

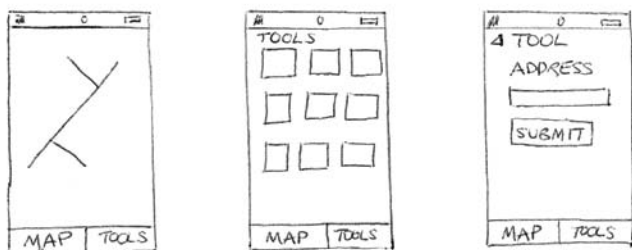
Developing a Custom ArcGIS Application for the iPad 2

By Rory Biggadike, WebMapsolutions.com

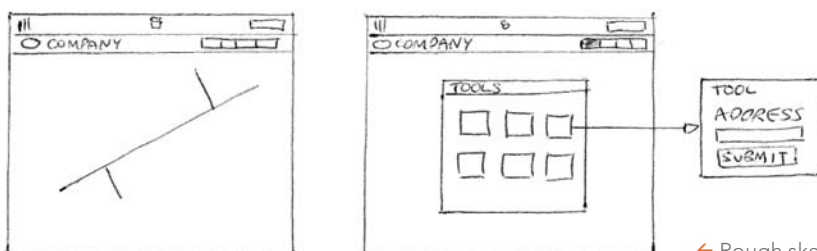
Steve Jobs said we've entered the post-PC era. Goldman Sachs has called tablets "one of the most disruptive forces in computing in nearly three decades." Only time will tell if this is truly the beginning of the end of the PC, but there is little doubt that the advent of mobile computing has already had a huge impact on society. Portability; instant Internet access; geolocation; and simple, more intuitive interaction are among the key benefits. Using ArcGIS on a mobile device in the field, in combination with geolocation (GPS), provides the context missing from existing GIS interaction.

This article is partially a discussion of ArcGIS for Mobile and partially a tutorial on developing an ArcGIS app. The tutorial portion describes the process of building an ArcGIS application using ArcGIS API for Flex 2.4, porting the code to Adobe AIR, and installing the app on an iPad 2. It is targeted at a wide audience and will attempt to avoid technical details. The goal is to provide a base for a broader discussion on real-world applications of ArcGIS on mobile devices.

IPHONE



IPAD



An Introduction to the Mobile Market

The mobile market is made up of laptops, smartphones, and tablets. Historically dominated by the iPhone and iPad, new launches by other manufacturers have started to challenge Apple's preeminence. The most popular platforms (mobile operating systems) are Apple's iOS, Android, Symbian, Windows, and BlackBerry. Mobile device screen size is an important application development and design consideration. Screen sizes range from the 2.6-inch HP Veer through the 3.5-inch iPhone and 9.7-inch iPad 2 to the 10.1-inch Samsung Galaxy Tab. Any ArcGIS map viewer needs to be designed with the screen size in mind. Larger screen sizes lend themselves to more complex applications.

← Rough sketches of application designs

Accessing Mobile Applications

There are two ways to access applications on a mobile device. The first is to simply fire up the mobile web browser and access an existing web application. From a user's perspective, there are disadvantages to this approach. Arguably the biggest is the inability to access applications built using Adobe Flash/Flex or Microsoft Silverlight from any Apple iOS device. This relates to Apple's restrictions on installing plug-ins such as the Flash player. Thus the Esri 2.3.1 Flex viewer can be accessed from most mobile devices, but not the iPhone or the iPad 2. Apple is by no means alone in imposing these types of restrictions.

A second way to access an application on a mobile device is to install the application on the device. Apple, BlackBerry—even Amazon—offer both free and fee-based applications targeted at different platforms. These are downloaded, installed on the device, and optimized for both mobile use and specific OS platforms. Installed applications have advantages over web-based applications. Traditional web pages are targeted at PCs and mouse interaction. Finger interaction is quite different. Installed applications are built with this in mind. In addition, an installed application potentially allows data storage on the device using a lightweight transactional database engine such as SQLite.

There is an additional aspect to installed apps—native versus hybrid apps. For example, applications written in Objective-C are native to the Apple iOS. Applications like those written using Adobe AIR use browser interfaces and native mobile components and



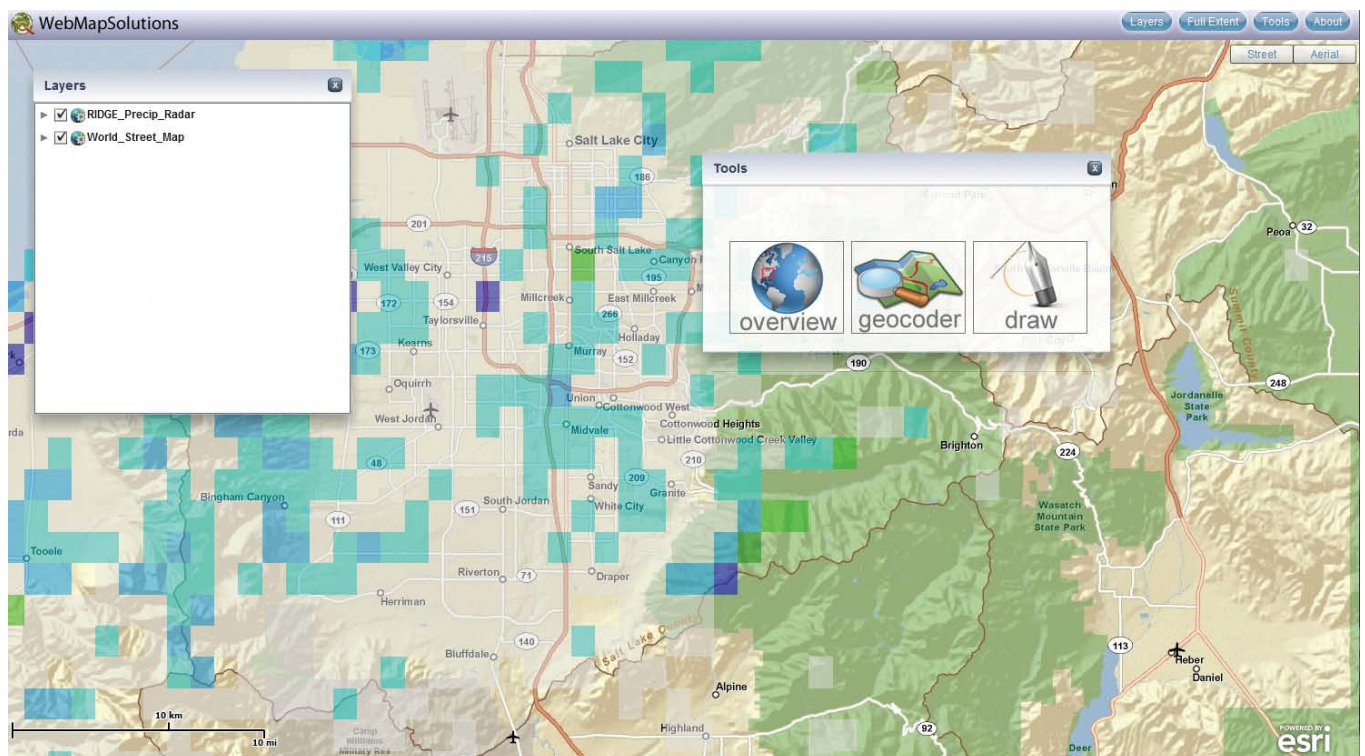
↑ ArcGIS tools pop-up with Tools Dashboard

are hybrid apps. [Adobe AIR (Adobe Integrated Runtime) is a cross-platform development environment for creating applications that can be run on the desktop or mobile devices using Adobe Flash, Adobe Flex, HTML, and Ajax.]

This article will focus on building an installed application using Adobe AIR. (The author has no affiliation with Adobe.) The decision to use AIR was driven by the recent release of Flash Builder 4.5. Using this new version of the Adobe developer tool, it is now possible to build a single application that runs on multiple platforms. With AIR you can create one application with one code base that runs on many devices rather than creating one application that requires multiple code bases (in multiple native languages) to run on multiple devices.

Esri is taking an interesting approach. The ArcGIS community can publish its data through ArcGIS Online. By installing the new ArcGIS native application, users can access this data from their mobile devices. →

↓ The final app (See flexmappers.com/arcuser/v3/.)



ArcGIS for Mobile: Development Process

The next part of the discussion focuses on the development process: from rough design to application framework to actual coding. From a programmer's perspective, this is where the fun begins. For clients and users, it is (hopefully) the evolution of a new and useful tool. This article was written in conjunction with the actual development of the application. It includes thoughts about this process as well as idle thoughts about developing mobile apps.

Design

All ArcGIS application development projects start with design. In practice, how design is done varies. A particularly effective approach is to grab a pencil and paper and start scribbling. The map tools are important and will need some thought. The limited screen real estate on these mobile devices means a dashboard approach might work. A single screen or pop-up shows a list of tools that can be opened within that screen. More on this later.

Application flow is something hard to represent in rough sketches. There are formal ways to step through and define the flow of an application (e.g., from button click to tab change). In this example, the flow will be tested during the coding phase.

Application Framework

It is now time to step gently into the technical part of this article: the development phase. Time has been spent brainstorming. A design—the blueprint for the developer—is in place. But before coding commences, the use of an application framework needs to be discussed.

The developer needs to consider whether the mobile application should be built using a programming framework. Model-view-controller (MVC) is a pattern commonly used with these types of applications. *[MVC separates the user interface from the business logic of the application.]* Swiz is a formal framework popular with Flex and AIR developers. Using a framework like Swiz, which follows the MVC pattern, helps teams work together on an application, provides code organization, and simplifies common tasks. This application will use the Swiz framework.

Application Development

It is finally time to start coding. As promised, the article will avoid most of the technical coding details. This initial coding effort will focus on an iPad 2 application. The work will be done in Adobe Flex initially and then migrated to an AIR mobile app. The first phase is focused on setting up the Swiz framework and designing the application header. The header includes buttons that open widgets managed by the framework.

During the brainstorming session, it was decided that a logo and four buttons (Layers, Full Extent, Tools, and About) should be included in the header. Users commonly zoom to full extent and wish to access the layer list so including these buttons in the header made sense.

The widgets are built from a common code base or template so changes to the general widget's look and behavior will propagate across the widgets. The general widget should be draggable and closable and cannot be moved off the screen.

ArcGIS iPad 2 Application Development

The next phase focuses on styling, basemaps, and layers. Styling is an ongoing process. Styling for buttons and headers uses easily changed style sheets. Using style sheets, with the Degrafa Flash library, headers have been given gradients and buttons have been styled beyond the basic look and feel. Interchangeable street and aerial basemaps can be changed using the buttons in the top right corner and have basic styling that resembles Google maps.

Layers can be added to the basemap. An ArcGIS weather layer is shown being loaded by the application. Using a configuration file, it would be possible for any appropriate ArcGIS endpoints to be loaded by the application in a similar way. Beyond visualizing layers, these widgets define the application and are the focus of the next phase of the development effort.

Widget Development

Four widgets were planned for the application. It has been designed to allow the easy addition of other widgets at a later stage. Any number of widgets could be added to the application. This development effort includes a small subset of widgets for illustrative purposes.

The completed ArcGIS Flex application includes fully functional versions of the widgets. Using the dashboard approach, this tool is selected and opened from within the Tools pop-up.

Flex ArcGIS to AIR ArcGIS

The application has reached a point where porting to Adobe AIR can be considered. Remember Flex is a web technology, designed to run in a browser using the Flash player. Adobe AIR is an installed technology that is independent of a browser. The new Flash Builder 4.5.1 includes mobile support. Thus, building AIR-based applications targeting smartphones or tablets for all mobile platforms just became considerably easier.

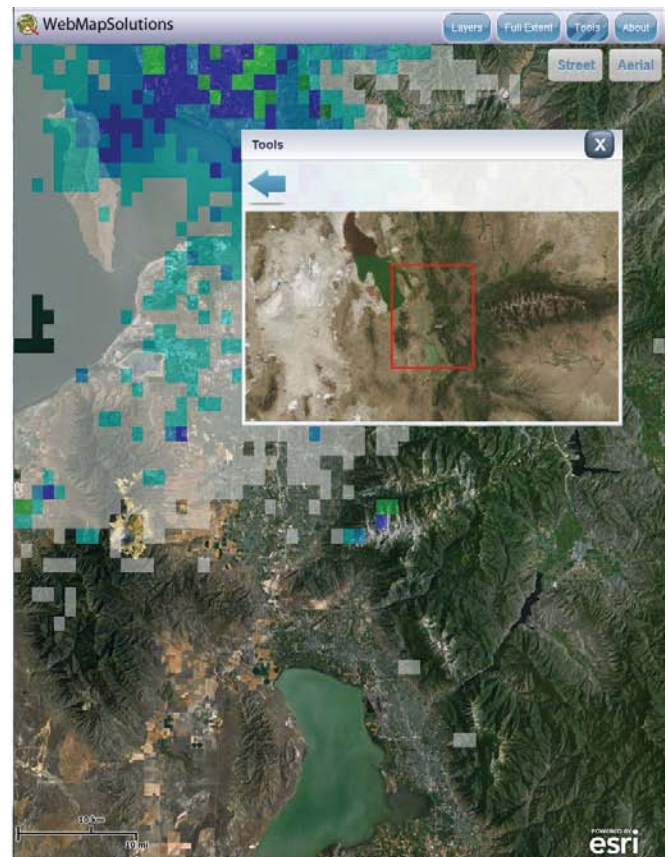
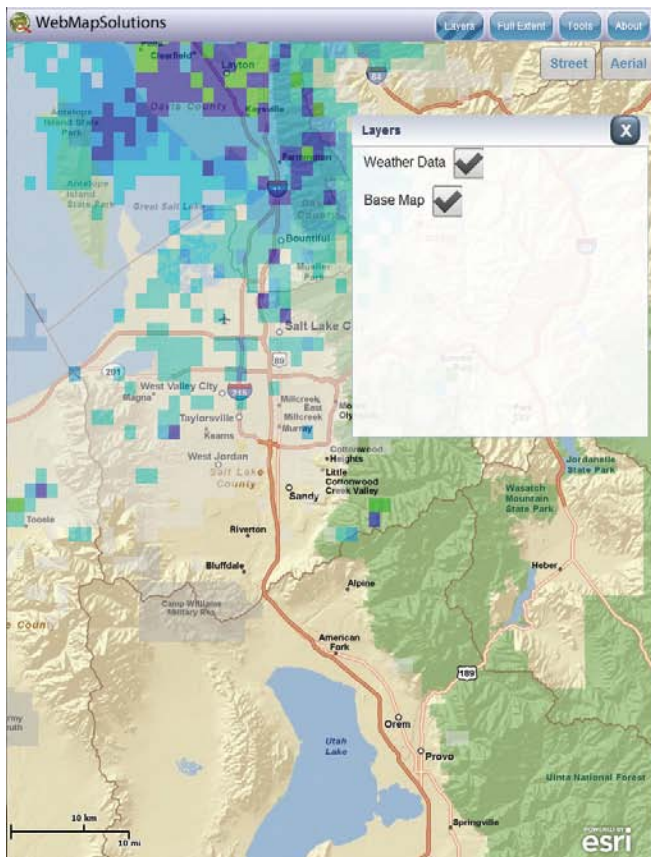
The move from Flex to AIR was straightforward. Some changes were made to the layout, particularly in the tools widget. Since the application is being built for the iPad 2, screen real estate still presents limitations versus a PC. The app needs to be designed with larger buttons that do not overload the interface. The balance must be carefully thought through.

The biggest change was to the Layers widget. This was originally based on a widget available in the Esri Flex 2.4 viewer. Though it ran nicely in the Flex application, in AIR it threw a number of errors. In its present form, the Layers widget cannot be ported to a mobile AIR app so the widget was reworked for the mobile app. All other tools seem to work nicely in the iPad 2 simulator. The next step is running the application on the actual device.

ArcGIS from the Web to the iPad 2

There are a series of steps to walk through to move an application from PC-based development to the iPad 2. The details are beyond the scope of this article. Adobe has made packaging the AIR application very easy in Flash Builder 4.5.1.

Apple has been similarly helpful, making certificate generation intuitive through its iOS development portal. The iPhone/iPhone



↑ The ArcGIS application ported to AIR (See a video showing the app running on an iPad 2 at www.webmapsolutions.com/ios-arcgis-ipad/.)

Touch Application (.ipa) file is a cross-compiled executable generated by Flash Builder that can be uploaded directly to the iPad 2 from iTunes.

Minor changes were made to the layout after testing on the iPad 2. Both landscape and portrait views of the application look good. Esri recently released the 2.4 version of the ArcGIS API for Flex. Many mobile improvements have been made. This has been included in the application and allowed for some code improvements.

Summary and Conclusions

Writing a Flex-based ArcGIS web application and porting it to Adobe AIR for installation on the iPad 2 proved to be both easy and seamless. The release of Flash Builder 4.5.1 and ArcGIS API for Flex 2.4 are significant milestones that have made building ArcGIS apps for mobile devices much easier. This AIR application will run on many mobile platforms beyond the Apple iOS.

Soon, this application will be released to the Apple App Store. This will allow readers of this article to test the application. It is anticipated that users may wish to add their own data to the viewer. This could be done by using a configuration file hosted by the user. This aspect was beyond the scope of the article, but the application was built with the potential for further development in mind.

ArcGIS offers many exciting mobile possibilities, and with the new releases from Esri and Adobe, the community can now in earnest start extending and broadening location-based solutions.

About the Author

Rory Biggadike is senior developer and principal at WebMapsolutions.com. The company builds web and mobile location-based solutions. Biggadike has been developing web-based GIS solutions for 14 years. Much of his development experience has been with Esri software—from MapObjects IMS to ArcIMS and ArcGIS. Mobile has become an important part of his work with a focus on both GIS and location-based services (LBS).

16th annual
GIS SUMMER SCHOOL

JULY-AUGUST
SUMMER 2012

- Project Based Curriculum
- Field Trips & Tours
- Networking Events

Learn More
Online

www.extension.ucr.edu/esri

**MORE
POWER
TO YOU**

UNIVERSITY OF CALIFORNIA
UCRIVERSIDE
EXTENSION