

that mapps will eventually supplant paper maps. Even if one does not agree with that hypothesis, the fact that a cartographer can take a print map, and distribute it globally for free via mobile networks, translates to far more potential customers. Including a Quick Response (QR) code on the print map that links to a downloadable mapp will likely encourage people to pay for a paper map to put in their glove compartment, even if a company only makes print maps for a particular region or locale. The purpose of this tutorial is to demonstrate that transitioning map products from print to mobile is not necessarily an arduous process. In fact, it can be done within 15 minutes. Take your cartographic masterpiece created in ArcGIS or Illustrator and distribute it to millions of people around the globe!

Following the steps below, you will recreate an interactive mapp that originally began as an ArcMap MXD file, was exported as an Adobe Illustrator file, and is now a mapp on the Android Market. Your end product will have simple pinch-to-zoom and pan capabilities. Once you have mastered publishing to the Android Market, you can seek additional tutorials, perhaps including other *On the Horizon* pieces, to add more interactivity to your mapps. You can also take your old FreeHand or CorelDraw masterpiece and refashion it into a mapp using Adobe Illustrator so it remains marketable in the future.

REQUIREMENTS AND OPTIONS

Below are the requirements and optional peripheries needed to create mapps using Adobe Flash Professional CS5.5. If you have an Android 2.2 or higher device, you can download and view the end product of this tutorial from the Google Android Market. Visit market.android.com from any computer and search for “Muehlenhaus Mapp Tutorial.” Download it to any and all of your Android devices right from the Web.

Requirements:

- A PC or Mac computer that is capable of running Adobe Flash CS 5.5
- An Internet connection
- Flash CS 5.5 (a free 30-day trial can be downloaded from www.adobe.com)
- A Gmail account (also available for free)
- The zip file is available at www.muehlenhaus.com/tutorials/mapp.zip
Download and unzip it to a place you can easily access.

Options:

- An Android device running Android 2.2 (Froyo) or higher with a USB2 cable
- \$25 to register as an Android Developer

IMPORTING AN AI MAP INTO FLASH CS 5.5

As already mentioned, many professional cartographers and designers use Adobe Illustrator to create or stylize their final products. If you have never

used Adobe Illustrator, but are comfortable with ESRI ArcMap, you can still easily turn your paper maps into mobile mapps. Simply open your GIS map in ArcMap, go to File > Export, and under File Type, select AI (i.e., Adobe Illustrator). You will not need to install Adobe Illustrator, because you will directly import the AI file into Adobe Flash. For your convenience, I have posted the Adobe Illustrator file used in this tutorial (it is named map.ai and can be found at www.muehlenhaus.com/tutorials/mapp.zip).

SETTING UP FLASH FOR ANDROID DEVELOPMENT

The first thing you will see when you open Flash CS 5.5 is the splash screen. Do not close this. In the left-hand column under Create from Template, select AIR for Android. (If you were to create a mapp for Apple's iOS, you would select iPhone OS from the Create New column.) You will be provided some template options to choose from. All of the template options are 800 x 480 pixel stages. One is just a blank stage; the others offer some built in functionality—a swipe gallery, an accelerometer, and an options menu. In this tutorial, you will select the third one on the list, Options Menu. This template has a built-in menu that pops up when someone presses the menu button on their Android device. This pop-up has a built in Exit button, to which you can add more options, too, if you so desire. All of the templates may prove useful for your mapping needs, and I recommend experimenting with them. However, this tutorial will simply concentrate on bringing our static map into Flash CS 5.5, giving it pinch-to-zoom capabilities, and publishing it to the Android Market. After selecting the Options Menu Android template, you should immediately go to File > Save. Save the file as any name you like, but make sure you save it within the extracted folder created from *mapp.zip*.

SETTING UP THE STAGE IN FLASH

In most cases, you will want to change the stage (Flash's artboard). In the case of this tutorial, our map is meant to be displayed in landscape mode and it must have a white background, so the first thing we will want to do is set the stage to landscape and change the background color. Without selecting anything, open the Properties dialog (typically found along the right-side of the Flash program window).

Under the Properties tab in the Properties dialog, you will see several options, including Size and Stage. Reverse the numbers found by size by typing 800 into the box containing 480 and 480 into the box containing 800. Finally, select the color icon adjacent to Stage. Select the "Background color:" White.

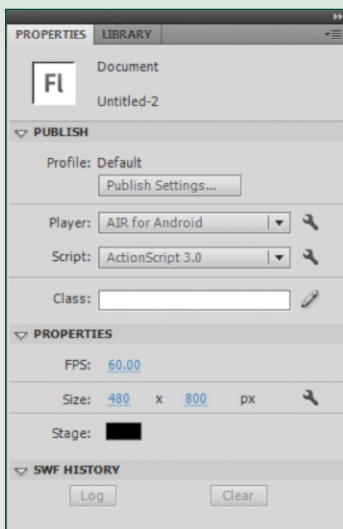


Figure 1. PROPERTIES panel.

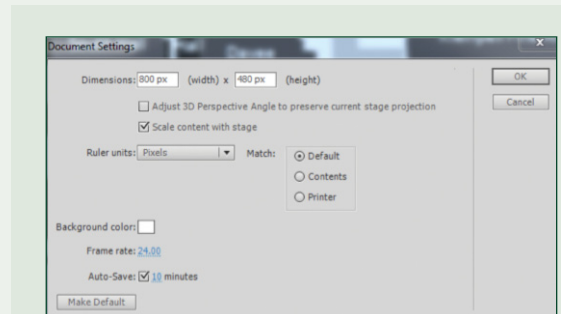


Figure 2. Setting the size dimensions.

DELETING THE DEFAULT BACKGROUND

The stage will realign itself correctly, but you will still have a black background with a fading grid on it. We will delete this grid and black background now. Go to the Layers panel and find the layer called Background. Click on the layer where it says Background. Underneath this you will see a Trash Can icon. Click it. A dialog window may pop up. Select Yes. The background will disappear.

REPOSITIONING AND ALIGNING THE OPTIONS MENU

Another problem with realigning the stage is that our Options box with the Exit button is now not aligned to the stage. Fortunately, this is easy to fix. If there is a padlock icon next to the Options layer, unlock it by clicking on the icon. Then use the Selection tool to select the Options box on the stage. Move the box to line up with the lower left corner of the stage. Once it is lined up with the stage, you can resize it. First, double-click on the box. The box was a symbol. By double-clicking, you are now working inside of the Options symbol. If you look at the timeline/layers panel again, you will see two layers. The bottom layer is the gray box. Select the gray box without selecting the Exit button by double-clicking anywhere over the gray box. Once it is selected, press Q on the keyboard. Resizing handles will appear. Grab the handle on the right-side and drag it to the right edge of the stage. Lock Layer 1 by clicking on the padlock symbol in the Timeline panel. If it is locked, unlock Layer 2. Choose the Selection tool and drag a box around the Exit button and text. Move to the desired location over the gray box, usually in the center. When you are done realigning the Options box, double-click anywhere outside of it to get back to the main stage.

IMPORTING THE ILLUSTRATOR MAP INTO FLASH

Now that the stage is set, we can import our Adobe Illustrator file into Flash. Go to the Timeline panel. Lock all of the layers by clicking on the padlock symbol above all of the layers. If the Options layer is not the top layer, click on it and drag it above the other layers. (We need our Exit button to appear on top of all other layers.) Then highlight the second to top layer (the Instructions layer in this example) by clicking on its title. Beneath the titles and to the left of the Trash Can icon is an insert New Layer icon. Click it. A new layer will be created. Double-click on its title to rename it whatever you want—in this case, I labeled it “map.”

With the new layer highlighted, go to File and select “Import to Stage.” Some seconds will pass as Flash prepares to import. A dialog will appear asking you to choose a file to import. (Note: you can import a variety of file types beyond an Illustrator file, and it is recommended you take a look to see what other formats you may be able to import in the future.) Find the mapp.ai file in the files you downloaded with this tutorial. Select it and after some time you will see a dialog similar to that in Figure 3.

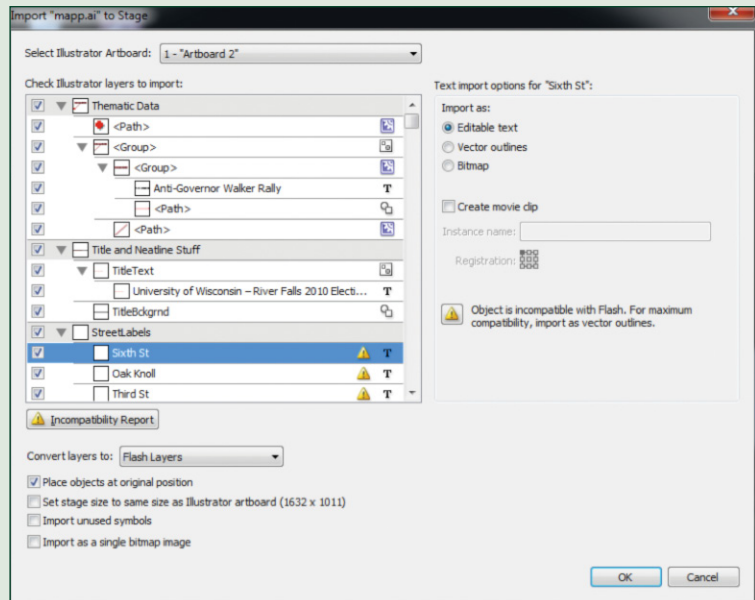


Figure 3. Import to Stage options.

Minimize all of the layers to a single row each by clicking on the arrows to the left of the layer titles. Then hold down the shift key and select all of the layers together. To the right you will see a “Create Movie Clip” box. Check it. Underneath the layers you will see a rectangular button with a warning symbol and the words “Incompatibility Report” (see Figure 3, number 2). Click this. Many times AI files contain text that is not compatible in Flash. There are several different techniques you can use to import text and make it work, but in the case of this exercise, we will default to what Adobe thinks is best. Check the box next to “Apply recommended import settings” and hit Close. Beneath

where you just selected you will see text that says “Convert layers to:”—select “Single Flash Layer.” Make sure that the “Place objects at original position” dialog is selected. Select “OK.” Your final dialog screen should look similar to Figure 4.

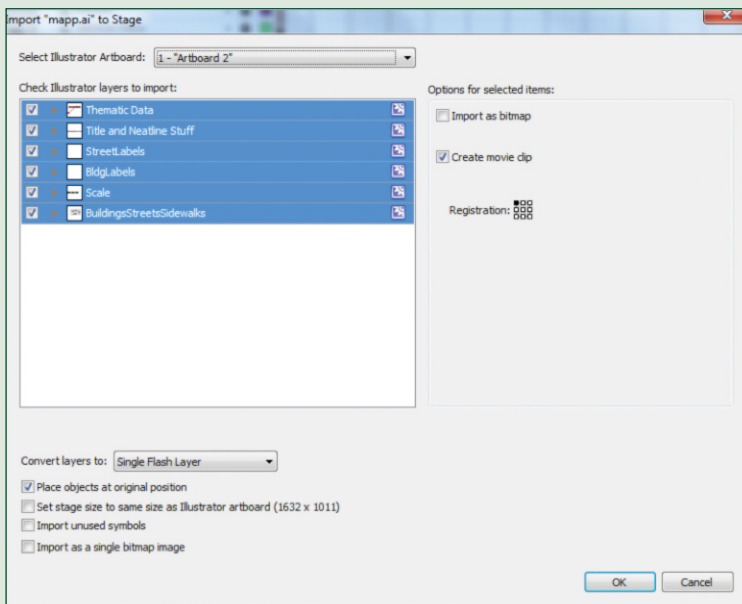


Figure 4. Final import options for mapp.ai

CHECKING AND FIXING MAP IMPORTATION GLITCHES

Several things will stand out upon the map’s importation. First, it doesn’t fit the stage. Second, each layer of the map is surrounded by its own selection box. Third, some of the text may be disheveled—even though we took Adobe’s recommendations for importation. The first and second glitches are easy to remedy, and will be resolved in the next several paragraphs.

The third glitch can be more problematic. With a complicated map, you may want to undo the Import and go through the process again, double-checking each incompatibility error to see if a better solution (i.e., something other than the default) presents itself. Another option is to delete the bad text in Flash and retype it with the Flash text authoring tool. In the case of this map, I recommend exercising your cartographic right to generalization—just delete any extraneous textboxes!

Once you have imported the AI file, all of the map layers should be on a single Flash layer that was renamed “maps.ai.” In order to add pinch-to-zoom functionality, you need to make all of the mapp components a single MovieClip (or symbol) with a centered registration. (If you are not entirely sure what this means, do not worry. Just follow the forthcoming directions.) Select everything in the mapps.ai layer by clicking the box with a dot in it on the layer under the number 1 (i.e., frame #1) in the timeline. Once all of our map components are highlighted with blue boxes go to Modify > Convert to Symbol, or simply press F8. Select MovieClip and give it a Name of “mapp_mc.” Where it says Registration, be sure to click the middle box. Then press OK. All of the separate selection boxes should meld into one.

RESIZING THE MAP TO FIT THE MOBILE DEVICE

Resizing your map is easy. Select the map on the stage. Go to Properties. If you do not, you may need to expand the Properties window a little bit by dragging the bottom edge down. Under Properties you will see the width and height of your map—make sure it is selected, otherwise you will not see it. Since we know that the stage is 800 x 480 pixels, we can make our map similarly sized by just typing in these values. Make sure that the lock symbol to the left of the width and height values is set to “Lock width and height values together.” This way your map will not become distorted when you type in new values. Type 800 for the width. Then, using the Selection tool, drag the map to the upper left-hand corner of the stage. It should fit almost perfectly.

ADDING CODE HINTS FOR MULTITOUCH PINCH-TO-ZOOM AND PANNING

The great thing about the Flash platform is that you do not have to be a programmer to create interactivity. Knowing some ActionScript will help, but fortunately, Flash provides simple “code snippets” that all users of Flash can simply insert into their documents to add some interactivity. Of primary interest to most mapp makers is making it possible to pinch-and-zoom or pan on any given mobile device. At this point, your Illustrator document is sufficiently imported into Flash and is ready to be made into a mapp. If you do not want to add pinch-to-zoom or panning to your map, you may skip the rest of this section. If you would like to make your map a little interactive, follow the directions below.

ADDING PINCH-TO-ZOOM CAPABILITIES

Select your “mapp_mc” MovieClip symbol on the stage. (Once selected, it should be highlighted with blue around it.) In the Properties panel, type “mapp” into the “<Instance Name>” text box. Go to Window > Code Snippets. You will see the Code Snippets dialog; select Mobile Gesture Events > Pinch to Zoom Event by double clicking on it. The Actions dialog will open up (Figure 5). You will see code has been written for you. There are some simple instructions accompanying the code. If you do not understand the code at all, do not worry; it will work. In order to put pinch-to-zoom on your own maps in the future, you merely need to make all of your map layers into a single MovieClip—as we did with this example above—put the registration point in the center, select your map, and choose “Pinch to Zoom Event.”

```
/* Pinch to Zoom Event
Pinching the stage will zoom in on the selected object.

Instructions:
1. The symbol's registration and transformation points should be at the center.
2. The symbol should be the background of your application and sized to match the stage.
*/

Multitouch.inputMode = MultitouchInputMode.GESTURE;

stage.addEventListener(TransformGestureEvent.GESTURE_ZOOM, fl_ZoomHandler_3);

function fl_ZoomHandler_3(event:TransformGestureEvent):void
{
    mapp.scaleX *= event.scaleX;
    mapp.scaleY *= event.scaleY;
}
```

Figure 5. Automatically generated code for the Pinch to Zoom Event in the Actions window.

ADDING PANNING CAPABILITIES

You often will want to include a panning option when your map is zoomed in, as it goes over the edge of the mobile devices screen,. Unfortunately, panning in Flash is still relatively clunky. It requires two fingers, which many people find unintuitive. Unintuitive or not, it is useful. Adding Panning is as easy as Pinch-to-Zoom.

Select your “mapp_mc” MovieClip on the stage. Open the Code Snippets dialog. Go to and double-click Mobile Gesture Events > Pan Event. Code will be added as it was for Pinch-to-Zoom. You will now be able to pan using two fingers adjacent to one another.

SETTING UP ADOBE AIR TO PUBLISH TO THE ANDROID PLATFORM

Your map is now a mapp. Yet, we still need to do a few things in Flash to get it ready for the Android Market. Flash uses Adobe AIR to export the Flash file as an APK file (Android App). AIR needs some data and information from you before exporting. The first thing you will do is hit SAVE. Sometimes

Flash freezes when exporting using AIR. Once you have saved your file, deselect everything by clicking off-stage somewhere with the Selection tool. Then click on Properties. You should click the Wrench icon next to where it says “Player: Air for Android.”

There is a lot of information that needs to be filled out. There are four tabbed menus (Figure 6). Start with the General tab. You should decide what you want your output file to be called—your actual app file. It must end with “.apk.” Then, you should give your mapp a name that will be visible in the Android Market. Finally, you should give your mapp an i.d. All AIR apps start with “air,” but you can add your own naming convention afterwards. You also must give your mapp a version number. (Importantly, when you make changes to your mapp in the future and want to upload the new mapp to Android Market, the mapp must have a higher version number than the one previously uploaded. This way everyone with your mapp installed will get a message from the Market that an update is available for download.) For this map, because it is best viewed in Landscape mode, you should select “Landscape” under Aspect Ratio. I prefer that my mapp takes up the full screen, so I typically select that as well.

The Deployment menu requires a little more work. First, you need to create a Developer’s Certificate. Basically, this is a certificate that validates the mapp as being yours. If you do not have one yet, you easily can create one by clicking on the Create button. You simply need to fill in the required information, create a password, and, once it is created, type your password into the box below. Make sure that the certificate is good for at least 25 years, as Android requires this for apps to work. Select “Remember Password for This Session” if you do not want to type the password in every time you test your mapp. Select “Google Android Market” as your AIR runtime. Also, if you have the Android Software Development Kit (SDK) installed, you can plug in an Android device (e.g., I use the Nexus One) and when you hit publish, the mapp will be installed and brought right up on your phone for actual device testing. If you do not know what the Android SDK is, or if you do not have an Android device, simply leave the “After Publishing” options deselected.

The Icons tab is one of the most important in many ways. This is where you decide what icon you would like to show up in the Android Market when people search for your app. It is also the icon that will show up in user’s app drawers or as their app shortcut. The icon must be one of three sizes (36 x 36, 48 x 48, or 72 x 72 pixels). You may need to use Photoshop or another graphics software editor to make this. You cannot create an app without an icon. I have one ready with my business’s logo (included in mapp.zip) on it at all times, particularly when I am testing a mapp, so I can just plug the icon in and move on to testing without too much hassle. Save a logo somewhere, browse to it, and select it.

The Permissions tab menu is easier than the others to fill out, but these settings should not be taken lightly. It is here that you can decide what aspects of a user’s phone you want to access and potentially control. If you make a map and want people to be able to take a picture and have the picture be embedded on your map at the exact location they took it, you will need to check the “Camera and Access_Fine_Location” boxes. One thing to remember: people using smartphones are increasingly wary of malicious software. Users will see what

your app is requesting permission to access before installing your mapp. If your mapp is merely an atlas of census data and your users see that you want to access their exact location, take pictures with their camera, record audio, and write to their storage space, hopefully they will not balk at installing your product. Select only what you need for the mapp to function properly. I almost always include a link inside of my mapps, so checking Internet access is a must here. (Note: if you, too, want to include a link to a webpage, look under Code Snippets and it will show you how.) Most users expect apps to have Internet access and will not be wary of installing your mapp.

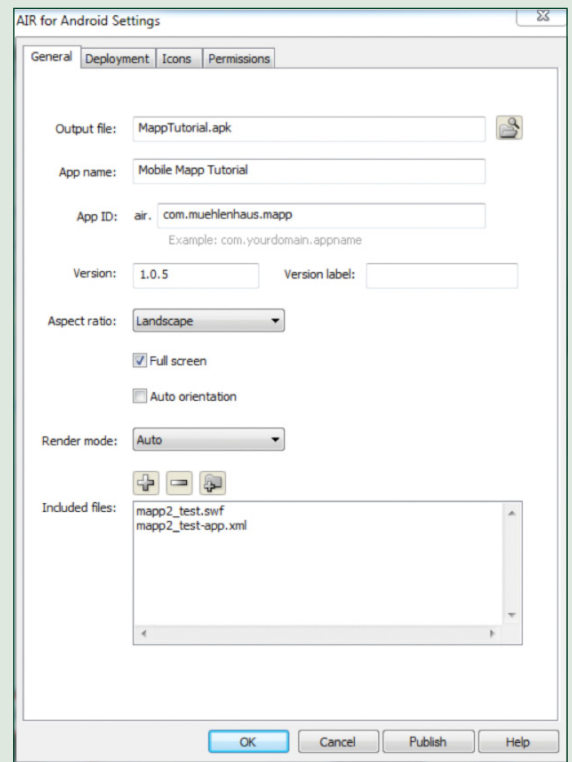


Figure 6. AIR for Android settings. All four tabs contain options that must be filled out.

CREATING YOUR MAPP FILE

Once you have taken care of the prerequisites found under each tabbed menu, you are ready to publish. I typically hit OK, and save my file before continuing so that, if Flash crashes when publishing, I don't have to refill all of the information. When you are ready, click on Publish as found in the AIR for Android Settings dialog. This will produce both a SWF file and also an APK file that can be uploaded to the Android Market. (If you do not necessarily need an APK file yet, as you are still going to make some changes to your mapp, you can simply hit Ctrl-Enter to see what your mapp would like on a mobile device. Note, however, that you will still need to select Publish at some point before you can upload an app to the Android Market.)

It may take some time, but eventually a new window should open with the Adobe AIR icon and the name of your mapp. You will see your mapp as though it were on a device. If you want to test the Exit button, go to Device > Menu. (This mimics hitting the Menu button on an Android device.) The Exit button appears. If you click on Exit, the window closes. This represents that the mapp would close on the phone—i.e., your mapp is working!

The creation of your mapp is done. You are now ready to upload it for the world to use!

SETTING UP AN ANDROID MARKET DEVELOPER'S ACCOUNT

Before you can post your mapp for the world to see—or, if you are an entrepreneur, purchase—you must set up an Android Market Developer account. There are now many Android markets, but Google's is the first stop for most shoppers and developers. Others you may want to register to develop for include Amazon's App Market and Barnes & Noble's NOOK Market. The ideal thing about Google's Android Market is that there is no review or censorship process as there is with nearly all other markets—most notoriously Apple's iTunes—so you can see your results very quickly. Also, Google's Android Market is remarkably cheap to register for compared with most other markets. Apple and Amazon charge \$100 a year to be a mobile developer. Google charges a one-time fee of \$25.

Becoming an Android developer is remarkably easy. Have a credit card readily available (or a Google Checkout account) and go to market.android.com/publish. Log in if you have a Google Account. If you do not, create one right there and then log in. You will see a very simple screen asking you for \$25. Hit continue. You will be asked for company information, credit card information, an address, and other such information. Fill in the form and finish the transaction, and you are ready to upload your mapp to the Android Market.

PUBLISHING YOUR MAPP TO THE MARKET

Once you have set up your account, it is largely a straightforward process to get it online. However, there are a few things you need to do before sending off your mapp. Once you log into the publisher site, you should see a screen similar to that in Figure 7, though with no apps listed. Select "Upload Application." You will see an Upload New APK dialog. Browse to where you saved your Flash file and you should see an APK file by the same name in the folder. Select it and hit upload.

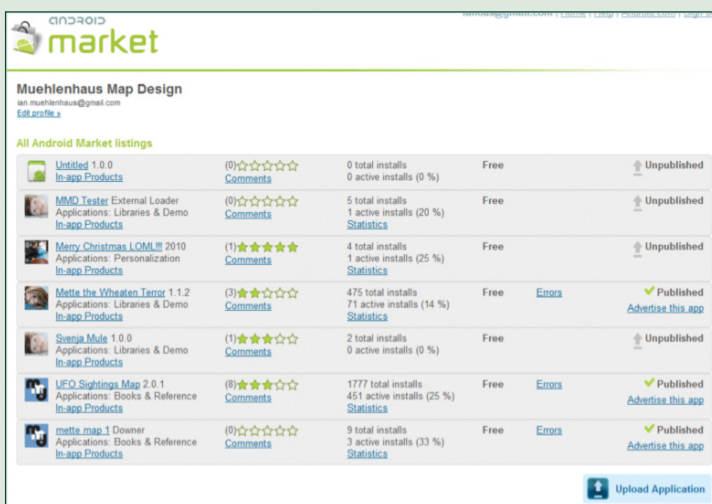


Figure 7. Android Market publishing screen.

The Android Market is beginning to require a minimum amount of information about your apps before you can upload them (see Figure 8). The requirements include at least two screenshots that mobile users can view before deciding whether or not to download, and a high resolution application icon—this is the icon that will be used at various places in the Android market for advertising your app. These images must be made to the sizes specified by Google, and again, you may need to edit and crop your images in a graphics application. You can provide a link to a promotional video, feature graphic, and a promotional graphic; these are not required, however.

Fill in the rest of the information as you see fit. On this page you can also determine in which countries you would like your mapp to appear on the Android Market. You also can decide whether you want to charge a fee for your app. (Note: if you do, you will have to set up a financial account with Google. That is not difficult, but requires some more forms.) You also must give your app a rating, so users know whether it is suitable for children. Finally, there is the option of selecting copy protection. You can decide whether your map is valuable enough to warrant the larger file size it entails.

Near the top of the screen you will see two tabs; the one you are on says “Product Details.” Next to it is one that says “APK Files”; the second tab is where you upload your actual mapp to the servers if you did not do so earlier. If you already uploaded it, you still need to activate your app on the market. Click the APK Files tab. You should see your APK file with the icon file you selected

The screenshot shows a web form titled "Listing details" for an Android Market listing. The form contains the following fields and values:

- Language:** English (en) (with a link to "add language" and a note: "Star sign (*) indicates the default language.")
- Title (English):** From Print to Mobile mAp (26 characters, 30 max)
- Description (English):** This mapp was created using nothing more than Flash CSS.5 and an ESRI ArcGIS map exported as an Adobe Illustrator file. To learn how to take your GIS and map projects and make them mobile mapps, please read "Cartographic Perspectives" available on the North American Cartographic and Information Society Website (www.nacis.org). There a tutorial using this mapp is available in an issue from 2011. (580 characters, 4000 max)
- Recent Changes (English):** The only change made to this mapp since writing the tutorial is... I added a link to the NACIS Website. Click on the middle part of the map to have your browser take you there. (178 characters, 500 max)
- Promo Text (English):** (Empty field, 0 characters, 80 max)
- Application Type:** Applications (dropdown menu)
- Category:** Education (dropdown menu)

Figure 8. Required details for listing in the Android Market

in Flash. To the right there will be several options: Activate or Delete. Click on Activate. Congratulations! You have just created and distributed your first mapp.

Your mapp will be live to the world and downloadable within an hour, but normally well before then. You can create and upload as many mapps or apps as you want. You are now an Android Developer!

WHAT ABOUT DEVELOPING FOR THE IPHONE AND IPAD?

The methods laid out in this tutorial also can be used to create mapps for Apple's iOS devices and other companies' smartphone technologies. In order to create a mapp for these devices, you simply need to register as an iTunes developer. The fee, as of this publication, is \$100 a year. When you open Flash, instead of selecting an AIR for Android template, select AIR for iOS. The rest of the process in Flash will be remarkably similar. However, Apple must approve any app you create; the review process may take some time, whereas with the Android market your app is typically downloadable within minutes of posting. Also, Apple is known to decline apps without providing any real feedback as to why. For example, the Opera Web Browser was only accepted by Apple when given a Mature (17+) rating.