فایل های موجود را نشان دهد و دیگری برای تغییر نام یا حذف آن ها استفاده شود. از آن جا که ویندوز موبایل استاندارد، صفحه ی لمسی را پشتیبانی نمی کند تمامی عملیات فایل ها از طریق گزینه های منو انجام می شوند. در این مقاله ما توضیحات مربوط به ویرایشگر متنی را نخواهیم گفت و روی ایجاد نرم افزاری که روی انواع مختلف دستگاه ها کار کند متمرکز می شویم. هر چند که برنامه ی نمونه ی Mobile Edit برای دانلود دم دست مان هست.

در شکل ۸ شما فرم اصلی را در حال اجرا روی ویندوز موبایل ۶ حرفه ای می بینید. با وجود تفاوت ظاهری اش با آن چه در شکل ۷ دیده اید، ولی این فرم از همان فرم اصلی که در شکل ۷ دیدید مشتق شده است. فقط یک کنترل مخصوص ویندوز موبایل ۶ حرفه ای به آن اضافه شده است.



<http://m0911.wordpress.com>

یعنی یک Document List و همان طور که از نامش پیدا است برای نمایش فهرست فایل ها به کار می رود. با این که فرم اصلی حاوی یک کنترل ویرایشگر متن است ولی به هنگام طراحی آن را نمی بینید چون کنترل «فهرست فایل ها» (Document List) روی آن قرار گرفته است. داخل برنامه با کدنویسی کاری می کنیم که کنترل «فهرست فایل ها» در زمانی که لازم نیست نشان داده نشود. کنترل هایی که بخشی از فرم پایه هستند در فرم مشتق شده هم به ارث می رسند. کنترل هایی را که از طریق ارث بری در فرم جدید قرار گرفته اند، با یک علامت فلش در گوشه ی بالا سمت چپ شان (همانند کنترل منو) مشخص می شوند.



شکل ۸: فرم پایه ی مشتق شده برای ویندوز موبایل حرفه ای

از آن جا که دستگاه های ویندوز موبایل حرفه ای از کنترل های بیشتری پشتیبانی می کنند، خود نرم افزار ساده تر می شود و به راحتی از امکان صفحه ی لمسی استفاده خواهد کرد. امکان استفاده از کنترل «فهرست فایل ها» برای مدیریت فایل ها یعنی این که دیگر نیازی به طراحی فرم جدید نیست. از آن جا که هدف اصلی ما ایجاد یک فایل باینری (اجرای) برای استفاده در هر دو پلتفرم ویندوز موبایل استاندارد و حرفه ای است، به هنگام شروع اجرای آن می توانیم دستگاهی را که برنامه در آن اجرا می شود، تشخیص دهیم و فرم مناسب آن را نمایش دهیم.

NET Compact Framework 3.5. به راحتی امکان تشخیص نوع دستگاه را برای تان فراهم می کند. قطعه کد زیر را - که بخشی از فایل Program.cs است - ببینید:

```
static void Main()
{
    if (SystemSettings.Platform == WinCEPlatform.PocketPC)
    {
        Application.Run(new MainFormTouch());
    }
    else if (SystemSettings.Platform == WinCEPlatform.Smartphone)
    {
        Application.Run(new MainFormNoTouch());
    }
}
```

<http://m0911.wordpress.com>

بسته به نوع دستگاه، یکی از دو فرم `MainFormNoTouch` یا `MainFormTouch` را به عنوان فرم اصلی برنامه مان استفاده می کنیم. مزیت اصلی ارث بری فرم این است که ما امکانات فرم اصلی (`Main`) را در ویندوز موبایل حرفه ای، دوباره استفاده می کنیم. مثلاً اگر کاربر گزینه `New` را در منو کلیک کند، صرف نظر از نوع دستگاهی که برنامه در آن اجرا می شود، مدیر رویدادی (`event handler`) که برای گزینه `New` ی منو در صفحه `MainFormNoTouch` تعریف کرده ایم، اجرا خواهد شد:

```
protected virtual void menuNew_Click(object sender, EventArgs e)
{
    autoSave = false;
    if (tbWorkArea.Text.Length != 0)
    {
        if (MessageBox.Show("Save contents before creating a new file",
            "Save current contents",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Question,
            MessageBoxDefaultButton.Button1) == DialogResult.Yes)
        {
            EditLogic.WriteFile(fileName, tbWorkArea.Text);
        }
        tbWorkArea.Text = string.Empty;
    }
    autoSave = true;
    fileName = EditLogic.FileName;
}
```

این کد از کاربر خواهد پرسید که آیا فایل های در حال استفاده را قبل از ایجاد فایل جدید، ذخیره کند یا نه. همان طور که می بینید امکانات خود ویرایشگر در کلاس دیگری به نام `EditLogic` قرار دارد که لایه ی واسط کاربر را به طور کامل از لایه ی اپلیکیشن لاجیک (`Application Logic`) جدا می کند. اگر برنامه در دستگاه با سیستم عامل ویندوز موبایل حرفه ای اجرا شود، یکی از قابلیت هایی که باید در ابتدای اجرای برنامه غیر فعال شود کنترل «فهرست فایل ها» است. این کار را با ابطال مدیر رویداد `menuNew_Click` مطابق قطعه کد زیر انجام می دهیم:

```
protected override void menuNew_Click(object sender, EventArgs e)
{
    base.menuNew_Click(sender, e);
    documentList.Visible = false;
    MinimizeBox = false;
}
```

زمانی که کاربر در ویندوز موبایل حرفه ای گزینه `New` را انتخاب می نماید، ابتدا کد نوشته شده در فرم پایه اجرا می شود سپس دو خط کد که فقط در ویندوز حرفه ای در دسترس هستند اجرا می شوند. این دستور ها به ترتیب برای مخفی کردن کنترل «فهرست فایل ها» برای دیده شدن کنترل «جعبه متن» و نمایش یک دکمه `OK` در نوار عنوان فرم، اجرا می شوند. برای ابطال مدیر رویداد `menuNew_Click` باید آن را به طور دستی به فایل `MainFormTouch.cs` اضافه نماییم. اگر روی گزینه `New` در محیط طراحی فرم دابل کلیک کنید، ویژوال استودیو یک مدیر رویداد جدید به نام `menuNew_Click_1` ایجاد می کند که در زمان اجرا، پس از فراخوانی مدیر رویداد `menuNew_Click` اصلی در فرم پایه، فراخوانی خواهد شد.

### توجه به منابع سیستم

صرف نظر از مدل دستگاه هر زمان که فرم اصلی برنامه مخفی شود (در پس زمینه قرار گیرد) یک رویداد فراخوانی می شود. به طور مشابه زمانی که فرم اصلی برنامه مجدد فعال می شود و در پیش زمینه ویندوز قرار می گیرد، رویداد دیگری فراخوانی می شود. این رویداد ها را می توانیم برای آزاد سازی منابع سیستم هنگامی که برنامه در پس زمینه اجرا می شود، استفاده نماییم. (مثلا غیر فعال کردن تایمر) در این حال برنامه حافظه ی مورد استفاده اش را زمانی که نیازی به آن نیست، آزاد می نماید و حتی الامکان از کمترین پردازش و به تبع آن کمترین توان باتری استفاده می کند.

اگر برنامه ی Mobile Editor در پس زمینه قرار بگیرد به طور موقت محتوای جعبه متن را در یک فایل ذخیره می کند (کد زیر):

```
protected virtual void MainFormNoTouch_Deactivate(object sender, EventArgs e)
{
    if (fileName != string.Empty && tbWorkArea.Text.Length != 0)
    {
        EditLogic.WriteFile(fileName, tbWorkArea.Text);
        tbWorkArea.Text = string.Empty;
        autoSave = true;
    }
}
```

و هنگامی که برنامه دوباره در پیش زمینه قرار بگیرد، محتوای فایل مذکور را بازیابی کرده، در جعبه متن tbWorkArea نمایش می دهد (کد زیر):

```
protected virtual void MainFormNoTouch_Activated(object sender, EventArgs e)
{
    if (autoSave)
    {
        tbWorkArea.Text = EditLogic.ReadFile(fileName);
    }
}
```

تولید نرم افزار هایی که در هر دو ویرایش حرفه ای و استاندارد ویندوز موبایل کار کند، امکان پذیر است، اما این که فکری هم برای معماری نرم افزار بکنید بسیار مهم و اساسی است. به ویژه زمانی که ملزم باشید امکانات سخت افزاری مختلف را در دستگاه های متنوع در نظر بگیرید، این کار پیچیده تر خواهد بود که صد البته از حوصله ی این مقاله خارج است.



شکل ۹: برنامه ی **Mobile Edit** در حال اجرا روی دستگاه های مختلف

برای استفاده از ازت بری فرم ها در محیط طراحی **Windows Forms** باید یک ایده ی درست و حسابی داشته باشید. البته هر کنترلی را می توانید روی فرم پایه قرار دهید به شرطی که بخواهید آن را در فرم های مشتق شده نیز استفاده نمایید. فقط یادتان باشد اگر می خواهید کارکرد یک کنترل را در فرم مشتق شده تغییر دهید، باید مشخصه ی **Modifiers** کنترل را با **protected** یا **public** مقدار دهی کنید. اگر مشخصه ی **Modifiers** یک کنترل را با **private** مقدار بدهید، در فرم مشتق شده فقط می توانید آن را مخفی یا بندکشی کرده، یا آن را تغییر اندازه بدهید.