

A Guide to App Studio 9.2

Contents

App Studio	4
Note for users of App Studio 9.1	5
Understanding App Studio	6
Understanding the AVE-Mag and AVE-Doc formats	7
Understanding layout families	8
Understanding App Studio publishing	9
Understanding App Studio Factory	11
Understanding guided reading	12
Creating an App Studio issue	13
Creating an App Studio issue for AVE-Mag	13
Creating an App Studio issue for AVE-Doc	14
Adding interactivity to an App Studio issue	14
Adding an interactive picture to an App Studio issue.....	16
Adding a picture slideshow to an App Studio issue.....	17
Adding a layout slideshow to an App Studio issue.....	21
Adding a movie to an App Studio issue.....	25
Adding audio to an App Studio issue.....	27
Adding a button to an App Studio issue.....	28
Adding an HTML view to an App Studio issue.....	29
Adding a scrollable layout to an App Studio issue.....	31
Removing interactivity from an item.....	35
Working with interactivity actions	35
Creating a Change Content Index action.....	36
Creating a Go to Web action.....	36
Creating a Go to Page action.....	37
Creating a pop-up.....	37
Creating a Show Fullscreen action.....	38
Creating Play, Pause, Stop, and Toggle Play actions.....	39
Creating page stacks	39
Using dual orientations	40
Synchronizing content between orientations.....	41
Setting the Contents page	41

Reviewing Digital Publishing asset usage.....	41
Exporting an App Studio issue.....	42
Setting App Studio issue export options.....	43
Replacing resources in exported App Studio issues.....	45
Installing Quark App Studio Issue Previewer.....	46
Testing an App Studio issue.....	47
Creating an App Studio app.....	50
Registering devices on the App Studio publishing portal.....	50
Registering with Quark App Studio Issue Previewer.....	50
Registering on the App Studio Publishing Portal.....	51
Creating an App Studio app.....	51
Preparing to submit an App Studio app to Apple.....	51
Building an app with App Studio Factory.....	59
Purchasing an App Studio app template license.....	61
Exporting an app with App Studio Factory.....	62
Testing an App Studio app.....	62
Installing an App Studio app on your iPad.....	63
Testing an App Studio app in the iOS Simulator.....	63
Managing App Studio issues.....	64
Creating an App Studio Publishing Portal account.....	64
Registering an App Studio app.....	65
Creating an App Studio Publishing Portal title or collection.....	66
Purchasing an App Studio issue license.....	67
Posting App Studio issues to a Web server.....	68
Publishing an App Studio issue.....	68
Setting up in-app purchases.....	69
Setting up auto-renewable subscriptions.....	71
Creating an App Studio product.....	71
Downloading an App Studio app certificate.....	72
Submitting your app to Apple.....	74
Legal notices.....	75

App Studio

The App Studio feature lets you create a customized app for the iPad®, distribute that app through the Apple® App Store, and then create and publish content that your customers can purchase and download from within the app.

- ➔ App Studio for Windows requires .NET Framework 3.5 SP1 for Windows XP. All editions of App Studio require the latest Flash plug-in.

Note for users of App Studio 9.1

Due to iCloud-related changes in Apple's policies, if you built an app that uses non-embedded issues with version 1.0 of App Studio Factory, you must rebuild that app with App Studio Factory 1.1 or later, and you must submit the rebuilt app to Apple as a new version of the app.

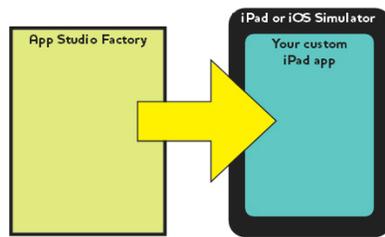
If you have received an e-mail from Apple stating that your app stores a fair amount of data in its Documents folder and that you should instead store that data in the application bundle, the caches directory, or the temp directory, rebuilding your app with App Studio Factory 1.1 or later will fix this issue.

Understanding App Studio

The App Studio feature consists of several parts.

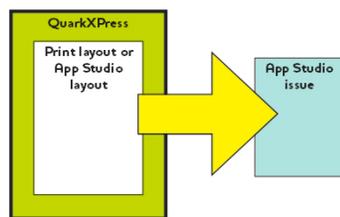
There are two parts to any App Studio solution:

- An *App Studio app* is an app that you build with an application called App Studio Factory or with the App Studio Framework. (App Studio Factory is included with QuarkXPress. You can license the App Studio Framework if you want to develop your own App Studio apps.) After you build and test your App Studio app, you can submit it to the Apple App Store. If it is approved, it will be made available to your customers in the App Store.



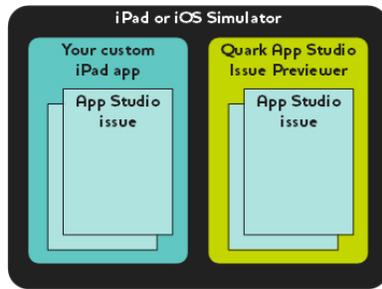
You can use App Studio Factory to create your own custom iPad apps.

- An *App Studio issue* is the digital equivalent of a book or a magazine. You can create and export App Studio issues in QuarkXPress.



You can use QuarkXPress to create App Studio issues.

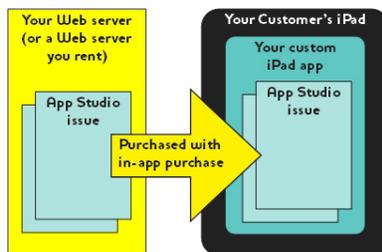
Regardless of your business model, you must produce these two components separately, and your customers will use your App Studio app to view your App Studio issues. You and your customers can also view and test App Studio issues in an app named Quark App Studio Issue Previewer, which is available from the Apple App Store. You can also run this app in the iOS Simulator on your Mac OS computer.



You can view App Studio issues in your own custom app or in Quark App Studio Issue Previewer.

Once you've created your app and your issues, you can make the issues available on a Web server, set up in-app purchases for each of your issues on the Apple developer Web site, and then submit your app to the Apple App Store.

Once Apple approves your app and its content, your customers will be able to download your App Studio app from the Apple Store and use it to browse, purchase, and read your App Studio issues. And you don't even have to set up your own eCommerce solution — the money from issue purchases goes automatically into your account through Apple's in-app purchase feature.



Your customers can buy your issues from within your custom app. When they do, the issues download to their iPad from your Web server (or a Web server you rent).

For more about how App Studio issue publishing works, see "[Understanding App Studio publishing](#)."

Understanding the AVE-Mag and AVE-Doc formats

You can create an App Studio issue in one of two formats:

- **AVE-Mag** format supports scrollable layouts and different layouts for horizontal and vertical orientations. It does not support spread viewing, however, and it does not support zooming. This format is most appropriate when you are creating layouts from scratch for viewing on the iPad.
- **AVE-Doc** format supports spread viewing and zooming, but it does not support scrollable layouts and different layouts for different orientations. This format is most appropriate when you want to convert existing layouts for viewing on the iPad.

You can export an App Studio issue from two different types of layouts:

UNDERSTANDING APP STUDIO

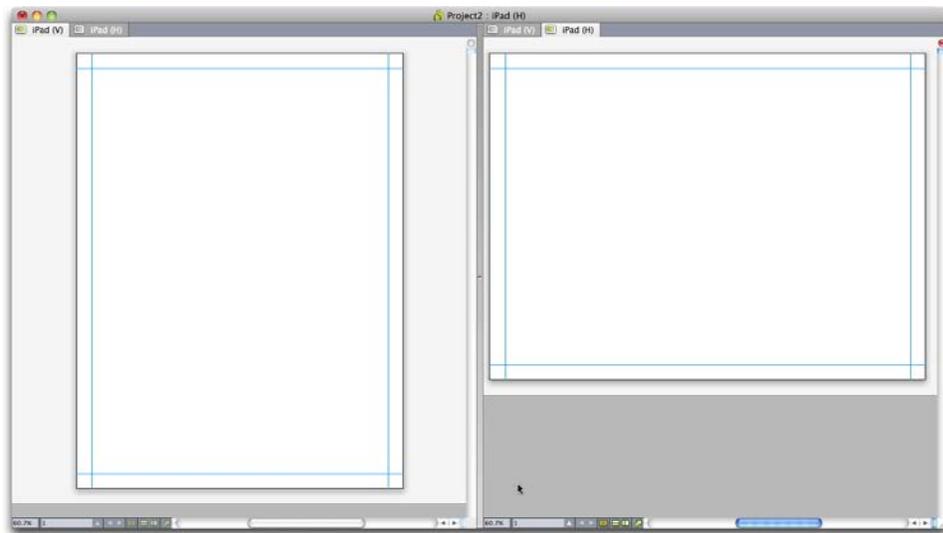
- You can export a Print layout as an AVE-Mag issue or an AVE-Doc issue. If you use a Print layout, however, you cannot use dual orientations.
- You can export an App Studio layout in AVE-Mag format only. For more information about App Studio layouts, see "[Creating an App Studio issue for AVE-Mag.](#)"

You can create a project with a new Print or App Studio layout from scratch, or you can convert an existing Print layout for use on a digital device.

➔ If you are planning to create a AVE-Doc issue, the layout can be any size.

Understanding layout families

A QuarkXPress project file can contain multiple layouts (for more information, see "Projects and layouts" in *A Guide to QuarkXPress*). When you create an App Studio layout, QuarkXPress creates a separate layout for each device/orientation combination you checked in the **New Project** dialog box. All of these layouts are members of a single *layout family*.



A layout family displayed in split view. Each tab across the top represents a layout that is part of the layout family.

A layout family is a group of layouts that represents a single App Studio issue. One layout family might include a horizontal layout for the iPad, a vertical layout for a different type of tablet, and both vertical and horizontal layouts for a third type of tablet. All of the layouts in a layout family are bound to one another in terms of page count; if you add or delete a page in one of a layout family's layouts, that page is added or deleted from *all* of the layouts in the layout family.

Layout families make it easy for you to keep content synchronized between the various editions of each page of content. For example, assume that you're working on a 30-page issue, and you discover that you need to add a new page after page 12. You can go to any of the layouts in the layout family and add the new page, and the page is automatically

inserted in every other layout in the layout family. This is much easier than manually adding the new page to each layout one-at-a-time.

When you create a layout family, QuarkXPress displays the project in a split view. You don't have to work on the layout in split view, but it can be helpful when you're trying to keep different editions of a page consistent.

Another helpful feature for App Studio issues is the shared content feature. This feature lets you put copies of a picture or a chunk of text into different layouts, and then automatically keep the copies synchronized. So, for example, if you correct a name on page five of one layout, the name can be automatically corrected in all of the other layouts too. For more information, see "Working with shared content" in *A Guide to QuarkXPress* and "[Synchronizing content between orientations](#)."

Understanding App Studio publishing

How does content make it onto your customers' iPad? To understand how App Studio issues are delivered to App Studio apps, you must understand how the Quark App Studio Publishing Portal (<http://appstudio.quark.com>) works.

Each App Studio app you create can host one or more *titles* or *collections*.

- A *title* is a periodical, such as a magazine or newspaper.
- A *collection* is a group of related non-periodical issues. For example, if you are releasing a group of related books that you want to make available in a particular app, you could put them into a collection.

Assuming that the app template you use supports it, you can publish one or more titles and/or collections in an app. Also, you can distribute a particular title or collection through more than one app.

An *issue* is a particular member of a title or collection. For example, if you release a given title monthly, you would be creating twelve issues of that title per year.

You can create an App Studio issue in QuarkXPress, export it from QuarkXPress as an App Studio issue (.zave) file, and then upload the .zave file to your Web server. Your customers can then purchase and download the issue from within your app.

- ➔ An App Studio issue file has a .zave suffix, but is actually a ZIP archive that contains all of the assets and data used by the issue.

Each issue has an *issue type*. The issue type represents a particular type of issue with its own price point. For example, for a title that represents a magazine, you might have one issue type for regular issues, plus a second issue type for special issues that cost a little bit more. (The issue type for regular issues is called the *default issue type*.) For a book collection, you might have a default issue type for regular books, plus an additional issue type for books that cost more because they have more pages or are by a particular author. You can name issue types any way you like.

So how do you get paid for selling issues? There are several ways:

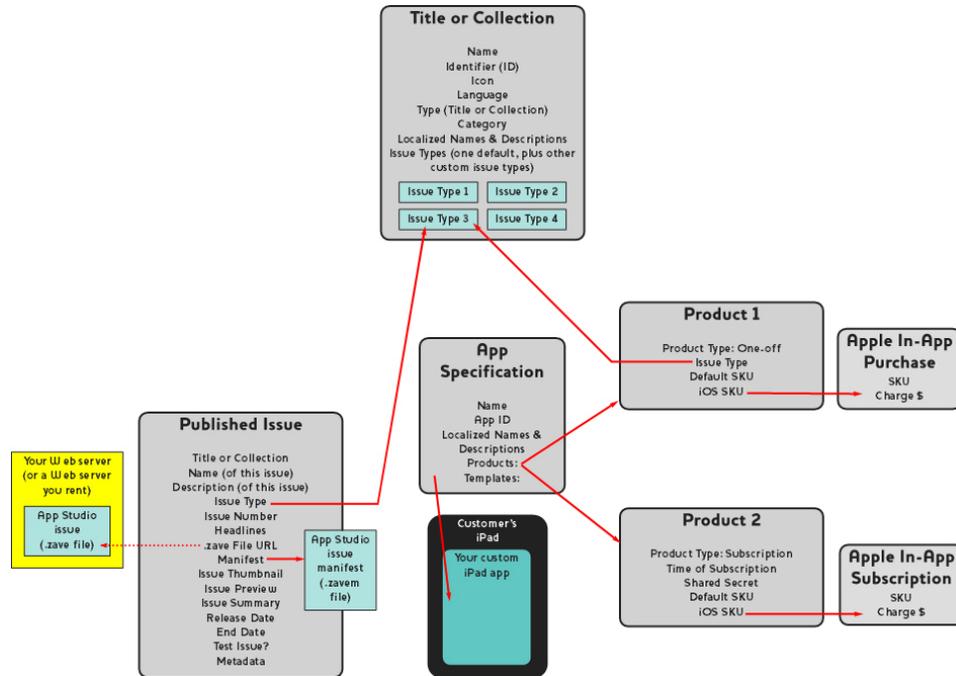
- You can use an Apple consumable in-app purchase. Your customer chooses to purchase each issue individually. A single in-app purchase entry can accommodate all issues of a particular type; for example, you might have one in-app purchase for regular issues, plus a second in-app purchase for special issues.
- You can use an Apple subscription in-app purchase. Your customer purchases a subscription, and then they can download each issue as it becomes available without having to make another purchase. When the subscription runs out, it renews automatically according to Apple's subscription rules.
- You can embed the issue in the app, and then sell the app.
- If you build your own app with the App Studio Framework, you can integrate with your own e-commerce system for subscriptions (subject to approval by Apple).

When your customer purchases an app or an issue, the proceeds for the sale are deposited in your account by Apple.

How do you assign a price to each issue? You do this by creating *products*. A product associates a particular issue type with a particular Apple in-app purchase or a particular Apple subscription. When your customer purchases an issue, the Publishing Portal looks at that issue's issue type, then gets the subscription or in-app purchase for that issue type and sends it to the app.

You create and manage your App Studio apps and contents using the Quark App Studio Publishing Portal (<http://appstudio.quark.com>). With this Web application:

- You can create a title or collection, along with the list of issue types it supports.
- For each issue type in a title or collection, you can create a product that links the issue type to an Apple in-app purchase or subscription.
- For each individual issue, you can specify the date when it should become available and, optionally, the date after which it should no longer be available for sale. (Issues downloaded before the end-of-availability date remain on the customers' devices even after the end-of-availability date, but can no longer be downloaded.)
- You can create an app specification, which associates a particular App Studio app with particular products.



The relationships you can set up on the Quark App Studio Publishing Portal

When your customer launches your App Studio app, the app asks the Publishing Portal for the products the app is associated with. The Publishing Portal then looks at the issue types associated with each of those products, and returns information about all issues that match that issue type, including their metadata and pricing info. Your customer can browse information about each individual issue or subscription, including preview pictures, headlines, a description, and a graphic summary. When the customer decides to buy the issue, he or she simply taps a Buy button, makes the purchase through the Apple App Store, and then downloads and reads the issue.

Understanding App Studio Factory

App Studio Factory is a Mac OS desktop application that you can use to create App Studio apps. No programming is required; you simply choose an app template from a selection of app templates, then customize that app template to make it your own.

An *app template* lets you create an App Studio app with a particular set of abilities. There are several types of app templates, including the following:

- With an *embedded issue template*, you can create an app that includes a single issue embedded right in the app. Customers who buy this app get the issue automatically, and cannot buy other issues. Embedded issue templates are useful when you want to sell a single publication as an app.
- With a *kiosk template*, your customers can browse a selection of issues that are associated with a particular tile or collection (or, with some kiosk templates, a particular set of titles

UNDERSTANDING APP STUDIO

or collections). When they find one they want, they can purchase it using Apple's in-app purchase or subscription feature. Kiosk templates are best suited for magazine titles.

- With a *bookstore template*, a customer can browse the issues available in one or more collections. Bookstore templates are best suited for book collections.

When you launch App Studio Factory, you can select the app template you want. Once you have selected an app template, you can customize it in a variety of ways, inserting your own text and graphics with the App Studio Factory user interface. When you're done, you can test the app in the iOS Simulator on a Mac OS computer by simply clicking the **Export to Simulator** button.

If you decide you're happy with the app template you've chosen, and you want to submit the app you've built with it to the Apple app store, you must purchase a license for the type of app template you've used at the Quark store (<http://shop.quark.com>) or through a Quark authorized reseller. (For more information, see "[Purchasing an App Studio app template license](#).")

Understanding guided reading

Guided reading mode is a feature available in layout view. Guided reading mode lets end users easily read an article from beginning to end in layout view.

To enter guided reading mode while viewing an issue on the iPad, zoom in on the article until it fills the screen. When the **Guided Reading** button displays, tap it to enter guided reading mode. In this mode, the portion of an article you're reading automatically scales to fill the screen, and you can navigate forward and backward in the article by tapping the left and right edges of the screen, respectively.

In Print layouts, App Studio automatically enables guided reading mode for every reflow article you create. The order of the various parts of a story is determined by the order of the components in that story's reflow article. For more information about reflow articles, see "Creating reflow articles" in *A Guide to QuarkXPress*.

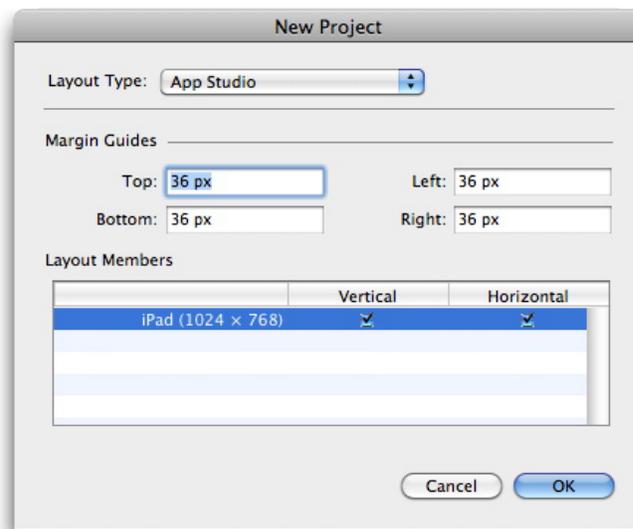
- ➔ Guided reading works only in AVE-Doc issues. You can create AVE-Doc issues only from Print layouts.

Creating an App Studio issue

The topics below explain how to create an App Studio issue and add interactivity to it.

Creating an App Studio issue for AVE-Mag

You can export an AVE-Mag issue from a Print layout, but to take full advantage of the features available for this format, you must create an App Studio layout. To create an App Studio layout for AVE-Mag export, choose **File > New Project** and select **App Studio** from the **Layout Type** drop-down menu.



New Project dialog box for an App Studio layout

To position the margin guides, enter values in the **Margin Guides** fields.

- ➔ The default top margin value provides enough space for the AVE reader app's navigation bar to display.

To specify which devices and orientations you want to support, check **Vertical** and/or **Horizontal** for those devices in the **Layout Members** list.

CREATING AN APP STUDIO ISSUE

When you click **OK**, QuarkXPress creates a project with a layout family that contains layouts for each device/orientation combination you checked. For more information, see "[Understanding layout families](#)."

QuarkXPress displays the new project in a split view, so that you can work with multiple members of the layout family at the same time. For more information, see "Splitting a window" in *A Guide to QuarkXPress*.

- ➔ You cannot export an App Studio layout in AVE-Doc format.
- ➔ By default, all new App Studio layouts include an RGB version of black named **Black (RGB)**. You should use this version of black instead of the default CMYK **Black** color to achieve the darkest black in the exported App Studio issue, especially for text.

Creating an App Studio issue for AVE-Doc

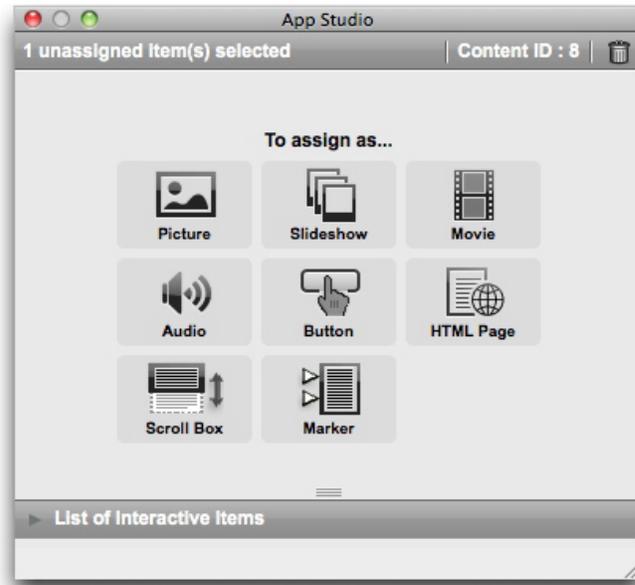
If you plan on exporting your issue in AVE-Doc format, you must create your issue as a Print layout. (For information about creating a Print layout, see "Working with Projects" in *A Guide to QuarkXPress*.)

If you are creating a new layout from scratch, you should create the print layout at the size of your target device. To view the sizes of some common devices, choose **File > New Project**, select **App Studio** from the **Layout Type** drop-down menu, and then look at the **Layout Members** list. Once you have the size of your device, choose **Print** from the **Layout Type** drop-down menu and specify the layout's size in pixels (for example, **1028px** by **768px**).

If you are converting an existing Print layout for use as an issue, you can leave it at its existing size. When you export the layout, you can specify how the issue will display.

Adding interactivity to an App Studio issue

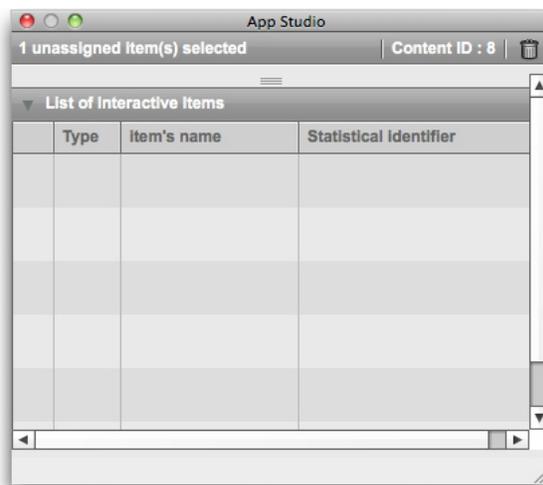
You can add a variety of types of interactivity to an App Studio issue, including slideshows, movies, buttons, sound, and HTML with the **App Studio** palette (**Window** menu).



App Studio palette (top area)

This palette works a little bit differently from other palettes. Before you can interact with this palette, you must activate it by clicking it. So, for example, to click a button in this palette, you would click the palette once to activate it, then click the button.

This palette includes two areas. The top area lets you work with features. The bottom area, **List of Interactive Items**, lets you view a list of the items in the current layout that have mobile app interactivity applied to them. If you click an interactive item in this list, it becomes selected and the active view scrolls to it so you can edit it.



App Studio palette with List of Interactive Items area expanded

Once you apply interactivity to a picture box, the application adds an icon to the box to show what kind of interactivity it has. To view these icons, make sure **View > Visual Indicators** is checked. The icons are as follows:

- : Picture
- : Slideshow
- : Movie
- : Audio
- : Button
- : HTML page
- : Scrollable layout
- : Marker

The **App Studio** palette includes a number of file selection controls. If you are logged into Quark Publishing System (QPS), you can choose whether to select a file from the file system or from QPS. If you choose **QPS** from such a button, the **Asset Picker** dialog box lets you choose a QPS asset.

- ➔ Quark Publishing System is a dynamic publishing platform that provides a broad range of functionality to streamline and automate your publishing process. For information about using App Studio with QPS, see *A Guide to QPS*. For general information about QPS, visit <http://www.quark.com>.

Adding an interactive picture to an App Studio issue

An interactive picture initially displays in a box, but takes over the entire screen when double-tapped. App Studio supports interactive pictures in PNG, JPEG, GIF, TIFF, and EPS formats. To add an interactive picture to an App Studio issue:

- 1 Display the **App Studio** palette (**Window** menu).
- 2 Select the box you want to contain the picture.
- 3 In the **App Studio** palette, click **Picture**.
- 4 Enter an internal name for the picture in the **Item's Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette. (This step is optional.)
- 5 Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior. (This step is optional.)
- 6 To display the picture in full-screen mode, check **Enable fullscreen**. When a user views the picture, he or she can dismiss it by tapping it.
- 7 To display a background when the aspect ratio of the picture does not match the aspect ratio of the display area, choose a color with the **Cell Background Color** control.
- 8 To change the opacity of the background color, specify a percentage with the **Alpha** control. (100% = 100% opaque.)

9 To specify the picture to display, click the **Picture** tab, then click **File** and choose a picture file. Use the cropping rectangle and slider to determine how the picture displays in the box.

10 To add pan-and-zoom effects to the picture, click the **Pan & Zoom Effect** tab.

- Check **Animate Pan & Zoom Effect** to turn this feature on.
- Use the **Delay** field to specify the delay before panning and zooming begins.
- Use the **Duration** field to specify the duration of the panning and zooming.
- Use the **Amplitude** drop-down menu to specify a maximum variation (as a percentage) of the original picture size.

11 To trigger an action when the user taps the picture, click the **Actions** tab. (For more information, see "[Working with interactivity actions](#).")

➔ You can add multiple actions to one picture.

12 To display a caption for the picture, click the **Caption** tab and enter a caption in the field. The caption displays at the bottom of the picture.

- Check **Show Caption** to display the caption on the box.
- Check **Show Caption in Fullscreen Mode** to display the caption in fullscreen mode.

13 To finish configuring the picture, click a different object or a blank part of the layout.

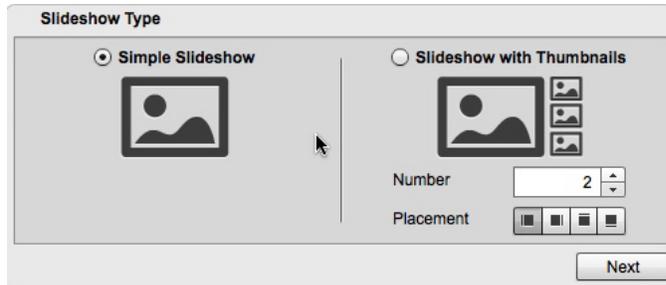
➔ If you import a static picture into the picture box using **File > Import**, the static picture will be obscured by the interactive picture in full-page view. Only the static picture will display in thumbnail views of the page.

Adding a picture slideshow to an App Studio issue

App Studio supports picture files in PNG, JPEG, TIFF, PDF, and EPS formats. To add a slideshow of picture files to an App Studio issue:

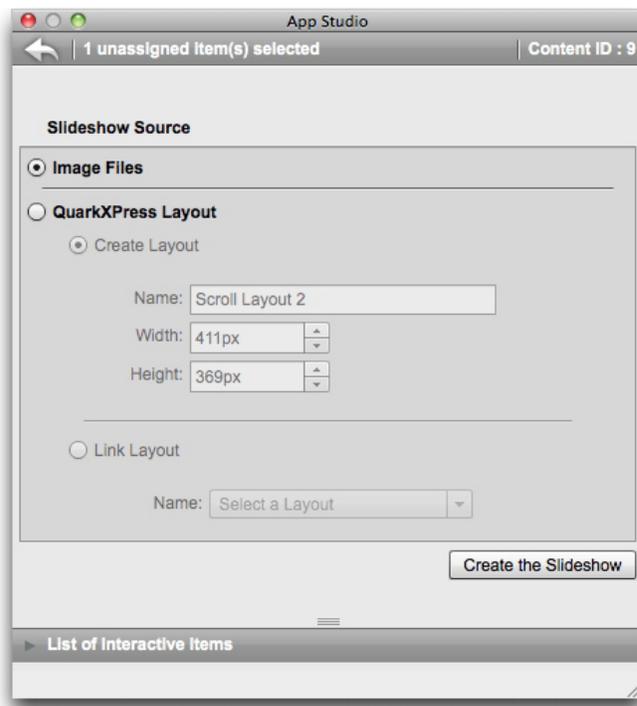
- 1** Display the **App Studio** palette (**Window** menu).
- 2** Select the box you want to contain the slideshow.
- 3** In the **App Studio** palette, click **Slideshow**. The **Slideshow Type** controls display in the palette.

CREATING AN APP STUDIO ISSUE



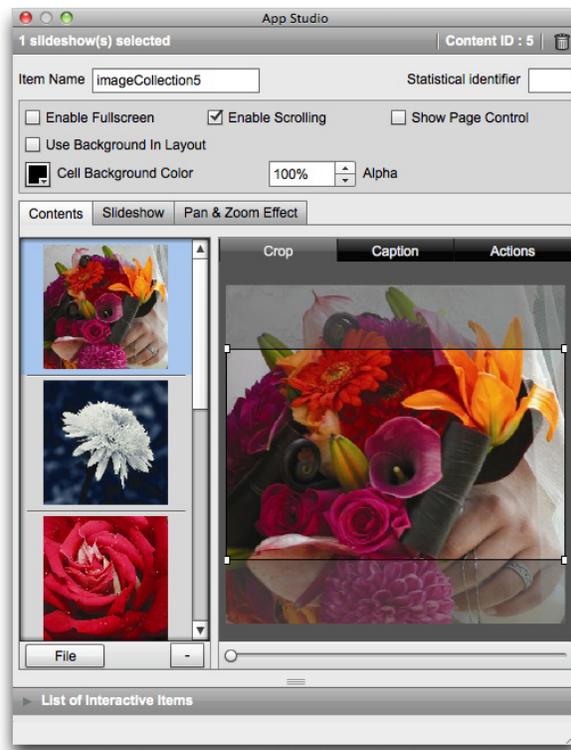
Slideshow options

- 4 To create a slide show without thumbnail previews, click **Simple Slideshow**. To create a slideshow with thumbnail previews, click **Slideshow With Thumbnails**, then specify the number and placement of the thumbnail boxes. You can move the thumbnail boxes if you want.
- 5 Click **Next**. The **Slideshow Source** controls display in the **App Studio** palette.



Slideshow Source controls

- 6 Click **Image Files**, then click **Create the Slideshow**. The **Contents** tab of the next screen displays.



Contents tab

- 7 Enter a name for the slideshow in the **Item Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette.
 - 8 Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior.
 - 9 To display the slideshow in full-screen mode, check **Enable fullscreen**. When a user taps the slideshow twice, it displays in full-screen mode. The user can then dismiss it by tapping it.
 - 10 To allow the user to scroll through the pictures in the slideshow by swiping, check **Enable scrolling**. If you do not check this option, you must set up a **Change Content Index** action to switch between the pictures in the slide show. For more information, see "[Working with interactivity actions](#)."
 - 11 To show a page control when the slideshow displays, check **Show Page Control**.
 - 12 To display a solid color in areas of the slideshow that are not occupied by a picture, uncheck **Use Background In Layout**, then use the **Cell Background Control** to specify a color and the **Alpha** control to set the color's transparency.
- ➔ The **Opacity** value applies only to the background. The slide images always display 100% opaque.

13 To add a picture to the slideshow, choose **File** (for a file in the file system) or **QPS** (for a file in QPS) from the button at the bottom of the list on the left, then choose a file. You can change the order of pictures in a slideshow by dragging and dropping them.

➔ To delete a picture, select it and click the minus button (-).

14 To set the thumbnail crop for a slide, select it in the list on the left and click the **Crop** tab. Use the slider at the bottom to resize the image, then drag out a cropping rectangle from upper-left to lower-right.

15 To add a caption for a slide, select it in the list on the left and click the **Caption** tab. Enter a caption in the field that displays in that tab.

- Check **Show Caption** to display the caption on the box.
- Check **Show Caption in Fullscreen Mode** to display the caption in fullscreen mode.

16 To trigger an action when the end user swipes or taps on a slide, select the slide in the list on the left and click the **Actions** tab.

- To add an action when a user swipes on a slide, choose an action from the **Add Scroll Actions** drop-down menu, then click **Create**.
- To add an action when a user taps on a slide, choose an action from the **Add Tap Actions** drop-down menu, then click **Create**.

For more information, see "[Working with interactivity actions](#)."

➔ You can add multiple actions for each gesture to the same slide.

17 To control the playback of the slideshow, click the **Slideshow** tab.

- To enable animation, check **Enable Animation**.
- Use the **Transition Duration (ms)** field to specify the transition duration in seconds.
- Use the **Image Duration (ms)** field to specify the image display duration in seconds.
- Use the **Effect** drop-down menu to specify the type of transition.

18 To add pan-and-zoom effects to the slideshow, click the **Pan & Zoom Effect** tab.

- Check **Animate Pan & Zoom Effect** to turn this feature on.
- Use the **Delay (ms)** field to specify the delay before panning and zooming begins.
- Use the **Duration (ms)** field to specify the duration of the panning and zooming.
- Use the **Amplitude** drop-down menu to specify a maximum variation (as a percentage) of the original picture size.

19 To finish configuring the slideshow, click a different object or a blank part of the layout.

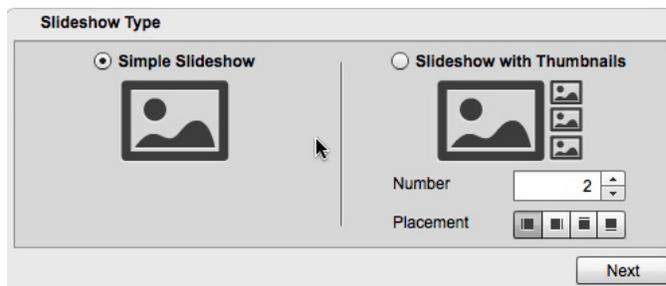
Adding a layout slideshow to an App Studio issue

A layout slideshow is a slideshow that displays the pages of a different layout in the project. This allows you to design each slide of a slideshow in QuarkXPress.

- ➔ App Studio interactivity in a layout slideshow does not work when the slideshow preview displays, but *does* work when the user views the slideshow in full-screen mode.

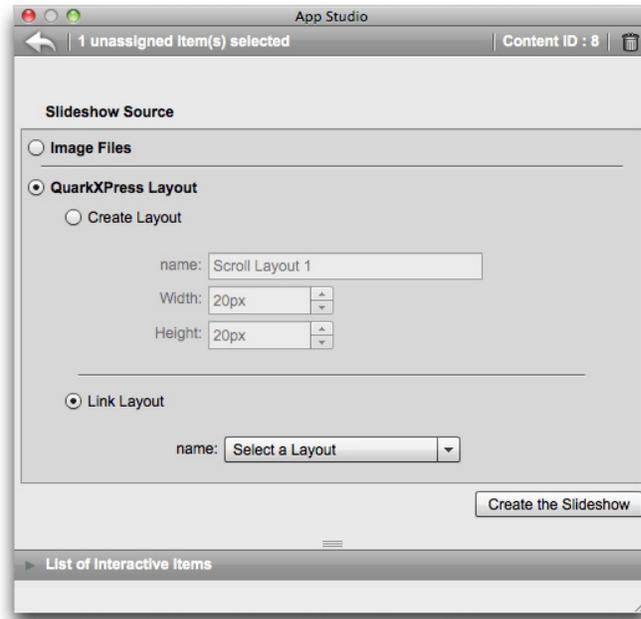
To add a layout slideshow to an App Studio issue:

- 1 Display the **App Studio** palette (**Window** menu).
- 2 Select the box you want to contain the slideshow.
- 3 In the **App Studio** palette, click **Slideshow**. The **Slideshow Type** controls display in the palette.



Slideshow options

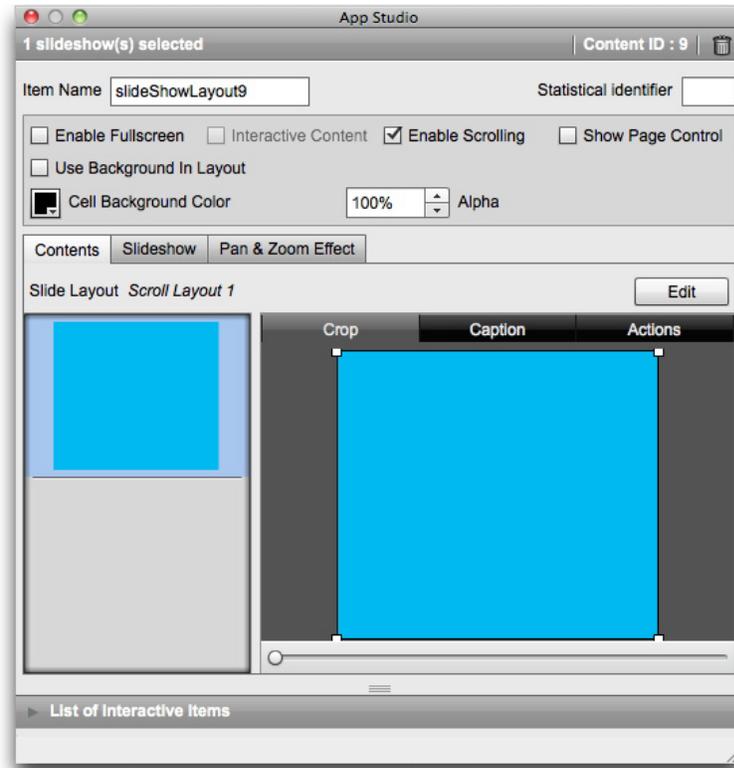
- 4 To create a slide show without thumbnail previews, click **Simple Slideshow**. To create a slideshow with thumbnail previews, click **Slideshow With Thumbnails**, then specify the number and placement of the thumbnail boxes. You can move the thumbnail boxes if you want.
- ➔ If you create a layout to go with the slideshow, the application will automatically create the number of pages you specify in the **Number** field.
- 5 Click **Next**. The **Slideshow Source** controls display in the **App Studio** palette.



Slideshow Source controls

- 6 Click **QuarkXPress Layout**.
- 7 If you want to create a new layout to serve as the layout slideshow, click **Create Layout**, and then do the following:
 - Enter a name for the layout in the **Name** field.
 - Enter a width and height in the **Width** and **Height** fields. If you want the slideshow to fill the screen, the aspect ratio of the layout should match the aspect ratio of the target device. By default, a layout created in this way is the same size as the main layout, so that the layout slideshow is suitable for viewing in full-screen mode.

If you want to use an existing layout in the active project as the scrollable layout, click **Link Layout** and choose that layout's name from the drop-down menu. (At display, the layout is scaled to fit in the box.)
- 8 Click **Create the Slideshow**. The next screen displays.



Contents screen

- 9 Enter an internal name for the picture in the **Item Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette. (This step is optional.)
- 10 Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior. (This step is optional.)
- 11 To display the slideshow in full-screen mode, check **Enable Fullscreen**. When a user views the slideshow, he or she can dismiss it by tapping it.
- 12 To enable App Studio interactivity in the layout, check **Interactive Content**. If you don't check this box, App Studio interactivity will not work in the layout slideshow, but the slides may display more quickly.
- 13 To allow the user to scroll through the pictures in the slideshow by swiping, check **Enable Scrolling**. If you do not check this option, you must set up a **Change Content Index** action to switch between the pictures in the slide show. For more information, see "[Working with interactivity actions](#)."
- 14 To show a page control when the slideshow displays, check **Show Page Control**.
- 15 To display a solid color in areas of the slideshow that are not occupied by a picture, uncheck **Use Background In Layout**, then use the **Cell Background Control** to specify a color and the **Alpha** control to set the color's transparency.

➔ The **Opacity** value applies only to the background. The slide images always display 100% opaque.

16 To edit the layout used in the slideshow, click **Edit**.

17 To set the thumbnail crop for a slide, select it in the list on the left and click the **Crop** tab. Use the slider at the bottom to resize the picture, then drag out a cropping rectangle from upper-left to lower-right. You can also crop the thumbnail by changing the cropping of the picture in the picture box.

➔ Do not rotate the picture in the picture box or change its skew. These changes will not display in the final issue.

18 To add a caption for a slide, select it in the list on the left and click the **Caption** tab. Enter a caption in the field that displays in that tab.

- Check **Show Caption** to display the caption on the box.
- Check **Show Caption in Fullscreen Mode** to display the caption in fullscreen mode.

19 To trigger an action when a slide changes, select it in the list on the left and click the **Actions** tab. Choose an action from the **Add Action** drop-down menu at the bottom, then click **Create**.

- To add an action when a user swipes on a slide, choose an action from the **Add Scroll Actions** drop-down menu, then click **Create**.
- To add an action when a user taps on a slide, choose an action from the **Add Tap Actions** drop-down menu, then click **Create**.

For more information, see "[Working with interactivity actions](#)."

➔ You can add multiple actions for each gesture to the same slide.

20 To control the playback of the slideshow, click the **Slideshow** tab.

- To enable animation, check **Enable Animation**.
- Use the **Transition Duration (ms)** field to specify the transition duration in seconds.
- Use the **Image Duration (ms)** field to specify the image display duration in seconds.
- Use the **Effect** drop-down menu to specify the type of transition.

21 To add pan-and-zoom effects to the slideshow, click the **Pan & Zoom Effect** tab.

- Check **Animate Pan & Zoom Effect** to turn this feature on.
- Use the **Delay (ms)** field to specify the delay before panning and zooming begins.
- Use the **Duration (ms)** field to specify the duration of the panning and zooming.

- Use the **Amplitude** drop-down menu to specify a maximum variation (as a percentage) of the original picture size.

22 To finish configuring the slideshow, click a different object or a blank part of the layout.

- ➔ You can resize the box that contains a layout slideshow.
- ➔ To preview a different slide, select the picture box, choose **Item > Modify**, click the **Layout** tab, and change the value of the **Page** drop-down menu. (In an exported issue, the slideshow always starts with the first slide regardless of this setting.)

Adding a movie to an App Studio issue

App Studio supports only movie files that can be played on the target device. Currently, this includes the following:

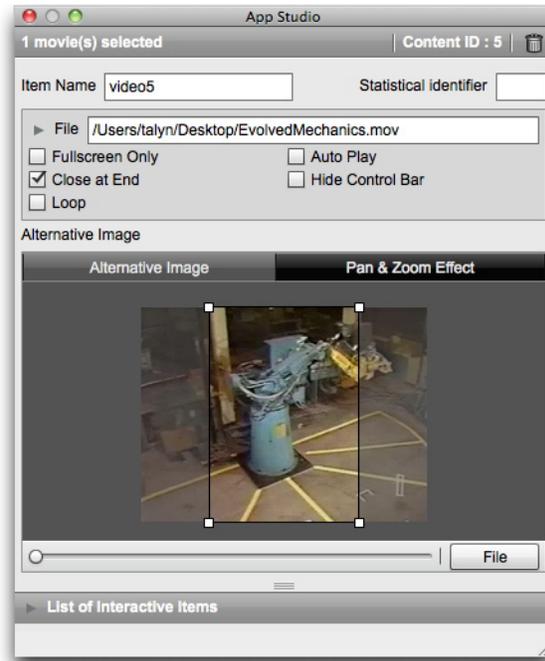
- H.264 video at up to 720p and 30 frames per second, Main Profile level 3.1, with AAC-LC stereo audio up to 160 Kbps/48kHz, in .m4v, .mp4, or .mov file format
- MPEG-4 video at up to 2.5 Mbps/640 x 480 pixels/30 frames per second, simple profile, with AAC-LC stereo audio up to 160 Kbps per channel/48kHz, in .m4v, .mp4, or .mov file format

- ➔ For more information, see <http://www.apple.com/ipad/specs/>.
- ➔ To verify that a movie file will work in an app studio issue, view the video in Safari on a physical iPad. The iOS Simulator edition of Safari can play movie files that a physical iPad cannot.
- ➔ You can start and stop the playback of video files with actions. For more information, see "[Creating Play, Pause, Stop, and Toggle Play actions.](#)"

To add a movie to an App Studio issue:

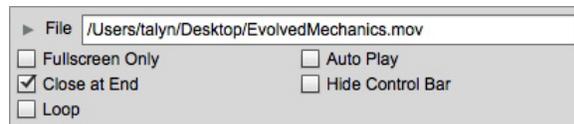
- 1** Display the **App Studio** palette (**Window** menu).
- 2** Select the box you want to contain the movie.
- 3** In the **App Studio** palette, click **Movie**.

CREATING AN APP STUDIO ISSUE



Movie options

- 4 Enter an internal name for the picture in the **Item's Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette. (This step is optional.)
- 5 Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior. (This step is optional.)
- 6 To specify the location of the movie, expand the **File** area. To specify the location of a movie on the Web, click **URL** and then enter the movie URL in the **URL** field. (Note that the URL must point directly to the movie file; for example, a YouTube page URL will not work.) To specify a movie on the local file system, click **File** and then click **Define** and navigate to the movie file.



Expanded **File** area

- 7 To make the movie always play in full-screen mode, check **Fullscreen Only**.
- 8 To make the movie play automatically when the page it is on displays, check **Auto Play**.
- 9 To make the movie overlay close when the movie is done playing, check **Close at End**.
- 10 To hide the movie controls, click **Hide Control Bar**.
- 11 To make the movie play repeatedly, click **Loop**.

12 To specify an image to display when the movie is not playing, click the **Alternative Image** tab, then choose **File** (for a file in the file system) or **QPS** (for a file in QPS) from the button at the bottom, then choose a file. Use the slider at the bottom to resize the image, then drag out a cropping rectangle from upper-left to lower-right.

13 To pan and zoom the alternate image, click the **Pan & Zoom Effect** tab.

- Check **Animate Pan & Zoom Effect** to turn this feature on.
- Use the **Delay** field to specify the delay before panning and zooming begins.
- Use the **Duration** field to specify the duration of the panning and zooming.
- Use the **Amplitude** drop-down menu to specify a maximum variation (as a percentage) of the original picture size.

14 To finish configuring the movie, click a different object or a blank part of the layout.

Adding audio to an App Studio issue

You can associate an audio file with a picture box. When the end user views the issue, the box is replaced with audio controls that allow the sound to be played.

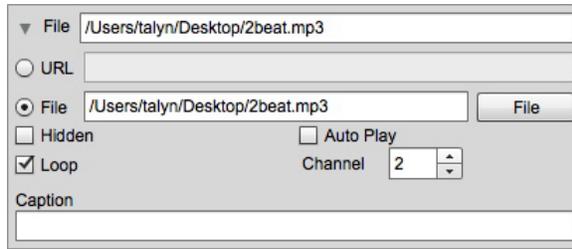
➔ You can start and stop the playback of audio files with actions. For more information, see "[Creating Play, Pause, Stop, and Toggle Play actions.](#)"

➔ App Studio supports only audio files that can be played on the target device. For more information, see <http://www.apple.com/ipad/specs/>.

To add audio to an App Studio issue:

- 1** Display the **App Studio** palette (**Window** menu).
- 2** Select the box you want to replace with the audio controls.
- 3** In the **App Studio** palette, click **Audio**.
- 4** Enter an internal name for the picture in the **Item's Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette. (This step is optional.)
- 5** Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior. (This step is optional.)
- 6** To specify the location of the audio file, expand the **File** area. To specify the location of an audio file on the Web, click **URL** and then enter the URL of the audio file in the **URL** field. To specify an audio file on the local file system, click **File** and then click the **File** button and navigate to the audio file.

CREATING AN APP STUDIO ISSUE



Expanded **File** area

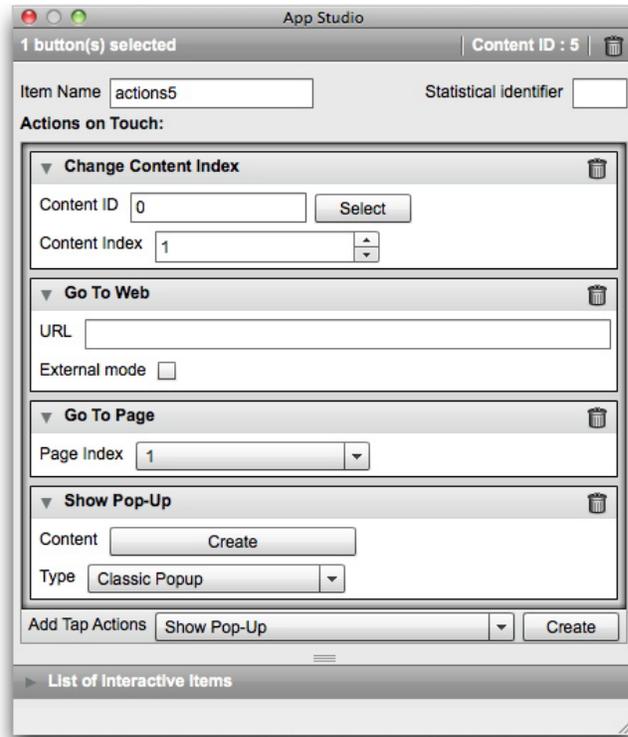
- 7** To hide the audio controls, click **Hidden**.
- 8** To make the audio play automatically when the user displays the page the picture box is on, check **Auto Play**.
- 9** To make the audio play repeatedly, click **Loop**.
- 10** To set the channel of the audio, enter a number in the **Channel** field. If you want a sound to play at the same time as another sound, the two sounds must be in different channels. If a sound in one channel is playing, and a second sound in the same channel starts playing, the first sound stops.
- 11** To display a caption at the bottom of the audio box, enter the caption in the **Caption** field.
- 12** You can specify an alternate image to display in the picture box if there is no Internet connection on the user's device. To specify a file, click the **Alternative Image** tab, then choose **File** (for a file in the file system) or **QPS** (for a file in QPS) from the button at the bottom, then choose a file. Use the slider at the bottom to resize the image, then drag out a cropping rectangle from upper-left to lower-right.
- 13** To pan and zoom the picture, click the **Pan & Zoom Effect** tab.
 - Check **Animate Pan & Zoom Effect** to turn this feature on.
 - Use the **Delay** field to specify the delay before panning and zooming begins.
 - Use the **Duration** field to specify the duration of the panning and zooming.
 - Use the **Amplitude** drop-down menu to specify a maximum variation (as a percentage) of the original picture size.
- 14** To finish configuring the audio element, click a different object or a blank part of the layout.

Adding a button to an App Studio issue

To add a button to an App Studio issue:

- 1** Display the **App Studio** palette (**Window** menu).
- 2** Select the picture box you want to make into a button.
- 3** In the **App Studio** palette, click **Button**.

- 4 Enter an internal name for the picture in the **Item's Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette. (This step is optional.)
- 5 Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior. (This step is optional.)
- 6 To add an action for the button, use the **Actions on Touch** area. Choose an action from the **Add Action** drop-down menu at the bottom, then click **Create**. You can add multiple actions to the same button. For more information, see "[Working with interactivity actions.](#)"



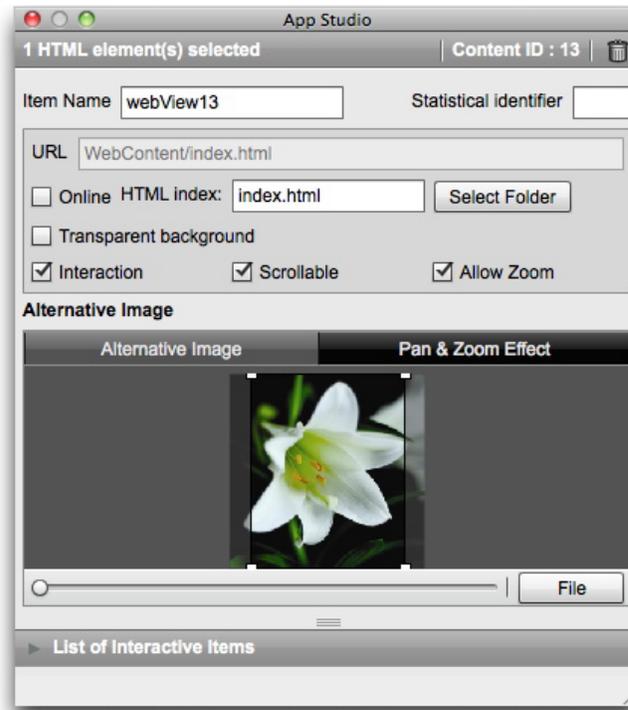
Actions added to a button

- 7 To finish configuring the button, click a different object or a blank part of the layout.

Adding an HTML view to an App Studio issue

You can use an HTML view (a Web overlay) to include changeable content (such as ads) or custom interactivity to an App Studio issue. You can use an HTML view to display HTML, PDF, Word, Excel®, PowerPoint®, RTF, and several other types of files. On the output device, the WebKit framework is used to render the content. To add an HTML view to an App Studio issue:

- 1 Display the **App Studio** palette (**Window** menu).
- 2 Select the box you want to contain the HTML content.
- 3 In the **App Studio** palette, click **HTML Page**.



HTML options

- 4 Enter an internal name for the picture in the **Item's Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette. (This step is optional.)
- 5 Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior. (This step is optional.)
- 6 To specify the location of the content, enter its URL in the **URL** field. You can do this in the following ways:
 - Check **Online** and enter the URL of a Web page (for example, <http://www.quark.com>). The URL must begin with a protocol such as <http://>.
 - Uncheck **Online**, click **Select Folder**, select the folder that contains the target content, then enter the name of the target file in that folder (for example, [index.html](#) or [embedded.pdf](#)). All of the contents of that folder will be embedded in the issue.
- 7 To display the content with a transparent background, check **Transparent background**.
- 8 To allow the user to interact with the content, check **Interaction**.
- 9 To make the content scrollable, check **Scrollable**. You should uncheck this control if the content fits entirely in the box, because otherwise the content will display with scroll controls even if it is not scrollable.
- 10 To allow the user to zoom in on the content, check **Allow Zoom**.

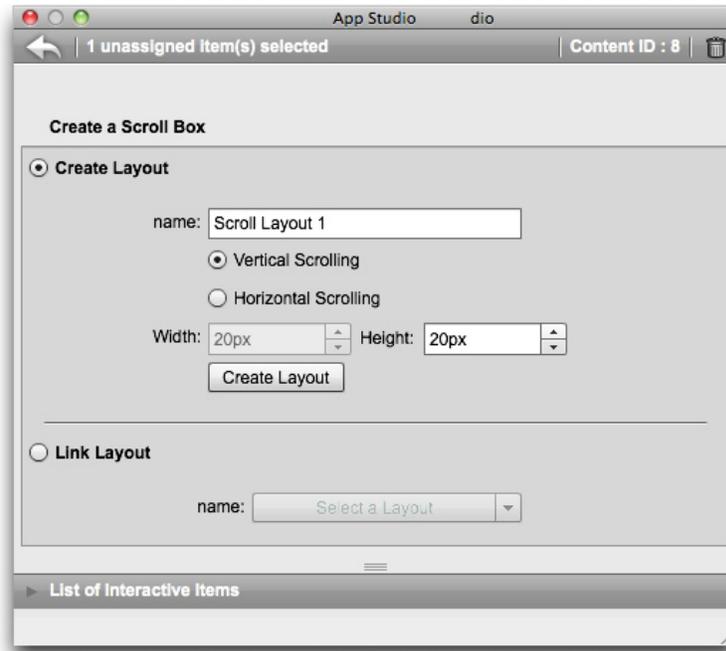
- 11** To specify an image to display when the content is not available, click the **Alternative Image** tab, then click **File** and choose a file. Use the slider at the bottom to resize the image, then drag out a cropping rectangle from upper-left to lower-right.
- 12** To add pan-and-zoom effects to the content, click the **Pan & Zoom Effect** tab.
- Check **Animate Pan & Zoom Effect** to turn this feature on.
 - Use the **Delay (ms)** field to specify the delay before panning and zooming begins.
 - Use the **Duration (ms)** field to specify the duration of the panning and zooming.
 - Use the **Amplitude (ms)** drop-down menu to specify a maximum variation (as a percentage) of the original size.
- 13** To finish configuring the HTML view, click a different object or a blank part of the layout.

Adding a scrollable layout to an App Studio issue

The scrollable layout feature lets you add a scrollable area to a page of an App Studio issue. The content for the scrollable area comes from a different layout (the scrollable layout). Once you set up a scrollable layout, you can populate it with whatever you want, whether it's a long run of text, a large panoramic picture, or a series of interactive elements. You can then use that scrollable layout in multiple layouts within a layout family. (For more information, see "[Understanding layout families](#).")

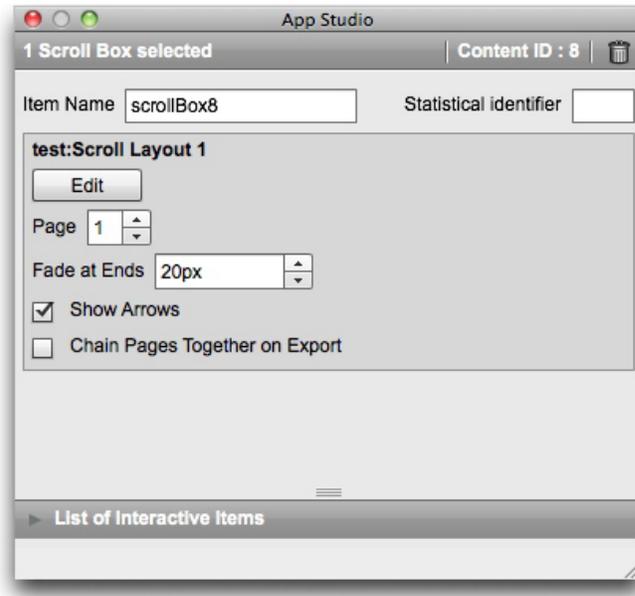
To set up a scrollable layout:

- 1** Navigate to a layout that is part of a layout family.
- 2** Draw a picture box to represent the size and the location of the scrollable area. Make sure the box is selected.
- 3** Display the **App Studio** palette (**Window** menu).
- 4** In the **App Studio** palette, click **Scroll Box**. The scrollable layout controls display in the palette.



Scrollable layout controls

- 5 If you want to create a new layout to serve as the scrollable layout, click **Create Layout**, and then do the following:
 - Enter a name for the layout in the **Name** field.
 - To specify the scrolling direction, click **Vertical Scrolling** or **Horizontal Scrolling**.
 - To specify the width or height of the scrollable layout, enter a value in the **Width** or **Height** field. (You can specify this value only in the direction that the layout will scroll.)
 - Click **Create Layout**. A new layout is added to the active project and linked to the selected box.
- 6 In some cases, you might want to link a second box to a same scrollable layout. If you want to use an existing layout in the active project as the scrollable layout, click **Link Layout** and choose that layout's name from the drop-down menu. The scrolling direction is determined by the orientation of the layout; if it is wider than it is tall it scrolls horizontally, and if it is taller than it is wide it scrolls vertically. The size of the selected box is adjusted to fit the layout.
- 7 Once you have linked the scrollable layout to the box, the **App Studio** palette displays the following controls.



Scrollable layout controls

- To edit the scrollable layout, click **Edit**.
- To change the page that displays in the scroll box by default, change the value in the **Page** field.
- To make the ends of the scroll box fade into the background on the device, check **Fade at Ends**.
- To provide scrollbars as a visual cue that the area is scrollable, check **Show Arrows**.
- If the scrollable layout includes multiple pages, and you want the pages to be stitched together into one long scrollable element, check **Chain Pages Together on Export**.

Once you have linked the scrollable layout to the scroll box, the box's size will be locked in the non-scrolling direction. If you want to change the size of the box in this direction, you must resize the layout associated with the box. You can navigate to that layout at any time by Control+clicking/right-clicking the box and choosing **Composition Zones > Edit**. You can also choose **Item > Composition Zones > Edit**.

If you want to view different portions of a scrollable layout in its box, double-click the box with the **Picture Content** tool  and then click and drag the item content in the box.

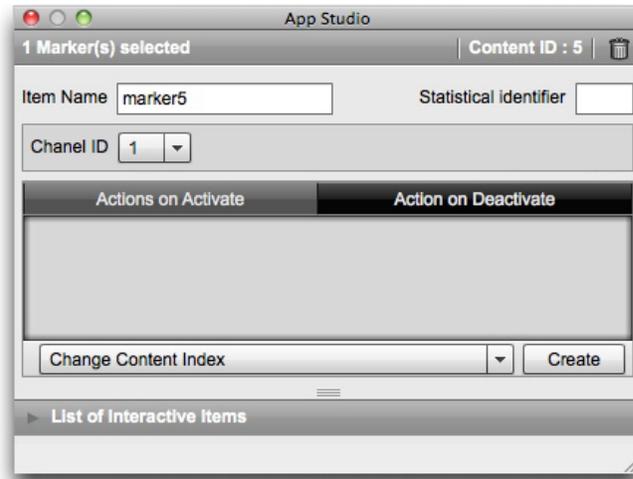
Adding a marker to a scrollable layout

A scrollable layout can include markers, which are boxes that trigger actions when they scroll into and out of view. For example, you can create a slideshow, then use markers in a scrollable layout to change the slide as different parts of the story in the scrollable layout appear.

CREATING AN APP STUDIO ISSUE

To add markers to a scrollable layout:

- 1 Create a picture slideshow or layout slideshow.
 - 2 Create a scrollable layout on the same page.
 - 3 To edit the scrollable layout, select it and then choose **Item > Composition Zones > Edit**. The scrollable layout displays in a new window.
 - 4 Draw a box for each slide you want to display, and position it near the point in the layout where you want that slide to display. For example, if you want the slideshow to display a picture of a person mentioned in an article, position the box near the place in the article where the person is mentioned.
- ➔ You can have multiple channels in a single scrollable layout, each channel consisting of a separate series of marker boxes. The marker boxes in one channel might control a movie, while the marker boxes in another channel might control which slide displays. If you are using multiple channels, you might consider making all of the boxes in a channel use the same background color, so that you can easily tell them apart.
- 5 Display the **App Studio** palette (**Window** menu).
 - 6 Select a box to be made into a marker.
 - 7 In the **App Studio** palette, click **Marker**. The **Marker** controls display in the palette.



Marker options

- 8 Enter an internal name for the marker in the **Item's Name** field. This name displays in the **List of Interactive Items** area in the bottom of the **App Studio** palette.
- 9 Enter a unique value in the **Statistical identifier** field. In the future, this value may be used to track user behavior.

- 10** If the marker is part of a channel, choose a channel identifier from the **Channel ID** drop-down menu.
- 11** To specify what will happen when the marker item scrolls onto the screen or becomes more prominent on the screen than any other markers in the same channel, click the **Actions on Active** tab, then add the appropriate action.
- To change the image displayed in a slideshow, choose **Change Content Index**. For more information, see "[Creating a Change Content Index action](#)."
- 12** To specify what will happen when the marker item scrolls off of the screen or becomes less prominent on the screen than any other markers in the same channel, click the **Actions on Deactive** tab, then add the appropriate action.
- To change the image displayed in a slideshow, choose **Change Content Index**. For more information, see "[Creating a Change Content Index action](#)."
- 13** To finish configuring the picture, click a different object or a blank part of the layout.

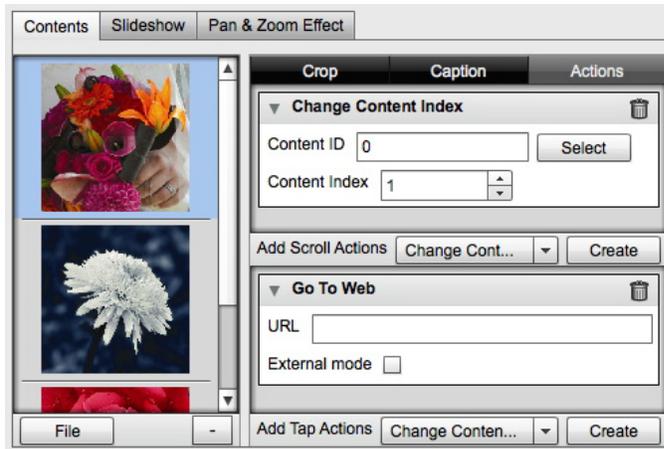
Removing interactivity from an item

To remove interactivity from an item, select the item, then click **Remove Item's Interactivity**  in the **App Studio** palette. The item is not deleted.

Working with interactivity actions

Actions let you add interactivity to items in a layout that you will be exporting as an App Studio app. The actions available include:

- **Change Content Index:** Lets you change the image displayed in a slideshow.
- **Go To Web:** Navigates to a URL when the user touches the selected item. The URL is displayed in an overlay.
- **Go To Page:** Jumps to the page number you indicate when the user touches the selected item.
- **Show Pop-Up:** Lets you create a pop-up window that displays a slideshow, movie, or HTML content.
- **Show Fullscreen:** Lets you make an interactive object take over the screen.

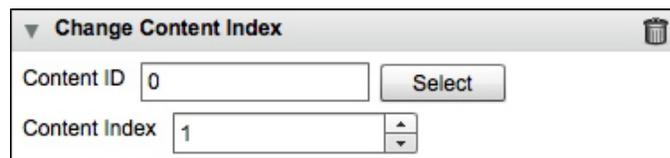


Actions applied to a slideshow

Creating a Change Content Index action

The **Change Content Index** action lets you change the currently displayed picture in a slideshow. To create a **Change Content Index** action:

- 1 Choose **Change Content Index** from the **Add Action** drop-down menu, then click **Create**.



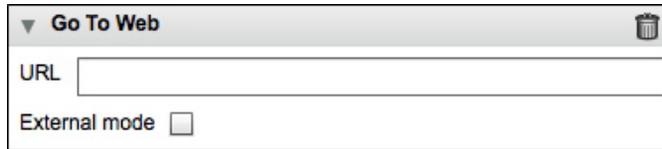
Change Content Index action

- 2 To indicate which slideshow to change, click **Select** and then the target slideshow box in the layout. Alternatively, you can enter the target slideshow box's content ID in the **Content ID** field. (To get the content ID, select the slideshow box and look at the upper right corner of the **App Studio** palette.)
- 3 To indicate which picture to display, specify a value in the **Content Index** field. Note that the first picture in a slideshow is picture 1.

Creating a Go to Web action

The **Go to Web** action displays the contents of a URL in an overlay when you tap the box the action is associated with. To create a **Go to Web** action, first choose **Go to Web** from the **Add Action** drop-down menu, then click **Create**. Then enter the URL for the action in the **URL** field.

- ➔ The URL must begin with a protocol such as `http://`.

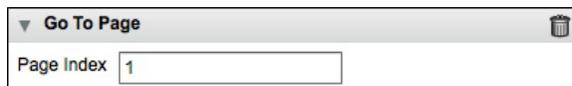


Go to Web action

To make the URL display in the device's default Web browser, check **External mode**.

Creating a Go to Page action

The **Go to Page** action jumps to a different page in an issue. To create a **Go to Page** action, first choose **Go to Page** from the **Add Action** drop-down menu, then click **Create**. Then enter the absolute page number of the target page from the **Page Index** field.



Go to Page action

- ➔ To get the absolute page number of the target page, Option+click/Alt+click the page icon in the **Page Layout** palette (**Window** menu).

If you enter the number of a page that does not yet exist, the **Page Index** field is highlighted in red.

To jump to a page in a different layout, enter the number of the layout followed by a hyphen and then the number of the page. For example, to go to the 9th page in the 3rd layout in a project, enter **3-9**. Using this method, you can create a table of contents in a scrollable layout.

- ➔ If you change the location of a page that is targeted by a **Go to Page** action, you must also change the page number in that **Go to Page** action. For example, if you move page 20 so that it becomes page 18, you must change any **Go to Page** actions that go to page 20 so that they go to page 18 instead.

Creating a pop-up

The **Show Pop-Up** action lets you create a pop-up that displays a slideshow, movie, HTML page, PDF, or other content. To create a pop-up in an App Studio issue:

- 1 Create a **Show Pop-Up** action using the **Add Action** drop-down menu. This drop-down menu is available in the **App Studio** palette anywhere where you can associate an action with something.

CREATING AN APP STUDIO ISSUE



New **Show Pop-Up** action

- 2 Choose an option from the **Type** drop-down menu:
 - **Classic Pop-Up** creates a pop-up window with a border.
 - **Centered Pop-Up** creates a pop-up window in the center of the screen with a shaded background around it.
 - **Fullscreen Pop-Up** creates a pop-up window that fills the entire screen.

The user can tap outside of a pop-up window to dismiss it.

- 3 Click **Create**. The **Create Element** screen displays in the **App Studio** palette.



Create Element screen

- 4 Click **Slideshow** to create a pop-up slideshow, **HTML Page** to create a pop-up HTML view, or **Movie** to create a pop-up movie. For more information, see "[Adding a picture slideshow to an App Studio issue](#)," "[Adding an HTML view to an App Studio issue](#)," and "[Adding a movie to an App Studio issue](#)."
- 5 To specify the size of the pop-up, enter values in pixels in the **Width** and **Height** fields at the top of the palette.

Once you have started working on the pop-up slideshow or HTML page that you're adding, you can return to the action that displayed it by clicking .

Creating a Show Fullscreen action

The **Show Fullscreen** action displays the target item in fullscreen mode.

- ➔ The **Show Fullscreen** action does not work with sound items.

To create a **Show Fullscreen** action, first choose **Show Fullscreen** from the **Add Action** drop-down menu, then click **Create**. To indicate the target object, click **Select** and then

click an interactive item, or enter the content ID of the item in the Content ID field. (An item's content ID displays at the top of the App Studio palette when the item is selected.)



Show Fullscreen action

Creating Play, Pause, Stop, and Toggle Play actions

The **Play**, **Pause**, **Stop**, and **Toggle Play** actions let you create controls that start and stop movies and audio.

➔ These actions work only with movie and sound items.

- The **Play** action starts a movie or sound from the beginning.
- The **Pause** action stops a movie or sound. If you execute a Play action after a Pause action, the movie or sound resumes where it left off.
- The **Stop** action stops a movie or sound. If you execute a Play action after a Stop action, the movie or sound plays from the beginning.
- The **Toggle Play** action pauses a movie or sound if it is playing, and resumes playing the movie or sound if it has been paused.

To create one of these actions, first choose its name from the **Add Action** drop-down menu, then click **Create**. To indicate the target object, click **Select** and then click an interactive item, or enter the content ID of the item in the Content ID field. (An item's content ID displays at the top of the App Studio palette when the item is selected.)



Play, Pause, Stop, and Toggle Play actions

Creating page stacks

The page stack feature lets you organize sections of content into "stacks" of pages. On the iPad, the user can display a horizontal line of thumbnails, each of which represents a section of content (such as an article in a magazine), with a scrollable vertical list of page thumbnails underneath each.



Page stacks

In full-page view, a page-stack design lets the user swipe horizontally to browse through articles and vertically to read them.

To organize an App Studio issue into page stacks, you must divide your layout into sections, with each section representing the first page of a page stack. In addition, you must set the starting number of each section to the absolute page number of its page.

To create a page stack:

- 1 In the **Page Layout** palette, Option+click/Alt+click the page you are on. The absolute page number of the page displays at the bottom of the palette, preceded by a +.
- 2 Choose **Page > Section** and enter the absolute page number from the **Page Layout** palette in the **Number** field.
- 3 Click **OK**.
- 4 Repeat these steps for each page you want to designate as the start page in a page stack.
- 5 When you export the layout, make sure you turn on page stacks in the **AVE Options** dialog box. For more information, see "[Setting App Studio issue export options.](#)"

Using dual orientations

If you plan to publish in AVE-Mag format, you can make your issue display differently in horizontal and vertical orientations. To do this, you must create an **App Studio** layout, which will automatically create two layouts in the same project: one for horizontal viewing, and another for vertical viewing.

To add a layout to a QuarkXPress project, choose **Layout > New**, choose **App Studio** from the **Layout Type** menu, then click **OK**. QuarkXPress automatically creates a layout for horizontal viewing and a layout for vertical viewing.

Synchronizing content between orientations

The commands described in this topic let you synchronize a page's content between the various layouts in a layout family. For example, if you create a table of contents on page 2 of the vertical layout for the iPad, you can use the commands described here to automatically turn that table of contents into shared content and copy it to page 2 of the horizontal layout for the iPad and/or the vertical layout for the Galaxy Tab. You may need to reposition and/or resize the content in the other layouts, but these commands automate the process of getting the content to the appropriate pages in each device and orientation, and they help ensure that that content stays consistent on all of the pages where it appears.

➔ For more information about shared content, see "Working with shared content" in *A Guide to QuarkXPress*. To understand how the commands described below work, you must first understand how shared content works.

Item > Copy to Other Layouts: This command turns all selected boxes into shared content and places copies of them on the same page of other layouts in the layout family. The submenu for this command lets you be more specific about how this copy is performed:

- **Same Position:** Puts the copies in the same position relative to the upper-left corner of the page, if possible. This option copies items to all other layouts in the layout family.
- **Use Relative Positioning:** Puts the copies in the same position relative to the page size and shape overall. This option copies items to all other layouts in the layout family.

Setting the Contents page

If you plan to export an issue in AVE-Mag format, you can specify which page contains the table of contents. When the user views the issue, they can tap once to display a bar at the top with a Content button, then tap the Content button to go to your table of contents page.

To specify which page contains the table of contents, choose **Layout > AVE Contents Page**, enter the absolute page number in the **Contents Page #** field, and then click **OK**.

Reviewing Digital Publishing asset usage

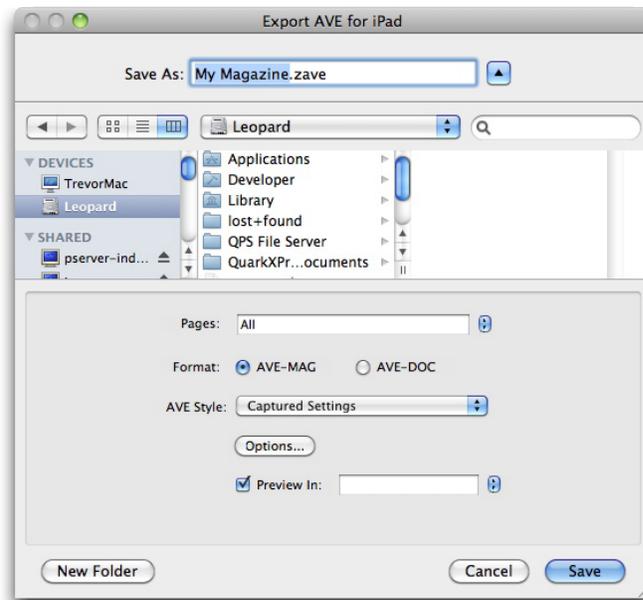
To see which assets have been used in Blio and App Studio interactivity in the active layout, display the **Digital Publishing** pane of the **Usage** dialog box (**Utilities** menu). Like the **Pictures** pane, the **Digital Publishing** pane lists the page number (a dagger symbol or PB indicates pasteboard) for each Digital Publishing asset in the layout. It also includes a **Type** column that indicates the asset type and a **Source** column that indicates whether the asset

CREATING AN APP STUDIO ISSUE

is a file or a folder. To view more information about the selected asset, expand the **More Information** area at the bottom.

Exporting an App Studio issue

To export an App Studio issue, choose **File > Export > Layout as AVE**. The **Export AVE for iPad** dialog box displays.



Export AVE for iPad dialog box

Enter a name for the exported .zave file in the **Save As** field. To avoid problems on the App Studio Publishing Portal, use a name that does not contain spaces and special characters.

Use the **Pages** field to specify the pages to include in the issue. Note that you can enter only a single page or a single page range such as 5–10.

Use the **Format** controls to specify whether you want to export in AVE-Mag or AVE-Doc format. For more information about these options, see "[Understanding App Studio](#)."

The **AVE Style** drop-down menu lets you choose an AVE output style. (For more information, see "Working with output styles" in *A Guide to QuarkXPress*.) To use the most recently used settings for this project file, choose **Captured Settings**.

If you click the **Options** button, the **AVE Options** dialog box displays. For more information, see "[Setting App Studio issue export options](#)."

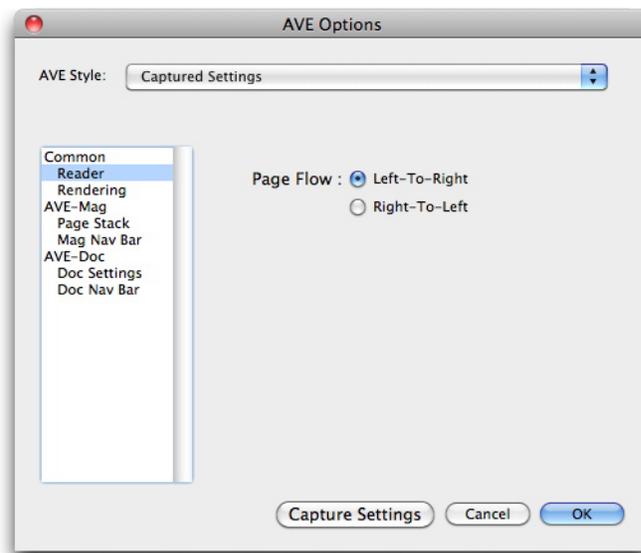
To preview the issue immediately after export in the Quark App Studio Issue Previewer app on your iPad, launch the app on your iPad, check **Preview in** and enter the upload URL for your iPad or instance of iOS Simulator (without the `http://` prefix). (For more information, see "[Testing an App Studio issue](#).")

When you're finished, click **Save**. QuarkXPress exports two files:

- The issue (.zave) file.
- The manifest (.zavem) file for the issue. You will need this file if you plan to publish the issue file using the Quark App Studio Publishing Portal (for more information, see "[Publishing an App Studio issue](#)")

Setting App Studio issue export options

If you click the **Options** button in the **Export AVE for iPad** dialog box, the **AVE Options** dialog box displays. This dialog box lets you control a variety of settings for AVE export.



AVE Options dialog box

The **Reader** pane lets you control whether pages flow left-to-right or right-to-left.

The **Rendering** pane includes the following controls:

- **PDF Output Style:** Lets you control the PDF output style used in creating the issue.
- **JPEG Quality:** Lets you control the JPEG quality used in creating the issue.
- **Device Resolution:** Lets you specify the resolution of the output device.
- **Use Resolution from Device Resource When Available:** Lets you use the device resolution specified in the **New Project** or **New Layout** dialog box rather than specifying a particular resolution.
- **Include Page Previews:** If you include page previews, the file size is larger but page turning is faster.

The **Page Stack** pane lets you specify whether sections should be converted to page stacks and, if so, how the stack should be oriented.

The **Mag Nav Bar** pane lets you specify a quality and size for horizontal and vertical navigation bars. (AVE-Mag export only.)

The **Doc Settings** pane lets you specify how pages display, turn guided reading on or off, control snapping from page to page, and specify whether zoom works or not. (AVE-Doc export only.)

The **Doc Menu Bar** pane lets you control the display of the menu bar that becomes available when the end user taps the issue. You can control the visibility and translucency of the menu bar, what type of bar it is, its height, its color, and which menu items it includes. (AVE-Doc export only.) The options for each menu item are as follows:

- **Type:** Lets you choose the type of menu item:
 - **Flexible Space:** Adds a flexible space with no associated command.
 - **Close:** Closes the issue and returns to the app.
 - **Settings:** Displays a pop-up that lets the end user control how the page displays, how it snaps, the maximum zoom, and whether or not to include a Read menu.
 - **Help:** Displays the Help for the app. This content is specific to each app template.
 - **Browser:** Lets the end user browse through thumbnail images of the issue's pages.
 - **Summary:** Displays the Contents page. (For more information, see "[Setting the Contents page.](#)")
 - **Label:** Lets you add a label to the menu bar.
 - **Page Label:** Adds a page label, in the form (Page X / Total)
 - **Add/Remove a Bookmark:** Lets the end user add a bookmark that points to the active issue page. Valid for AVE-Doc only.
 - **Add/Remove a Note:** Lets the end user add a note to the active issue page. Valid for AVE-Doc only.
 - **Show bookmarks and notes:** Displays all bookmarks and notes in the active issue.
 - **Show All Pages:** Displays all of the pages of the active issue in a grid.
- **Title:** Lets you enter a title for the menu item.
- **Color:** Lets you specify a color for the title.
- **Image:** Lets you display an image instead of a title.
- **B.G.:** Lets you specify an image to display behind the menu item.

You can use the plus (+) and minus (-) buttons to add and remove menu items.

The **Doc Nav Bar** pane lets you control the display of the document navigation bar, which displays at the bottom of the screen and lets the user navigate easily through the issue. It includes the following controls:

- **None:** Suppresses the document navigation bar.
- **Spread View:** Displays a spread-based document navigation bar.

- **Always Visible:** If this is unchecked, the end user can display the document navigation bar by tapping the issue.
- **Translucent:** If this is checked, the document navigation bar is translucent.
- **Height:** Lets you control the height of the document navigation bar.
- **Color:** Lets you control the color of the frame around the thumbnail.
- **Draw:** Click **Use Thumb** to display page thumbnails in the navigation bar or **Use Page Number** to display page numbers in the navigation bar.
- **Use Popup:** Check this box to display pop-up page previews above the navigation bar.
- **Popup Size:** Lets you specify the size of the longest dimension of the pop-up previews in pixels.
- **Display Full Screen Button:** Includes a button that fills the screen with a grid of page previews. The end user can then tap the page he or she wants to navigate to.
- **Stack View:** Displays a stack-based document navigation bar.
 - **Always Visible:** If this is unchecked, the end user can display the document navigation bar by tapping the issue.
 - **Width:** Lets you specify the width of the stack thumbnails in pixels.
 - **Height:** Lets you specify the height of the stack thumbnails in pixels.
 - **Article Gap:** Lets you specify the distance between articles in pixels.
 - **Page Gap:** Lets you specify the distance between pages in pixels.
 - **Scroll Deceleration:** Lets you control how quickly the pages scroll.

Replacing resources in exported App Studio issues

An App Studio (.zave) file is a .zip file with a different file name suffix. If you decompress a .zave file, you can have access to the files that make up the issue, including .jpg and .png files. If you need to update any of these resources, you can. If you do so, however, you must create a new manifest (.zavem) file that matches the updated .zave file.

To update a .zave file and create a new .zavem manifest:

- 1 Change the suffix of the .zave file to `.zip`.
- 2 Decompress the .zip file.
- 3 Update or replace resources, as necessary.
- 4 Compress contents of the folder containing the issue's assets to create a .zip file.
- 5 Change the suffix of the .zip file to `.zave`.
- 6 Launch the Ave Manifest Generator application. This file is located in the "App Studio" folder, which is in the QuarkXPress application folder. The **AVE Manifest Generator** window displays.

CREATING AN APP STUDIO ISSUE



AVE Manifest Generator window

- 7 Drag the new .zave file onto the window. A dialog box displays.
- 8 Click OK. The manifest file is generated. The manifest file has the same file name as the .zave file, but has a `.manifest` suffix. Another dialog box displays.
- 9 Click OK.
- 10 Change the name of the suffix of the manifest file to `.zavem`.

When you publish this file with the Quark App Studio Publishing Portal, you must also upload the new .zave file. For more information, see "[Publishing an App Studio issue](#)."

Installing Quark App Studio Issue Previewer

Quark App Studio Issue Previewer lets you test your App Studio issues on Mac OS. You can download this app to your iPad for free from the Apple App Store, or you can install it on your Mac and run it in iOS Simulator. To install Quark App Studio Issue Previewer on your Mac:

- 1 Verify that Xcode 4.0 or later is installed.
 - 2 Locate the "Issue Previewer for iOS Simulator.zip" file in the "App Studio" folder.
- ➔ For convenience, this file is supplied on both Mac OS and Windows. However, it will work only on a Mac OS computer.
- 3 Decompress the zip file in this folder. This creates a folder with a name that consists of a long series of characters.
 - 4 Copy the decompressed folder to `[Hard Disk]/Users/[User Name]/Library/Application Support/iPhone Simulator/[iOS version]/Applications`.
- ➔ On Mac OS 10.7 and later, the "Library" folder is hidden by default. To navigate to this folder, choose **Go > Go to Folder** in the Finder, then enter `~/Library` and click **Go**.

- 5 Navigate to [Hard Disk]/Developer/Platforms/iPhoneSimulator.platform/Developer/Applications and double-click the file "iOS Simulator." The iOS Simulator launches.
- 6 In the iOS Simulator, choose **Hardware > Device > iPad**.
- 7 In the iOS Simulator, swipe to the left to display the second screen if necessary to reveal the "Issue Previewer" icon.
- 8 To verify that the test app has installed correctly, click the icon labeled "Issue Previewer." The Quark App Studio Issue Previewer app should launch.

Testing an App Studio issue

You can test an App Studio issue using an app named Quark App Studio Issue Previewer. You can download this app for free from the Apple App Store, or run it in the iOS Simulator on your Mac OS computer. In either case, you need to upload the issue to the app.

To test an App Studio issue (a .zave file):

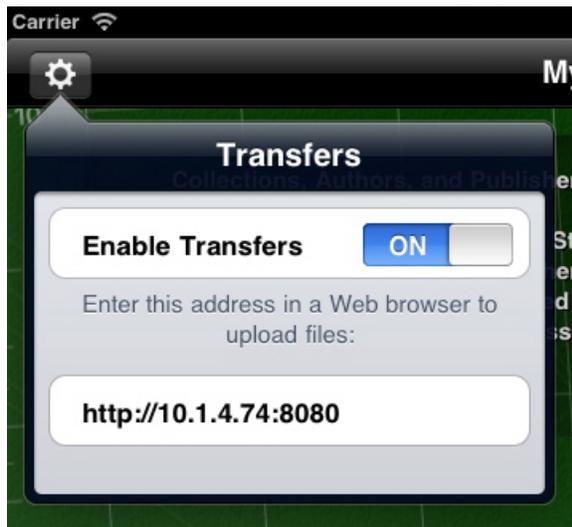
- 1 Export an App Studio issue (a .zave file). For more information, see "[Creating an App Studio issue](#)."
- 2 If you want to use the iOS Simulator, make sure the test app is installed. For instructions on how to install the Quark App Studio Issue Previewer app in the iOS simulator, see "[Installing Quark App Studio Issue Previewer](#)."
If you want to use your iPad, download the free Quark App Studio Issue Previewer from the Apple App Store.
- 3 Launch the Quark App Studio Issue Previewer app.

CREATING AN APP STUDIO ISSUE



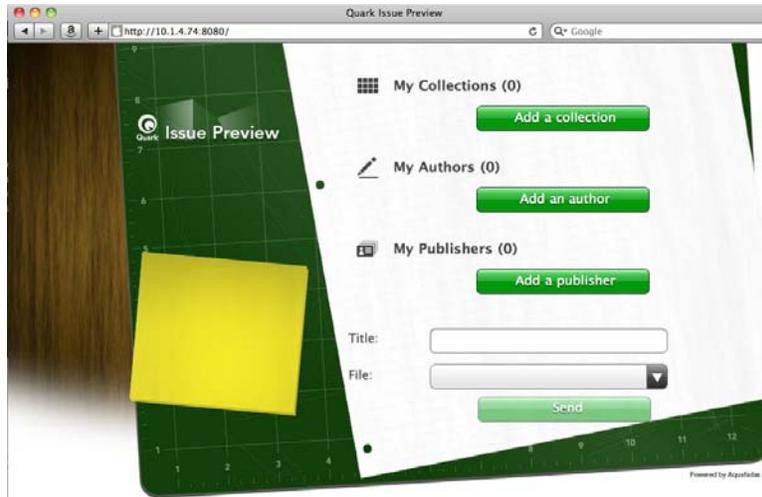
Bookshelf screen of the Issue Previewer app

- 4 Tap the button at the top left of the screen and set **Enable Transfers** to ON. The app displays a URL.



Upload URL displayed in the Issue Previewer app

- 5 On the computer where the .zave file is, launch a Web browser and enter the URL displayed in the app. An upload page displays.



Upload page for the Issue Previewer app

- 6 Enter a name for the issue in the **Title** field (optional).
- 7 Click the **File** field, navigate to the .zave file, and click **Open**.
- 8 To specify a collection, author, or publisher for the issue, click the appropriate button.
- 9 Click **Send**. The issue is uploaded to the app's bookshelf.
- 10 To view the issue, tap it once, then tap **Read**.
 - To switch from horizontal to vertical mode in the iOS Simulator, choose **Hardware > Rotate Left** or **Hardware > Rotate Right**.
 - To zoom in layout mode, use a pinch-out gesture. In the iOS Simulator, you can do this by pressing Option and then clicking and dragging on the issue.
 - To switch from reflow view to layout mode, use the button at the top of the screen.
- 11 To remove the issue from the bookshelf, tap the **Edit** button at the upper right, then tap the black X in the issue's corner. To return to normal mode, tap **Done**.

Creating an App Studio app

The App Studio Factory application makes it easy for you to create and export your own customized iPad app.

Every App Studio app is based on an app template. There are different app templates for different kinds of App Studio apps. For example, one template lets you create a magazine app, where customers can download issues periodically over time, and another template lets you embed a single issue in the app with no options to purchase others.

You can use any app template you want for testing, but you must purchase an app template license before you use it to build your final app for submission to the App Store. Until you purchase your app template, the App Studio apps you build will only be able to display test issues, and will only be able to do so on iPads and instances of iOS Simulator that you have registered with the App Studio Publishing Portal.

For more information, see "[Registering devices on the App Studio publishing portal](#)" and "[Purchasing an App Studio app template license](#)."

➔ You can also develop your own App Studio apps with the App Studio Framework.

Registering devices on the App Studio publishing portal

Before you can test your App Studio app in the iOS Simulator or on an iPad, you must register your iOS Simulator instance and your iPads on the App Studio publishing portal. There are two ways to do this.

Registering with Quark App Studio Issue Previewer

To register a device or iOS Simulator instance using Quark App Studio Issue Previewer:

- 1 Launch the app on the iPad or in the iOS Simulator. For more information, see "[Installing Quark App Studio Issue Previewer](#)."
- 2 Click the **Tester** tab.
- 3 Enter your App Studio Publishing Portal login and password.
- 4 Click **Connect**.

Once you have registered a device or an iOS Simulator instance on the App Studio portal, you can use that device or instance to test your App Studio app and issues.

Registering on the App Studio Publishing Portal

To register a device using the App Studio Publishing Portal:

- 1 Go to <http://appstudio.quark.com> and log in.
- 2 Click **Profile**. The **Profile** screen displays.
- 3 Click **Test devices**. The **Test devices** pane displays.
- 4 Click **Add a device**, enter the UDID for your iPad, click the Apple icon under **Platform**, and then click **Add**. (For more information, see <http://support.apple.com/kb/HT4061>.)
- 5 Click **Update**.

Once you have registered a device on the App Studio portal, you can use that device to test your App Studio app and issues.

Creating an App Studio app

In general, the process of creating an App Studio app is as follows:

- 1 Create the app resources. These are the picture files that give each app its distinctive appearance and branding.
- 2 Build the app in App Studio Factory.

The topics below describe each of these steps in detail.

- ➔ For convenience, App Studio Factory is included in the application folder for both Mac OS and Windows. However, it will work only on a Mac OS computer.

Preparing to submit an App Studio app to Apple

The process in general for preparing to submit an App Studio app to Apple is:

- 1 Create your Apple iOS Developer account.
- 2 Get your iPad's device ID.
- 3 Create your iOS Development Certificate.
- 4 Register your iPad so you can test on it.
- 5 Create your app ID.
- 6 Create your Development Provisioning Profile.
- 7 Create your App Store Distribution Provisioning Profile.
- 8 Create your app description.

CREATING AN APP STUDIO APP

The following topics explain this process in detail. For more information, see the iTunes Connect Developer Guide, which is available to Apple developers at itunesconnect.apple.com.

Creating your developer account

Before you can proceed, you must create an Apple iOS Developer account. To do so:

- 1 Go to <http://developer.apple.com/programs/ios/> and click **Enroll Now**.
- 2 Click **Continue** and follow the instructions on the screen. When you are asked to enroll as an individual or a company, choose the appropriate option.
- 3 Go to itunesconnect.apple.com, click **Contracts, Tax, and Banking**, and enter into an iOS® Paid Applications contract with Apple. If you plan to charge for your app or issues, this contract must be in effect, with valid bank info entered, before you submit your app to the Apple app store.

Getting your iPad's device ID

To test your app on your iPad, you must specify its device ID (that is, its UDID). One way to do this is with Issue Previewer (for more information, see "[Registering with Quark App Studio Issue Previewer](#)"). Here is another way to get your iPad's UDID:

- 1 Plug your iPad into your computer.
- 2 Launch iTunes 7.7 or later.
- 3 Select the iPad in iTunes.
- 4 Click the **Serial Number** label. The label changes to **Identifier (UDID)**.



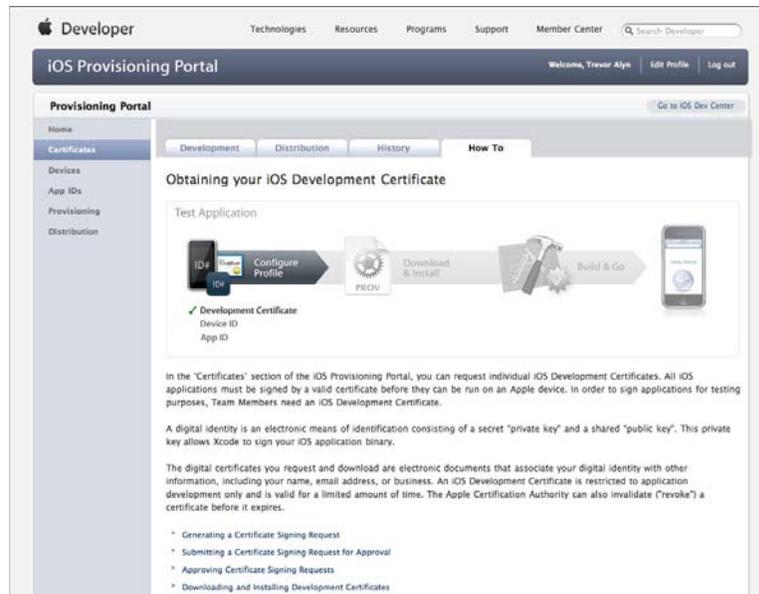
- 5 Press Command+C to copy the device ID to the clipboard.
- 6 Paste the device ID into a text file. It should be a very long code with both numbers and lower-case letters, but no hyphens or other characters.

Creating your iOS Development Certificate

You must have an iOS Development Certificate to develop apps and test them on an iPad. To create this Certificate:

- 1 Go to <http://developer.apple.com/devcenter/ios/> and log in.

- 2 Under **iOS Developer Program** in the sidebar, click **iOS Provisioning Portal**.
- 3 Click **Certificates** in the left sidebar.
- 4 Click the **How To** tab. The **Obtaining your iOS Development Certificate** screen displays.

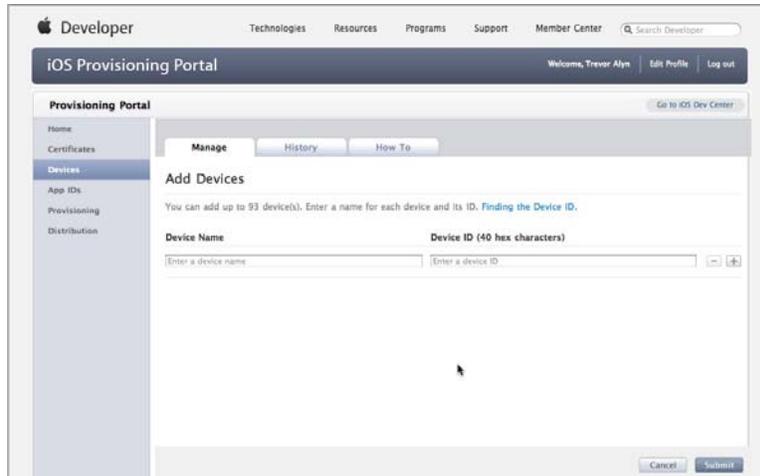


- 5 Follow the instructions, including those under the **Generating a Certificate Signing Request, Submitting a Certificate Signing Request for Approval, Approving Certificate Signing Requests, and Downloading and Installing Development Certificates**.

Registering devices

Until they are in the App Store, apps can be run only on devices that they are registered to run on. To register your iPad for testing:

- 1 Click **Devices** in the left sidebar.
- 2 Click **Add Devices**. The **Add Devices** screen displays.

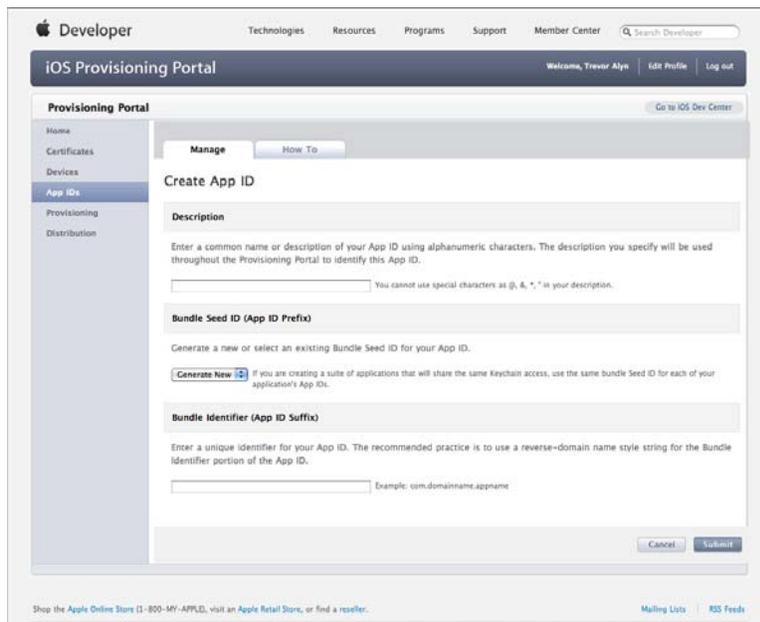


- 3 Enter a name for your iPad (for example, "Jane's iPad") in the **Device Name** field, then paste the iPad's UDID in the **Device ID** field.
- 4 To add an additional device, click the + button to the right of the **Device ID** field.
- 5 Click **Submit**.

Creating an app ID

Every app must have a unique ID. To create your app ID:

- 1 Click **App IDs** in the left sidebar. The **App IDs** screen displays.
- 2 Click **New App ID**. The **Create App ID** screen displays.



- 3 Enter the name of your app in the **Description** field. Note that this is NOT the name that will display for the app in the App Store.
- 4 Leave the **Bundle Seed ID (App ID Prefix)** drop-down menu set to **Generate New**.
- 5 In the **Bundle Identifier (App ID Suffix)** field, enter a unique identifier for your app, using the form `com.[your organization's name].[name of your app]`. For example, if the app is being created for 123 Productions, the bundle identifier might be `com.123productions.123magazine`.
- 6 Click **Submit**.

Setting up for subscriptions

Some App Studio app templates allow customers to purchase auto-renewable subscriptions to App Studio issues. In order for this feature to work, you must use such an app template, and you must do some additional setup on the Apple developer Web site.

- ➔ You must complete this step before you create your Provisioning Profiles. (For more information, see the two topics after this one.)
- ➔ If you do not want to offer auto-renewable subscriptions in your app, you do not have to complete this step.

To set up for auto-renewable subscriptions:

- 1 Click **App IDs** in the left sidebar. The **App IDs** screen displays.
- 2 Click the name of the app you want to enable subscriptions for. The **Configure App ID** page displays.
- 3 Check **Enable for Apple Push Notification Service**.
- 4 Click the **Configure** button for **Production Push SSL Certificate**. (You do not need a Development Push SSL Certificate.) The **Apple Push Notification Service SSL Certificate Assistant** screen displays.
- 5 Follow the instructions on the screen. The instructions will direct you to create a certificate signing request, upload it in the **Apple Push Notification Service SSL Certificate Assistant** screen, and download an Apple Push Notification Service SSL certificate. This file should have a name like "aps_production_identity.cer."
- 6 Double-click the certificate file to install it.
- 7 Launch the Keychain Assistant application (Applications > Utilities) and click My Certificates in the list on the left. A list of certificates displays.
- 8 Expand the certificate named **Apple Production iOS Push Services**. A key icon displays.
- 9 Select both **Apple Production iOS Push Services** and the line with the key icon.
- 10 Control+click the two selected lines and choose **Export 2 items**, then save the items in Personal Information Exchange (.p12) format with the file name

CREATING AN APP STUDIO APP

`server_certificates_bundle_sandbox.p12`. Do not enter a password. If a dialog box displays, stating that Keychain Access wants to export a key, enter your login password and then click **Allow**. (You may have to do this twice.)

11 Put the `server_certificates_bundle_sandbox.p12` file into a folder named "temp".

12 Launch the Terminal application (Applications > Utilities). A **Terminal** window displays. In this window, type the letters `cd` followed by a space.

13 Drag the "temp" folder onto the **Terminal** window. The path of the "temp" folder is inserted automatically.

14 Click the title bar of the **Terminal** window to make sure it is active, then press Return.

15 Enter the following command in the **Terminal** window, followed by a Return:

```
openssl pkcs12 -in server_certificates_bundle_sandbox.p12 -out
server_certificates_bundle_sandbox.pem -nodes -clcerts
```

The phrase `Enter Import Password` displays.

16 Press Return. The phrase `MAC verified OK` displays.

17 Quit the Terminal application.

18 Open the "temp" folder and locate the file named "server_certificates_bundle_sandbox.pem". Keep this file handy; you will need it later. (For more information, see "[Setting up auto-renewable subscriptions](#).")

➔ There are some additional steps necessary for auto-renewable subscriptions. These steps will be described in later topics.

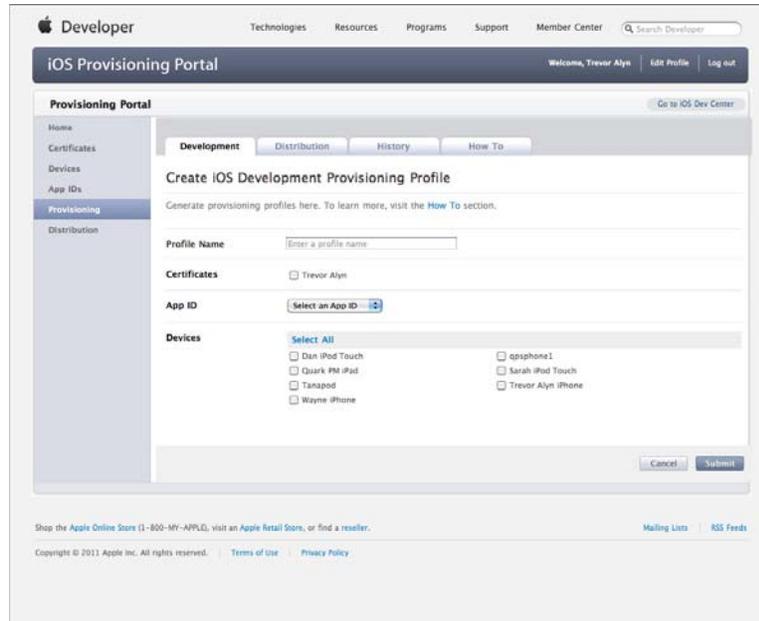
Creating a Development Provisioning Profile

A Provisioning Profile is a file that enables a developer to run an app on an iPad, rather than simply in the iPad simulator. There are two types of provisioning profiles: Development Provisioning Profiles (used while developing an app) and Distribution Provisioning Profiles (used to create final apps). To create your Development Provisioning Profile:

1 In the left sidebar, click **Provisioning**.

2 Click the **Development** tab.

3 Click **New Profile**. The **Create iOS Development Provisioning Profile** screen displays.

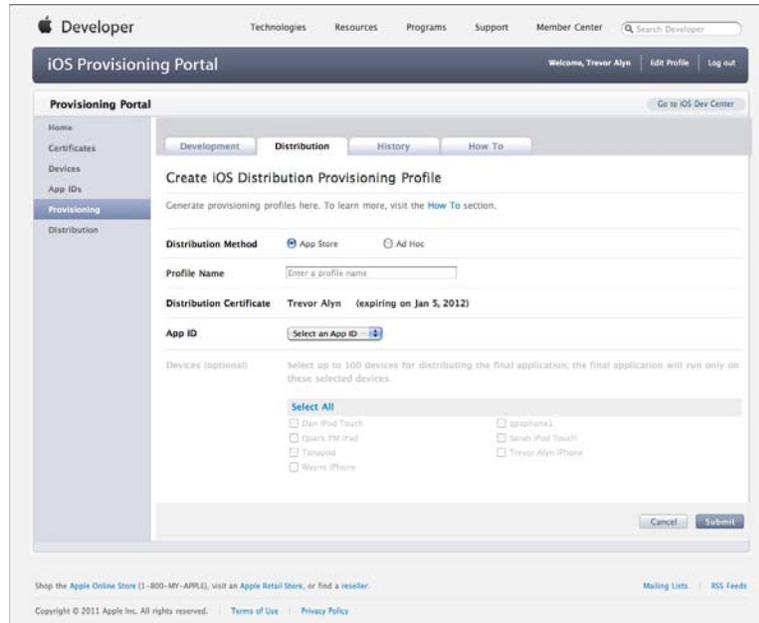


- 4 In the **Profile Name** field, enter "App Studio Dev Provisioning Profile".
- 5 Under **Certificates**, check the box for your certificate.
- 6 Choose your app's app ID from the **App ID** drop-down menu.
- 7 Under **Devices**, click **Select All** or manually check the boxes for all of the devices listed.
- 8 Click **Submit**. The **Development Provisioning Profiles** screen displays.
- 9 Before you can download the development provisioning profile, you must reload the page. After you reload the page, the **Download** button should become available for the Development Provisioning Profile you just created.
- 10 Click **Download** to download the Development Provisioning Profile to your desktop. This file will have a name like "App_Studio_Dev_Provisioning_Profile.mobileprovision."
- 11 Double-click on the downloaded file to install it.

Creating an App Store Provisioning Profile

You must have an App Store Distribution Provisioning Profile to build an app for submission to the App Store. To create an App Store Provisioning Profile:

- 1 In the left sidebar, click **Provisioning**.
- 2 Click the **Distribution** tab.
- 3 Click **New Profile**.
- 4 Under **Distribution Method**, click **App Store**.



- 5 In the **Profile Name** field, enter "App Studio Dist Provisioning Profile".
- 6 Choose your app's app ID from the **App ID** drop-down menu.
- 7 Click **Submit**. The **Distribution Provisioning Profiles** screen displays.
- 8 Before you can download the distribution provisioning profile, you must reload the page. After you reload the page, the **Download** button should become available for the distribution provisioning profile you just created.
- 9 Click **Download** to download the distribution provisioning profile to your desktop. This file will have a name like "App_Studio_Dist_Provisioning_Profile.mobileprovision."
- 10 Double-click on the downloaded file to install it.

Creating an app description in iTunes Connect

Before you can submit your app to Apple, you need to create a description for that app on the Apple developer Web site.

- ➔ If you plan to sell your issues, you must create in-app purchases before you execute the following steps. For more information, see "[Setting up in-app purchases](#)" and "[Setting up auto-renewable subscriptions](#)."

To create an app description:

- 1 Go to <http://itunesconnect.apple.com> and log in (if necessary). The **iTunes Connect** screen displays.
- 2 Click **Manage Your Applications**. The **Manage Your Apps** screen displays.
- 3 Click **Add New App**. The **App Information** screen displays.

- 4 In the **App Name** field, enter the name of your app as you want it to display in the App Store.
- 5 In the **SKU Number** field, enter a unique value that can be used to identify your app.
- 6 Choose the ID of your app from the **Bundle ID** drop-down menu. (This is the App ID you created in "*Creating an app ID.*") Make sure you DO NOT select **Xcode: Wildcard AppID - ***.
- 7 Click **Continue**. The availability date and price tier screen displays. Follow the instructions on the screen to set the price point and availability date of your app. You can choose the current date to get the app posted as soon as possible.
- 8 Click **Continue**. The metadata and rating screen displays. Complete the forms on this screen.
- 9 If you are charging for issues, you must associate your in-app purchases with the app. To do this, click **Edit** in the **In-App Purchases** area, then check all of the in-app purchases you want associated with the app and click **Save**.
- 10 In the **Uploads** area, upload a 512 x 512 pixel PNG file where it says **Large 512x512 Icon**, and upload a dummy screen shot where it says **iPad Screenshots**. You can change both of these pictures later.
- 11 Click **Ready to Upload Binary**.

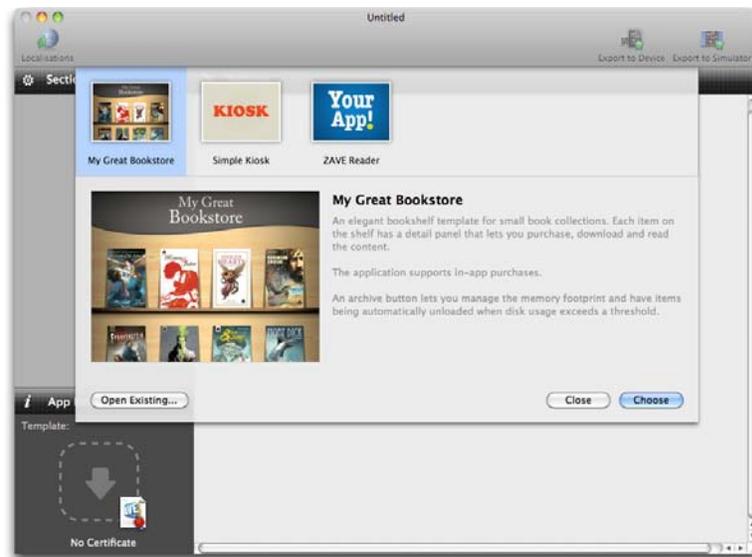
Building an app with App Studio Factory

App Studio Factory is an application for Mac OS (10.6 and later) that lets you build App Studio apps without having to do any programming. Before you create your app in App Studio, you will need your App Studio app certificate file. For more information, see "*Downloading an App Studio app certificate.*"

Also, you must have Xcode 4.0 or later installed. You can get Xcode for free from the Apple developer Web site at <http://developer.apple.com>.

To build an app in App Studio Factory:

- 1 Launch App Studio Factory (located in the QuarkXPress application folder). The App Studio Factory window displays.



App Studio Factory window

- 2 An App Studio app template determines the functionality and general layout of an app. You can choose from a number of different app templates. To see a description of an app template, click its icon at the top of the window. When you know which app template you want to use, select it and then click **Choose**.
- 3 To authenticate your app, drag your App Studio app certificate to the area on the lower left. (For more information, see "[Downloading an App Studio app certificate](#).") This will automatically fill several of the App Studio Factory fields for the app, including the app icon and the Bundle ID.
- 4 The **Sections** list on the left includes an entry for each type of app information you can customize. When you click a section, all of the configurable elements in that section display in the **Settings** area on the right. Click each of these sections, one by one, and customize anything you want to customize.

For pictures, you can view a large preview by clicking the magnifying glass icon. To replace a picture, simply drag a replacement PNG file onto the picture in App Studio Factory. The size (in pixels) that each picture needs to be displays beneath the picture.

 - ➔ Some fields are not editable. For example, the **Bundle ID** value is determined by the certificate. This is by design.
 - ➔ The icon in the App Icon field is automatically scaled down to fit the requirements of the template. You can replace this icon with an optimized version if you want.
 - ➔ You can display all of the fields and resources for the app by selecting all of the items in the **Sections** list.
- 5 If you are creating an app with an embedded issue, the **Sections** list includes an entry named **AVE Content**. Drag the issue you want to embed onto the **AVE file (.zave)** area.

To get the AVE Certificate file, log into the App Studio Publishing Portal, click **Publish**, choose the title or collection the issue belongs to, click the pencil icon next to the issue, click the **Downloads** pane, and then click **Download Ave Certificate**. Once the file has downloaded, drag it onto the **AVE Certificate** area.

- 6 Some templates are available with strings that have been translated into multiple languages. To include multiple languages in your app, click **Localizations** on the upper left, then check the languages you want to be included in the app. When the operating system matches one of these languages, the app will display the strings in that language.
- ➔ If you include (for example) French resources in your app, but your issues are not available in French, your customers may be unhappy. So you might want to check only the language that matches the language of the issues you are selling.
- 7 To test the app in the iOS Simulator application, click **Export to Simulator**.
- ➔ For information on testing issues in an App Studio app, see "[Testing an App Studio issue](#)."
- ➔ Although your app can display issues in the iOS Simulator, it cannot pull pricing data from the App Store. Consequently, pricing for issues probably will not match the pricing you have set up in iTunes Connect.

For information on exporting an app for testing on your iPad or submission to Apple, see "[Exporting an app with App Studio Factory](#)."

Purchasing an App Studio app template license

Before you can create a final app and submit it to the Apple App Store, you must purchase a license for the type of app template you have used to create your app. To purchase an app type template:

- 1 In App Studio Factory, click the Buy button at the bottom left of the main window, above the app certificate. The appropriate page opens in your default Web browser.
- 2 Follow the instructions on the screen to purchase the license for the type of app template you want to use. When you are finished, you will receive a serial number.
- 3 Log into the Quark App Studio Publishing Portal at <http://appstudio.quark.com> and click **App**. The **App Specifications** screen displays.
- 4 Click your app in the list on the left.
- 5 Under **Templates for this App**, click **Add an App Template**. The **Store** screen displays.
- 6 Enter the serial number in the field and then click **Enter**.
- 7 Click **OK**. Your app certificate is updated on the portal.
- 8 Download and install the updated app certificate. (For more information, see "[Downloading an App Studio app certificate](#)" and "[Building an app with App Studio Factory](#).")

- 9 Export the app. For more information, see "[Exporting an app with App Studio Factory](#)."

Exporting an app with App Studio Factory

When you export an app from App Studio Factory, you can export it for testing on your iPad, or you can export it for submission to Apple.

To begin the process, click **Export to Device**. The **Save As** dialog box displays.

If you want to export for testing on your iPad:

- Choose **Release** from the **Configuration** drop-down menu.
- Choose your Development Provisioning Profile from the **Profile** drop-down menu. (For more information, see "[Creating an App Store Provisioning Profile](#).")
- Choose the appropriate App Studio app certificate from the **Certificate** drop-down menu.

When you click **Save**, App Studio Factory creates a file with a `.ipa` suffix. To install your app on your iPad, drag the `.ipa` file to your iTunes icon in the dock, then sync your iPad with iTunes. (Alternatively, you can use the iPhone Configuration Utility; for more information, see <http://www.apple.com/support/iphone/enterprise/>.)

- ➔ The app will not work on the iPad unless you registered the iPad for testing. For more information, see "[Registering devices](#)."

If you want to export for submission to the Apple App Store:

- Choose **Distribution** from the **Configuration** drop-down menu.
- Choose your App Store Provisioning Profile from the **Profile** drop-down menu. (For more information, see "[Creating an App Store Provisioning Profile](#).")
- Choose the appropriate App Studio app certificate from the **Certificate** drop-down menu.

When you click **Save**, App Studio Factory creates a file with a `.ipa` suffix. For information on submitting this file to Apple, see "[Submitting your app to Apple](#)."

Testing an App Studio app

When you have finished creating your app and App Studio issue, you can test them on your iPad if you have one, or on a Mac OS computer running Mac OS 10.6.6 or later.

- ➔ Before you can test an issue, you must register the device or the copy of iOS Simulator you want to test it on. For more information, see "[Registering devices on the App Studio publishing portal](#)."
- ➔ Always test your app on a real device before you submit it. Real devices often work differently than the iOS Simulator.

- ➔ For instructions on testing an issue once you've got your app running, see "[Testing an App Studio issue](#)."

Installing an App Studio app on your iPad

To test your app and issue on your iPad:

- 1 Use App Studio Factory to generate an .ipa file. For more information, see "[Creating an App Studio app](#)."
- 2 In the Mac OS Finder, drag the .ipa file onto the iTunes icon. Note that this must be the copy of iTunes that is synced with your iPad.
- 3 Sync the iPad with iTunes. This should install your app on your iPad.

- ➔ Alternatively, you can use the iPhone Configuration Utility; for more information, see <http://www.apple.com/support/iphone/enterprise/>.

Testing an App Studio app in the iOS Simulator

To test your app and issue in the iOS Simulator application:

- 1 Verify that you are running Mac OS® 10.6.6 or later.
- 2 Download and install Xcode® 4.0 or later. You can download Xcode at <http://developer.apple.com/xcode/index.php>.
- 3 In App Studio Factory, open your app project and click **Export to Simulator**. The app opens in the iOS Simulator.

Managing App Studio issues

To test an App Studio issue and then make it available to customers who have your App Studio app, you must do the following things:

- 1 Upload the issue to a publicly available Web server and get its URL. Note that this must be a direct URL that ends with the name of the .zave file; a redirect URL will not work. You might need to ask your hosting provider to adjust the server's MIME type settings to allow the serving of .zave files.
- 2 Create an account on the App Studio Publishing Portal.
- 3 Create a title or collection entry on the App Studio Publishing Portal. If you are creating issues of a magazine, create a title that uses name of the magazine. If you are creating books, create a collection that identifies the book collection (such as a publisher or series name).
- 4 Register your app on the App Studio Publishing Portal. This lets you connect your App Studio issues with your App Studio app.
- 5 Purchase a license for the App Studio template you want to use for publishing.
- 6 Purchase an issue license for your issue on the App Studio Publishing Portal.

For the first step, you can use any Web hosting service you choose, as long as it lets you post the issue file with a direct URL. The topics below explain how to do the remaining steps.

Creating an App Studio Publishing Portal account

Before you can make your App Studio issues available to your customers, you must create an account on the App Studio Publishing Portal. There is no charge for creating an account. To create an App Studio Publishing Portal account:

- 1 In a Web browser, go to <http://appstudio.quark.com>. The App Studio Publishing Portal login page displays.
- 2 Click **Don't Have an Account?**, then follow the instructions on the screen.

Registering an App Studio app

- ➔ Before you can register an App Studio app in the App Studio Publishing Portal, you must create an App Studio Publishing Portal account (see "[Creating an App Studio Publishing Portal account](#)"), and create a title or collection (see "[Creating an App Studio Publishing Portal title or collection](#)").

To test your App Studio issues and then make them available to your customers, you must first register your App Studio app on the App Studio Publishing Portal. This lets you connect your App Studio issues with your App Studio app. To register an app:

- 1 In a Web browser, go to <http://appstudio.quark.com>. Log into the portal by entering your login and password. The Home page displays.
- 2 If you have not yet created a title or collection, create a title or collection. For more information, see "[Creating an App Studio Publishing Portal title or collection](#)."
- 3 If you have not yet created your app in iTunes Connect, create your app there. For more information, see "[Creating an app description in iTunes Connect](#)."
- 4 Under **Manage Your App Specifications**, click **Enter**. The **App Specification Wizard** page displays.
- 5 In the **App Specification Name** field, enter the name of your app as it should appear on this site, in App Studio Factory, and on the end user's device. The number of characters you can include in this name depends on the characters you use, but in general you should use 12 characters or less.
- 6 In the **Bundle ID** name, enter the bundle ID you created for your app on the Apple Developer Web site. For more information, see "[Creating an app ID](#)."

Bundle Identifier (App ID Suffix)

Enter a unique identifier for your App ID. The recommended practice is to use a reverse-domain name style string for the Bundle Identifier portion of the App ID.

Example: com.domainname.appname

Bundle ID

- 7 In the **Localized Name and Description** area, click **Add** to add a name and description for this app in a particular language. You can add as many different-language names and descriptions as you want.
- 8 In the **App Icon** field, upload a picture in PNG or JPEG format at a size of 72 x 72 pixels. This icon will be displayed on the device.
- 9 If you plan to deliver auto-renewable subscriptions through the app, click **Add a Certificate for Notifications**, then click **Browse** and upload your "server_certificates_bundle_sandbox.pem" file. (For more information, see "[Setting up for subscriptions](#).") This allows your app to notify customers when new issues become available.

- 10 Under **Download the App Certificate**, click the certificate icon. This will download an app certificate file, which you will need in order to build your app in App Studio Factory.
- ➔ If you have not yet purchased an issue license and an App Studio Factory app template license, you can download a new copy of the certificate after you purchase these items. However, you do not have to purchase them to test the app and published issues. For more information, see "[Downloading an App Studio app certificate](#)."
- 11 Before you can publish an issue, you must create a product. For more information, see "[Creating an App Studio product](#)."
- 12 Under **App Template Licenses**, you can purchase a license for the App Studio Factory app template you want to use. You can do this now, but you don't have to. For more information about app templates, see "[Building an app with App Studio Factory](#)."
- 13 Click **OK**. The app specification is added.
- 14 Click **Home** to return to the Home screen.

Creating an App Studio Publishing Portal title or collection

- ➔ Before you can create an App Studio Publishing Portal title or collection, you must create an App Studio Publishing Portal account (see "[Creating an App Studio Publishing Portal account](#)").

To make your App Studio issues available to your customers, you must first create a corresponding title or collection on the App Studio Publishing Portal. If you are creating issues of a magazine, create a title that uses the name of the magazine. If you are creating books, create a collection that identifies the book collection (such as a publisher or series name). To create a title or collection:

- 1 In a Web browser, go to <http://appstudio.quark.com>. Log into the portal by entering your login and password. The Home page displays.
- 2 Under **Manage Your Titles and Collections**, click **Enter**. The **Titles and Collections** screen displays.
- 3 In the sidebar, click **Add a Title or Collection**.
- 4 In the **Identifier** field, enter a value that follows the formula, `[domain].[company].[title or collection name]`. For example: `com.quark.quarkmagazine`. This is your App Studio title/collection identifier. You will use this value to tie a particular title to your app.
- 5 In the **Title or Collection Name** field, enter a name for the title or collection.
- 6 In the **Icon** field, upload a picture in PNG or JPEG format at a size of 128 x 128 pixels.
- 7 In the **Language** drop-down menu, select the main language in which the title or collection will be published. This does not have to be the same language as the app or the issues.

- 8 In the **Type** drop-down menu, select **Collection** for books or **Title** for periodical issues. If you choose **Title**, you can indicate how frequently the title is released.
- 9 In the **Category** area, select the relevant categories. These categories are used for search engine optimization.
- 10 In the **Localized Name and Description** area, click **Add** to add a name and description for this title or collection in a particular language. You can add as many different-language names and descriptions as you want.
- 11 In the **Issue Type** area, you can create different types of issues that should be sold for different prices. For example, you might have a **default** issue type for most of the issues of a magazine, a **Holiday Special** issue type for holiday issues that might sell at a different price, and so forth. Use this area to create a name for each issue type you want.
- 12 Click **OK**. The title or collection is added.
- 13 Click **Home** to return to the Home screen.

Purchasing an App Studio issue license

- ➔ Before you can purchase an issue license, you must create an App Studio Publishing Portal account (see "[Creating an App Studio Publishing Portal account](#)"), create a title or collection (see "[Creating an App Studio Publishing Portal title or collection](#)"), and register your app (see "[Registering an App Studio app](#)").

To make your App Studio issues available to your customers, you must first purchase one or more issue licenses on the App Studio Publishing Portal. Each license lets you publish an App Studio issue to a single application. To purchase issue licenses:

- 1 In a Web browser, go to <http://appstudio.quark.com>. Log into the portal by entering your login and password. The Home page displays.
- 2 Under **Purchase Publishing Licenses**, click **Enter**. The Publishing License Store screen displays.
- 3 Click **Purchase a License Serial Number**. The **Store** screen displays.
- 4 Click **Purchase a License Serial Number**. This link takes you to the Quark eStore, where you can buy licenses for issues and for app templates. Follow the instructions on the screen to purchase a serial number for an App Studio issue license.
- 5 Return to the Publishing Portal and enter or paste the serial number in the **Enter License Serial Number Field**, and then click **Enter**.
- 6 Click **OK**.
- 7 Click **Home** to return to the Home screen. You can now publish as many issues as you have licenses for. After that, you can publish only test issues.

Posting App Studio issues to a Web server

Before you can submit your app to the App Store, you must post your issues on a Web server. You can use a Web server that you own, or you can rent space inexpensively on someone else's Web server. For instructions on uploading issues to the Web server, talk to your Webmaster or Web server provider.

- ➔ Do not compress your .zave files before you post them. The URL that points to an issue must end with `.zave`.

Once you have posted your issues on the Web server, make sure you save the URL for each issue. You will need these URLs in order to publish the issues. For more information, see "[Publishing an App Studio issue](#)."

Publishing an App Studio issue

- ➔ Before you can make your App Studio issues available to your customers, you must create an App Studio Publishing Portal account (see "[Creating an App Studio Publishing Portal account](#)"), create a title or collection (see "[Creating an App Studio Publishing Portal title or collection](#)"), register your app (see "[Registering an App Studio app](#)"), and purchase an issue license (see "[Purchasing an App Studio issue license](#)").

- ➔ You do not have to purchase an issue license to publish test issues.

To publish an App Studio issue:

- 1 In a Web browser, go to <http://appstudio.quark.com>. Log into the portal by entering your login and password. The Home page displays.
- 2 Click **Publish**. The **Publish** screen displays.
- 3 Choose the title or collection you want to publish an issue of from the **Title or Collection** drop-down menu. Any issues in that title or collection display.
- 4 Click **Create a New Issue**. The **General** pane of the **Add Issue** screen displays.
- 5 Enter a name for the issue in the **Issue Name** field.
- 6 Enter a description for the issue in the **Description** field. When a customer taps an issue in a bookshelf or newsstand app, this description displays in a pop-up along with a preview, the price, and a summary.
- 7 Select an issue type from the **Issue Name** drop-down menu.
- 8 If the issue is part of a serial title, enter an issue number in the **Issue Number** field.
- 9 To display a headline in the **Description** field in your app, enter the headline in the **Issue Headlines** field. You can click the + button to the right to add additional headlines.
- 10 Click **Uploads** to display the **Uploads** pane.

- In the **Issue Thumbnail** field, upload a thumbnail picture of your issue's cover in PNG or JPEG format. The picture should be 256 pixels or less in width and height.
- In the **Issue Preview** field, upload a high-resolution preview picture for your issue in PNG or JPEG format. The picture should be 1024 pixels or less in width and height.
- In the **Issue Summary** field, upload a picture that summarizes the contents of your issue in PNG or JPEG format. This picture displays when a customer taps the **Summary** button while shopping for the issue. The picture should be 1024 pixels or less in width and height.

11 Click **Release Information** to display the **Release Information** pane.

- Use the **Release Date** field to specify the date when the issue should become available to users who have the corresponding app.
- If you want your issue to become unavailable as of a particular date, specify that date in the **End Date** field.
- If you want the issue to be a test issue, and to be available only in test editions of your app, check **Test Issue**. (If you do not own any issue licenses, you cannot uncheck this box.)

12 Click **.Zave File** to display the **.Zave File** pane.

- In the **URL of .zave File** field, enter the URL of the .zave file. This must be a direct URL that ends with the name of the .zave file.
- In the **Manifest** field, upload the .zavem file that QuarkXPress exported along with your .zave file. This is necessary for security reasons.

13 If you are developing your own app using the App Studio Framework, and that app uses metadata, click **Metadata** to display the **Metadata** pane and enter the appropriate metadata there.

14 Click **OK**.

➔ Your issue will not be available in your app until you associate it with the app. For more information, see "[Creating an App Studio product](#)."

Setting up in-app purchases

If you plan to charge for your issues, you must them as an in-app purchase with Apple. For information on how to do this, see "Managing Your In-App Purchases" in the *iTunes Connect Developer Guide* (available to registered Apple iOS developers at <http://developer.apple.com>).

The process, as it exists as of this writing, is as follows:

- 1 Go to <http://itunesconnect.apple.com/> and log in (if necessary).

- 2 Click **Manage Your Applications**. The **Manage Your Apps** page displays.
- 3 Click your app. The app's page displays.
- 4 Click **Manage In-App Purchases**. The **In-App Purchases** page displays.
- 5 Click **Create New**. The **Select Type** screen displays.
- 6 To specify what type of in-app purchase you're creating, click one of the following buttons.
 - **Consumable**: Click this option to create an in-app purchase for a Product. This option allows you to create a single in-app purchase and use it for every issue that matches a Product.
 - **Non-Consumable**: Click this option to create an in-app purchase for a single issue.
 - **Auto-Renewable Subscription**: Check this option to create an automatically renewing subscription.

The next screen displays.

- 7 In the **Reference Name** field, enter the title and date of the in-app purchase issue. For example:
 - If you are creating a Non-Consumable in-app purchase, enter something like `123 Magazine - Single Issue`.
 - If you are creating an Auto-Renewable Subscription in-app purchase, enter something like `123 Magazine Subscription`.
- 8 If you are creating a Non-Consumable in-app purchase, enter your bundle identifier, followed by the issue name and date, in the **Product ID** field. For example:
`com.123productions.123magazine.price.issue.default`.
- 9 To specify the language or languages of your issue, click **Add Language** and follow the instructions on the screen.
- 10 If you are creating a Non-Consumable in-app purchase, check **Cleared for Sale** and choose the price tier you want from the **Price Tier** drop-down menu. (For a corresponding list of prices, click **View Pricing Matrix**.)
- 11 If you are creating an Auto-Renewable Subscription in-app purchase, you must specify one or more available durations for the subscription. Click **Add Duration** and follow the instructions on the screen. . The **Create New In-App Purchase** page displays.
 - In the **Product ID** field, enter your bundle identifier, followed by the issue duration. For example: `com.123productions.123magazine.6months`.
 - Check **Cleared for Sale**.
 - Choose the price tier you want from the **Price Tier** drop-down menu. (For a corresponding list of prices, click **View Pricing Matrix**.)

- 12 In the **Screenshot for Review** area, click **Choose File** and upload a screen shot of what an issue of your app will look like when viewed on the iPad. Note that this image is only for review and will not be displayed in the App Store.

➔ This screen shot can be changed later.

- 13 Click **Save**.

Setting up auto-renewable subscriptions

If you want to sell auto-renewable subscriptions in your App Studio app, you must first create a Shared Secret on the Apple Developer Web site, then set up the auto-renewable subscriptions you need. For information on how to do this, see "Creating Auto-Renewable Subscriptions" in the *iTunes Connect Developer Guide* (available to registered Apple iOS developers at <http://developer.apple.com>).

You will need your Shared Secret when you create a Product for a subscription. For more information, see "[Creating an App Studio product](#)"

Creating an App Studio product

Once you have published an App Studio issue, you must link its issue type to the app (or apps) in which you want it to be available for sale.

On the App Studio publishing portal, a *product* is a link between an app and the issue type of a title. Once you have created a product, the app displays all of the corresponding title's issues of the selected type (assuming that the dates correspond with the current date).

To create an App Studio product:

- 1 Create an App Studio title or collection. (For more information, see "[Creating an App Studio Publishing Portal title or collection](#).") While doing so, create the issue type that corresponds to the type of issue you want to publish.
- 2 Create an iOS in-app purchase or subscription to correspond with the issue type. For more information, see "[Setting up in-app purchases](#)."
- 3 In the App Studio Publishing Portal (<http://appstudio.quark.com>), under **Manage Your App Specifications**, click **Enter**, then select the app you want to publish the issue for.
- 4 Under **Products for this App**, click **Add a Product**. The **Product creation** screen displays.
- 5 Select the target title or collection from the **Title or Collection** drop-down menu.
- 6 Choose the target option from the **Product Type** drop-down menu:
 - For a single issue (such as a book), choose **Publication**.
 - For a recurring issue (such as a magazine), choose **Subscription**.

MANAGING APP STUDIO ISSUES

- 7 Enter a product name in the **Product Name** field. This name should describe the product in general, not a specific issue of the product (for example, "Standard issue of magazine" or "Mid-tier priced book").
 - 8 *Publications only:* Enter an internal SKU in the **Default SKU** field. You can set this value to anything you like.
 - 9 *Publications only:* If the product is free, check **Product is Free**. Otherwise, enter the SKU of the in-app purchase or subscription in the **iOS SKU** field.
 - 10 *Subscriptions only:* To exclude a particular issue type from the subscription, check that issue type under **Types of Issues to Exclude**.
 - 11 *Subscriptions only:* Click **Subscription Details** to display the **Subscription Details** dialog box. Then:
 - Select a subscription frequency from the **Subscription Frequency** dialog box.
 - Enter your Shared Secret in the **Shared Secret** field. (For more information, see "[Setting up auto-renewable subscriptions](#).") To make the Shared Secret readable, check **Show Shared Secret**.
 - Enter an internal SKU in the **Internal SKU** field. You can set this value to anything you like.
 - Enter the SKU of the subscription in the **iOS SKU** field.
 - Click **OK**.
 - 12 Click **OK**.
- ➔ The list of issue types that is associated with a given app is stored in the App Studio app certificate file. To make the issues available in the app, you must use this app certificate file when you build your app. If you add issue types after you have built your app, you must download a new copy of your app certificate and use it to build a new version of your app. For more information, see "[Building an app with App Studio Factory](#)."

Downloading an App Studio app certificate

Before you can build an App Studio app in App Studio Factory, you must download an App Studio app certificate.

- ➔ Before you can download an App Studio app certificate, you must create an App Studio Publishing Portal account (see "[Creating an App Studio Publishing Portal account](#)"), create a title or collection (see "[Creating an App Studio Publishing Portal title or collection](#)"), and register your app (see "[Registering an App Studio app](#)").

To download an App Studio app certificate:

- 1 In a Web browser, go to <http://appstudio.quark.com>. Log into the portal by entering your login and password. The Home page displays.
 - 2 Click **App Specifications**. The **App Specifications** page displays.
 - 3 In the list on the left, select the app specification for the target app.
 - 4 Click the icon next to **Download the App Certificate**. The app certificate downloads to your desktop.
- ➔ Before you create your final app for submission to the Apple App Store, you will have to link at least one App Studio issue with your app certificate (for more information, see "[Creating an App Studio product](#)"). However, you do not have to do this to download an app certificate; you can download a "plain" app certificate, use it to develop your app in App Studio Factory, and then replace the "plain" app certificate with an updated app certificate when you're ready to create the final app for submission to the App Store.

Submitting your app to Apple

Before you submit your app to the App Store, make sure that you have followed all of the instructions in the "[Managing App Studio issues](#)" section. When you are finished:

- 1 Log in to your iOS developer account in the iOS Dev Center.
- 2 Click **iTunes Connect** in the sidebar on the right. The **iTunes Connect** screen displays.
- 3 Click **Manage Your Applications**. The **Manage Your Apps** screen displays.
- 4 Click the icon for your app. The app description screen displays.
- 5 Make any desired changes. In the **Uploads** area, upload final screen shots. (To take a screen shot on the iPad, press the Home button and then, while keeping it pressed, press the Power button once. The screen shot will be available in the Photos app.)
- ➔ If you are charging for issues, you must associate your in-app purchases with the app. To do this, click **Edit** in the **In-App Purchases** area, then check all of the in-app purchases you want associated with the app and click **Save**.
- 6 When you're sure everything is final, click **Save**.
- 7 In the Finder, navigate to `[drive]/Developer/Applications/Utilities` and double-click **Application Loader**.
- 8 Follow the instructions on the screen. When the **Choose an application** screen displays, choose the name of your app.
- 9 When the **Application Information** screen displays, click **Choose** and select the .ipa file you exported from App Studio Factory and click **Open**.
- 10 Follow the instructions on the screen. If you are asked whether your app contains cryptography, click **No**.

After your app has been successfully submitted, Apple will review it and contact you to let you know when it is approved.

Legal notices

© 1986-2022 Quark Software Inc. and its licensors. All rights reserved.

Protected by the following United States Patents: 5,541,991; 5,907,704; 6,005,560; 6,052,514; 6,081,262; 6,633,666 B2; 6,947,959 B1; 6,940,518 B2; 7,116,843; and other patents pending.

Quark, the Quark logo, Quark Publishing System, and QPS are trademarks or registered trademarks of Quark Software Inc. and its affiliates in the U.S. and/or other countries. All other marks are the property of their respective owners.

Index

.zave files 9, 45
 .zavem files 45

A

actions 35
 app IDs 54
 app specifications 10
 App Studio apps 6
 App Studio assets 41
 App Studio Factory 11, 50
 App Studio issues 6
 App Studio palette 14
 app templates 11
 app templates, embedded issue 11
 app templates, kiosk 11
 Apple 10
 Apple App Store 74
 Apple iOS Developer accounts 52
 apps, App Studio 9
 audio 27
 AVE-Doc 7, 14
 AVE-Mag 7, 13, 40

B

Blio assets 41
 bookstore template 12
 bundle IDs 59
 buttons 28

C

certificates 52
 collections 9

E

embedded content 29
 embedded issues 10

G

guided reading 12, 44

H

horizontal orientation 40
 HTML content 29

I

in-app purchases 10, 69
 interactivity actions 35
 iOS Development Certificate 52
 iOS Simulator 47, 63
 ipa files 63
 iPad apps 50
 issue types 9
 issues 9

L

layout families 8
 layout size 8
 layout slideshows 21

M

manifests 45
 markers 33
 menus 44
 movies 25

N

navigation bar 44

O

orientations 7, 40

P

page stacks 7, 39, 43
PDF content 29
pictures, adding 16
pop-ups 37
pricing for issues 69
products 10
Provisioning Profiles 56, 57

S

screen shots 59
scrollable layouts 31
sections 39
shared content 8
slideshows 17
spreads 7
subscriptions 10

T

testing 47
titles 9

U

UDIDs 52, 53
usage 41

V

vertical orientation 40

W

Windows 46, 50

Z

zoom 44
zooming 7