



Porting Quickoffice Premier to UIQ 3

Quickoffice Premier®, by Quickoffice, Inc. (Quickoffice) is the leading mobile office suite on the UIQ software platform. The award winning application allows Mobile Business Professionals to manage their office files directly on their smartphone. Quickoffice focuses exclusively on the mobile professional with its core mobile office suite as well as logical application extensions and services.

Quickoffice Premier allows users to view, edit, save, and forward Microsoft Office documents allowing users to be more productive, work smarter and save time. It is now possible for knowledge workers on the go to access, modify or create office documents such as contracts, agreements, business plans, sales presentation and financial reports – any Microsoft Word, Excel or PowerPoint document – right from their UIQ 3 smartphone.

Quickoffice distributes software applications on its own website and other leading aggregator sites such as Handango. Quickoffice was first integrated on the P910 from Sony Ericsson. The success of the P910 and its powerful framework created opportunities for Quickoffice to continue developing on this rich platform and to partner with UIQ Technologies for subsequent smartphone opportunities. Subsequently, Quickoffice became a UIQ 3 Selected Partner in late 2004.

Sony Ericsson, a UIQ licensee, which recently announced 2 new UIQ models, the M600 and P990 - elected to continue licensing and integrating the Enterprise-grade Quickoffice Mobile Office Suite to enable consistent smartphone market growth.

Porting to UIQ 3

UIQ 3 is a more mature platform to be deployed within mobile handsets, in that the software and service enabled by the OS can be developed easily by an increasingly skilled software development community. Symbian OS 9.1, upon which UIQ 3 is based, introduces an enhanced security feature that provides access to key Security API's. Network operators have driven this security requirement for the benefit of their customers and it is now available in the new OS.

The new UIQ 3 software platform includes extensive personalization possibilities, a rich user experience, as well as easy to use features. By having Quickoffice operate on this compelling new OS provides the user with a rich technology experience.

UIQ 3 brings the mobile developer community the valued mid-tier range of devices that will hopefully increase both the volume and reach of their software applications. The range of UI screen configurations and touch pad capabilities will also increase the likeliness of attracting new users to the software platform. UIQ 3 also provides developers the compelling benefit of building your application once and having it run seamlessly on the various UI and screen configurations with UIQ 3.

The Quickoffice Development Process

Taking Quickoffice from a binary break with Symbian 7 and UIQ 2.1 to Symbian 9.1 and UIQ 3 presented the team with a few specific challenges. By far the biggest issue we faced was to develop our software on a moving, still-being-developed OS platform. We are proud to say that Quickoffice Premier was one of the first applications developed on the new UIQ 3 software platform.

1. Platform Security

One of the biggest challenges for the developer community will likely be data caging. Either you receive the security capabilities you need, or you don't get the capabilities you need. We recommend you are mindful of the calling methods that can now fail with `KErrAccessDenied`, but most calling methods should have proper error handling structures. For example, if you find that a certain API your application relies on is no longer available, you will need to redesign your application to use different APIs.

Many applications will access Symbian files directly. Therefore, data caging can be more problematic for developers. Developers and their applications will no longer be able to do this in the Symbian 9.1 environment. Also, file handling is slightly different. You now get open file handles passed as arguments, and not filenames.

A good example of the differences in security is that Quickoffice supports email attachments. Quickoffice used to be responsible for opening these files. Now, due to data caging, the responsibility lies within the OS itself to open, as well as manipulate content and applications. This is entirely new to the OS. Although we have read access to the file, we can't do much else with the file. For example, we can't move the file, nor can we overwrite the file. In other words, our application needs to handle "*RFile handles*", rather than just filenames. This is quite a bit different from UIQ 2.1.

2. UIQ Framework

When moving Quickoffice to Symbian 9.1 / UIQ 3, we faced additional changes within the UIQ framework. Quickoffice is a very UI intensive application and therefore leverages much of the UIQ 3 framework. A key benefit of the new framework; it is designed to be more intelligent in design than UIQ 2.1. Which means the developer community will have to do less in the application. Unfortunately, the framework tries to be smart and can come up short on a few items if you don't use it as strictly recommended or don't select the proper configuration. This is demonstrated when the UIQ framework does something (with regard to layout, command handling, focusing of controls, etc) that you do not expect. The tricky part is then to work out why it's doing what it's doing, rather than just trying to modify your code to get it to do what you expect.

Developers may be tempted to do random experimentation in order to achieve desired behavior when the framework isn't cooperating. We recommend that you don't do this, although it might work in that particular case, it likely won't work on another UIQ 3 device or even for other execution paths on the same device. We recommend that engineers and developers work out why it's doing what its doing and then change your code to work with the framework. Although the "don't hack it, but fix it" mantra applies to any code, the consequences of modifying are significantly greater in UIQ 3. Since the UIQ framework does many things for you in UIQ3, it will do them differently on different phones.

A good example is the UIQ layout manager. In UIQ 3, the applications have a layout manager which is responsible for putting the controls on the view. You no longer construct the controls and place them in specific pixel positions. Now, you tell the layout manager what type of layout you want and have it work with you on the specific layout.

We had an issue where the layout manager kept 're-aligning' some of our controls in an unfashionable manner. The result was that you would get blank spaces between the controls. The layout manager seemed to do this if one of our controls was taller in height than 23 pixels, we could have worked around this by adjusting with the minimum and maximum sizes of the controls. This is what we did and it seemed to work well. But then we started to look at landscape modes and fullscreen modes. All of a sudden the layout manager went crazy on us again clearly the 23 pixels were not working for those particular UI configurations. So we could have done:

```
height = (ui == portrait) ? 23 : 24;
```

although this would have done the trick, most developers cringe at the sight of this code. More importantly, it quickly escalates into many more problems.

After some research into the issue, we found the problem was the type of layout manager we were using. The layout manager can be "row based" or *not* "row based". When it is row based (default), it has some complex heuristics to work out the sizes of each row depending on the controls on it. This is where the "*Microsoft paperclip syndrome*" kicked in; the system tries to be smart, but somehow fails. In this case it got the heuristics all wrong, resulting in blank spaces between the rows and controls, because we didn't use the layout that fitted our needs. Setting the layout manager to be 'non-row based' sorted out the problem (we didn't want row based anyway, we just didn't know there was such a thing).

3. Command Processing Framework

The command processing framework is a way to describe the UI representation, and utilizes the well known "command" Design Pattern. Unlike before, where you tell the framework what menu items you have, UIQ now has a concept of 'commands'. It may take some time, but you will get used to using this framework.

The command processing framework is a very powerful feature. The power of UIQ 3 is that it will let the framework place and sort out commands for you. You should not try and change the type of a command (e.g. from TypeScreen to TypeYes) just to get it to appear where you want. Rather you should just accept that the command will be where the framework wants them to be. If you were to run your app on a different UIQ 3 device, you'd not get the same result if you have been adjusting the commands to get them to appear where you want them on one phone. If you have set the type of the command correctly, the framework will place your commands correctly. Although you might want the command on the right softkey, the style guide might state it should be on the left softkey. Trust the framework to place the command where it should go. A 'positive command' should be of type "yes" and not of type "no" just to get it to appear somewhere else in the dialog.

We say the command processing framework is a powerful feature because it allows you to look at your application in a new and more dynamic perspective. Previously in 2.1, you would have a menu bar. When the menu bar was brought up and displayed, you would dim menu items as necessary and execute functions when the menus were selected. Now in UIQ 3, you have a much wider concept, commands. All the things your application does can be seen as an execution of commands. Yet, not all may come from the menubar or the softkeys.

Here's where UIQ 3 becomes powerful. As an example, we can have a look at our spreadsheet application, Quicksheet. Quicksheet has a 'format toolbar' with some options like setting the cell color, the text boldness, italics, etc. In the past, that would be a class on its own, which when certain icons are tapped would do certain things. In UIQ 3, you can just code those items as commands. By ensuring your toolbar (a Quickoffice internal class example) is deriving from MQikCommandOperator, you can now handle the placement of those commands yourself and setting the cell color becomes just another command in the application. Should you now move to a UIQ 3 phone without a touchscreen, this same command can be handled from the menu, rather than a toolbar.

As the example above shows, this paradigm becomes very powerful when you want your app to look different on different devices (e.g. touch screen versus non touch screen). You can have separate MQikCommandOperator's for your commands. That means you don't actually have to modify a lot of code. Your controller / engine still does the same thing for the command (e.g. setting the cell color in Quicksheet), but the command is just coming from a different place (e.g. menubar, softkeys, custom built toolbar, etc).

Your application is much more dynamic and can more easily accommodate different behaviors and different devices. This is because you've separated the command handling from the command displaying.

Although it is easy to port an application from UIQ 2.1 and get it to run on UIQ 3, we would argue that even a small redesign for UIQ 3 is a very good exercise. Just getting things to run is one thing, but redesigning your app to take full use of the UIQ 3 features is the best option. The software platform will likely be around for quite a while. So "getting it right" from the start will give you greater benefits in the long run.

About Quickoffice, Inc.

Quickoffice, Inc is a world leader in providing mobile office document software for smartphones. The company's flagship product, Quickoffice Premier, continues to receive awards and recognition from industry analysts, the media and users alike. The software allows users to open, view, edit and send their original, native Microsoft files without the need for desktop conversion or synchronization. Quickoffice has been thoroughly tested by the rigorous quality standards of leading mobile phone manufacturers and ships through major carriers such as Vodafone®, Orange®, China Mobile®, Cingular® and many more. Fully localized in 36 languages, and shipping worldwide as a built-in solution on millions of smartphones, Quickoffice has been a top-selling mobile office application since its market introduction. Privately held, Quickoffice has offices in Dallas, Texas and London, England.