QuarkXPress 2027

Brilliant content that works

quark.com

Digital Publishing with QuarkXPress 2027

November 2026

Contents

Introduction 1

Quark Digital Publishing Options 2

Defining your Goals and Limitations 4

Characterizing your content 4

Assessing your customers' needs 5

Planning for distribution 6

Assessing your infrastructure needs 6

Hardware, Operating Systems, and Formats 8

Hardware 8

Operating systems 9

Multimedia format limitations 9

Delivery formats 9

General Design Approach 11

Mistakes to Avoid 13

Projects and Layouts 15

Digital Layouts 15

Fixed Layouts 15

Flex Layouts 15

Fixed Layout 16

Creating a fixed layout 16

Custom device 19

Working with layouts 21

Layout controls 22

Adaptive scaling 23

Adding interactivity to digital layouts 25
Adding a 360° image 28
Adding animation 31
Adding audio 34
Adding a button 35
Adding a zoomable picture 36
Adding a scroll zone 38
Adding a slideshow 40
Adding a video 46
Adding a web view 48
Working with interactivity actions 50
Creating a Go to Page action 51
Creating a Play Sound File action 52
Creating a Show Pop-Up action 53
East Asian features in digital layouts 56
Using font fallback 57
Specifying story direction 57
Working with grouped characters 57
Working with rubi text 57
Applying emphasis marks 58
Working with mojigumi sets and classes 58
Working with non-breaking character sets 59
Text in digital layouts 59
Fonts in digital layouts 60
Hyperlinks in digital layouts 60
Groups in digital layouts 60
Synchronizing content between orientations 60
Updating missing files 61
Reviewing digital publishing asset usage 61
Default Color Profile for Digital Layouts 61

Working with Reflow 62
Creating reflow articles 63
Creating a reflow article from a selection 64
Creating a reflow article from pages 64
Adding content to a reflow article 65
Reordering components in a reflow article 65
Reflow style sheet mapping 65
Flex Layout 66
Creating a Flex Layout 67
Working with Flex Layouts 68
Flex Container Tool 68
Flex Layout Blocks 69
Flex Page Builder 71
Flex Object Browser 73
Flex Object Browser Enhancements in Flex Layouts 74
HTML Interactive Palette 75
Support for Page Name and Page Title for HTML5 Output in Flex Layouts 77
Find/Change Enhancement in Flex Layout 78
Hyperlinks Support 79
Interactivity Enhancements 79
Picture Zoom Interactivity in Flex Layouts 79
Slideshow Interactivity in Flex Layouts 81
Usability Enhancements 85
Improvements in Paste 85
Changing the Flex Item Order 85
Adding Layout Blocks by double-clicking them 85
Support for Animated GIF 85
Support for Alternate Text for Images in Fixed and Flex Layouts 88
Creating a TOC for ePub 89
Content based TOC in Reflow e-Pubs 89

Working with metadata 92
Support for Semantic Headers 93

Working with Output Styles 97
HTML5 Publication output styles 98

HTML5 Publications 101
Exporting as HTML5 Publication 101
Reader Controls 103
Web app configuration 104
reader-config.json 105
General app configuration 105
Features configuration 106
Social sharing configuration 108
Analytics configuration 109
Navigation configuration 110
View configuration 110
Privacy settings configuration 111
ui-config.css 112
manifest.json 112
Cookie consent and privacy policy configuration 113
Web app capabilities 114
Autofit and Autoscale HTML5 Publication 114
Zoom in/out page on mobile devices 114
Web app performance 114
Web app available for offline reading 115
Hosting an HTML5 package on a web server 115
An example for deploying on macOS 115
Enable Apache web sharing on Mac OS X 115
Test to see if the web server is running 115

Web share the digital output 116
View the digital output 116

Exporting for ePub **117**

Introduction

Digital publishing is a huge publishing opportunity. In fact, maybe a little too huge. There are so many options - and so many uncertainties — that decision-making can be a bit daunting. To make things even more complex, the answer to most of the questions you're likely to ask is probably, “It depends.”

The purpose of this Guide is to help you to make good decisions about digital publishing, so that your efforts will meet with success.

➡ This guide is for Fixed Digital Layouts. Please refer to Flex Layouts Guide for information about Flex layouts.

Quark Digital Publishing Options

The purpose of this topic is to present the digital publishing options available through Quark® for quick reference.

QuarkXPress allows you to configure and export items for digital publishing in ePub and HTML5 Publication output formats.

The ePub file is compatible with all prominent ePub Readers and Google Chrome. You can also export HTML5 Publications that can be viewed using Internet Explorer® (11 or later), Safari®, Google Chrome®, Microsoft Edge, and Firefox® (27 or later) and Opera.

You may want to refer back to this table as you read about the various aspects of digital publishing referred to in the rest of the document.

Media Type	ePub	HTML5 Publications
Description	A standard, book-centric format supported by many readers	A standard, HTML5 based format supported by all major browsers
Best for	Books	Online brochures, catalogs, magazines and any other web content
Reader	Nook, Google, Apple iBooks, Browser extension, Sigil, Calibre	Supported browsers are: Chrome, Safari, Edge, Firefox, and Opera.
Platforms	Multiple	Multiple
Layouts	Built into reader application (for reflowable ePub) or from QuarkXPress (for fixed-layout ePub)	From QuarkXPress
Content	Text, pictures, video, slideshows, HTML interactivity, and more	Text, pictures, video, slideshows, HTML interactivity and more
Distribution	Third party	Self hosted server
Available for QuarkXPress?	QuarkXPress 9.0 and later	QuarkXPress 2016 and later
Available for QPS?	QPS 9.0 and later	—-

Media Type	ePub	HTML5 Publications
Cost	No cost for authoring; distribution arrangement varies by distributor	No cost for authoring; hosting the content may incur some cost

Defining your Goals and Limitations

When you think about digital publishing, it's easy to assume that your first questions should be:

- Which devices do I want to publish to?
- What format do I want to publish in?
- Which tools do I want to use?

However, this may not be the best place to start. Different devices, formats, and tools have different strengths and weaknesses, and it's difficult to choose from among them unless you have a clear picture of your goals, needs, and limitations. The topics below can help you to get a better fix on these, so you can make an intelligent decision when it comes time to choose a direction.

Characterizing your content

To some extent, the choices you make about devices and format should depend on the nature of your content. Is your content highly designed, like a magazine or a brochure? Is it primarily textual, with just a few pictures? Is it designed primarily for visual impact, or for leisurely reading? The answers to questions like these can help you decide whether you need layout view, or reflow view, or both.

- A fixed layout view lets you carefully design pages for maximum visual impact, and allows you to include multimedia content such as sound and movies. A layout view also lets you maintain your branding to the greatest degree possible, because it provides maximum flexibility.
- A reflow view displays text and pictures only, usually with a user-controllable text size and font. A reflow view is less flexible than a layout view, but may be desirable because it makes it easy for the customer to read the text without distraction. Reflow view is well-suited to traditional books and reference materials.

If your content is designed to make a strong visual impression, if you need to maintain your branding, and if you want to include multimedia components, you might want to consider creating a fixed layout ePub or an HTML5 Publication.

If your content is primarily textual, and you don't have any need for fancy layout, you might want to consider using a format such as reflow ePub eBook. This format provides only a reflow view, and is compatible with a wide variety of readers.

If it is critical that your content be presented in a way that is uniquely yours, you can create a custom app which is the most labor intensive.

Another question you may want to ask yourself is, do you want or need to support multiple device orientations? Do your customers expect this functionality? If this is important to you, you should probably consider this when choosing a format. Many ePub readers support this functionality automatically, but not all of them.

If you're considering using multimedia elements such as audio and video, make sure you have the resources necessary to produce those multimedia assets. It's easy to get excited about the idea of using multimedia, but creating and using multimedia properly can involve a great deal of planning and expense. You may be able to reuse materials that your organization has already created for other media, such as the organization's Website.

Some other things to think about:

- Do you want to provide your users with industry-standard navigation controls and icons that they can use without having to think about it too hard, or do you want to create custom controls that provide a more unique experience?
- Do you plan to deliver the same content on the Web? If so, what (if anything) should differentiate the Web version of the content from the version delivered on handheld devices? If there are no meaningful differences between the presentation of content on your Website and the presentation of that content on a handheld device, you may want to consider whether you need an app experience at all.
- Do you need to publish in multiple formats, or only in an app presentation? Are you porting content from another format? Can you do a single-source approach, at least to generate the initial version of a layout that you can then customize for release as an app issue?

For more information about the formats discussed here, see "*Hardware, Operating Systems, and Formats*".

Assessing your customers' needs

It's probably safe to assume that you want your customers to have a positive experience with your content. To make sure this happens, consider questions such as the following:

- Do your customers want to be able to change the point size of the text? If so, consider a format that includes a reflow view.
- How quickly will your customers want to be able to download your content? Even on an unrestricted network, you might want to think about keeping the size of your files as small as possible, in order to prevent your customers from becoming frustrated with the wait.

- In what languages do your customers want to read your content? If you want to distribute content in multiple languages, you should make sure that the format and device you choose support those languages. For example, some East Asian and Eastern European languages use characters that are not available in many Western fonts, and some languages are read in different directions than Western languages. The ePub format supports East Asian and Eastern European languages on some devices, but it may not support them on all devices.
- Do you expect your customers to have an Internet connection when they're consuming the content? If so, you should make sure your content behaves gracefully when customers don't have access to a network. If you create HTML5 interactivity, you can provide a default or "fallback" image or other graphic element in each spot where you display content live via the Web.

For more information about the formats discussed here, see "*Hardware, Operating Systems, and Formats*."

Planning for distribution

Just as there are different ways to distribute print content, there are different ways to distribute content for handheld devices. The way in which you choose to distribute your content probably depends on what that content is for and how you plan to generate revenue.

Consider how often you plan to release content:

- Do you need to distribute small amounts of content on a frequent basis? If so, you might want to consider publishing content on News reader apps like Apple News, Flipboard, etc. This approach gives you continuous publishing, and revenue can come from both advertisements and the content you publish.
- Do you want to publish content for sale on an infrequent or irregular basis? For example, are you a publisher who wants to make book titles available to your customers? If this is the case, you may want to consider using the ePub format. If you have no need of layout view, you might consider a large existing distribution network such as Amazon or Apple iBooks, or one of many smaller distribution networks.

Assessing your infrastructure needs

Creating content for digital distribution is only one part of the digital publishing process. If you plan to deliver content for digital devices, you need to make sure that you have all the infrastructure necessary to do so.

For example, if you plan to host content on a Website, you need to consider how much space you're planning to need and how many downloads you anticipate having, because both of these typically affect the amount you are charged for hosting. In any case, you should track your usage information once you have launched, analyze it, and make any necessary adjustments to your hosting plan. (If you create an app that includes content served over the Web, you should also consider the hosting plan used for that content.)

Billing is another important aspect to consider. If you are selling individual apps or issues, you need to make sure you have the infrastructure necessary to take payments for that content in every market where you want to make that content available. Probably the easiest way to do this is to use an established network such as the App Store or Amazon, because these organizations are designed to make this aspect of business easy and transparent for content creators.

Finally, you should probably have a plan for promoting and marketing your content. You can hire an expert to handle this for you, or promote your content yourself using social media such as Facebook and Twitter, or you can do a combination of both.

Hardware, Operating Systems, and Formats

When you're thinking about publishing to handheld devices, it's good to know as much as possible about the hardware, operating systems, and formats currently available for such devices.

Hardware

It's important to know which types of hardware you want to make your content available on because different devices have different advantages and disadvantages.

For example:

- Physical size: If you're designing content for delivery on a smaller screen, you probably want to design it differently than you would for a larger screen. When designing for a large screen, you might augment the table of contents with thumbnail previews for each item — but on a smaller screen, you might want to keep thumbnail images to a minimum. On smaller screens, you might also want to give serious thought to providing access to all of your content in reflow view.
- Resolution: When designing on a computer monitor, keep in mind that the resolution you're designing at might be different from the resolution your content will be viewed at. If the resolution of your monitor is significantly higher or lower than the resolution of the device you're designing for, you need to keep that difference in mind during design. It also means you should do usability testing with your content early on, to avoid committing yourself to a design that may be unreadable when viewed on the target device.
- Network access: Does the device you're designing for have constant access to the Internet? Or is it designed to download content only from a proprietary server? If the device has constant access to the network, is it only at a limited speed or limited file size? You need to know the answers to these questions if you're planning to provide on-the-fly downloadable content such as updates or magazine issues.
- User interface: Selecting and zooming are a major part of the experience of consuming content on a digital device. Does the device you're designing for have a touch screen? If so, what gestures does it support? If you're designing for multiple devices, do you know what kind of gestures they all support? Does every device you want to use support multiple orientations?
- Orientation switching: Most devices support orientation switching, so regardless of the target platform, you'll probably have to think about doing two different layouts.

Operating systems

A sad fact of designing for digital devices is this: different operating systems require different apps. If you want to make content available natively on as many phones as possible, you may need to find or create an e-reader for iOS, Android, Windows OS, and webOS. Even the same OS can work differently on different devices (for example, different devices run different "flavors" of Android).

Fortunately, there are several ways to avoid having to write a native app for each operating system. For example:

- You can deliver content using a special version of your Website that is designed for use on mobile devices.
- You can deliver content using a generic platform that delivers players for different platforms. For example, you can view Flash content on any platform for which there is a Flash player. However, you cannot currently view Flash content on iOS, and Flash support is still evolving even on mobile platforms that do support it.
- You can deliver content using a proprietary platform that delivers players for different platforms. For example, HTML5 content is readable on a wide variety of platforms, and offers the ability to include interactive content for multiple platforms, as well as static text. HTML5 Publications from QuarkXPress can be read on variety of mobile and tablet devices including desktop browsers.
- You can deliver content using a generic format that can be read on almost any platform, such as HTML5, ePub or PDF

Multimedia format limitations

When choosing a delivery format, keep in mind that not all multimedia files are supported on all platforms. For example, video must be encoded to H.264 format in order to be viewed on iOS, and support for Flash is currently unavailable on iOS devices and limited on many other devices.

Delivery formats

There are several types of formats for delivering content to handheld devices:

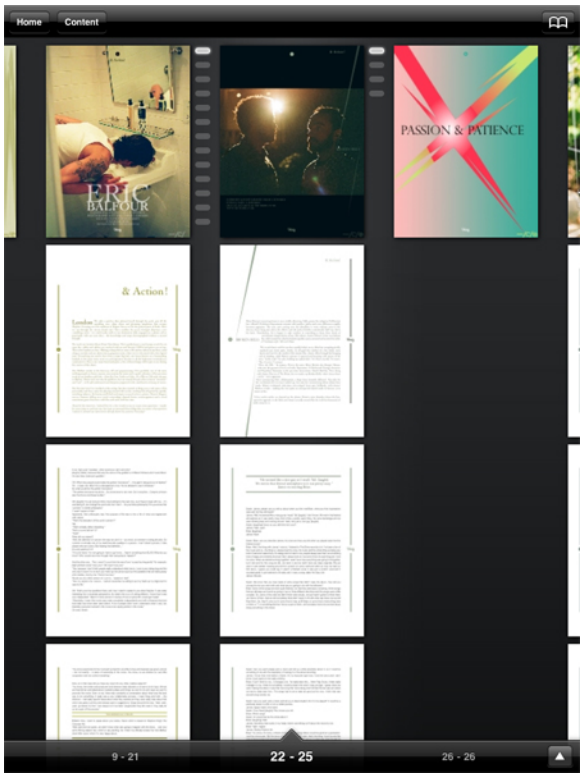
- Open formats such as HTML5, ePub, and PDF. Formats such as these can be attractive because of their ability to display on almost any device and operating system. However, you should keep in mind that it can be difficult to monetize content in an open format that has no established framework for control and distribution.

- Controlled, proprietary formats that can run on multiple devices and operating systems. Formats such as these often offer the advantage of a dedicated distribution system, which can make it easier for a publisher to reach a large audience without giving content away for free.

General Design Approach

Before you start designing anything, you should know as much as possible about the answers to the questions asked earlier in this document. The features and limitations of the format(s), operating system(s), and device(s) you're targeting may severely limit your choices when it comes to design. If you feel like you already have a good understanding of what you want to create, here are some suggestions on how to approach creating it.

- Decide upon a general approach. Avoid reinventing the wheel. Try to find an app/reader that looks and works the way you want it to, then think about what it is that you like about it and how you might improve on it.
- Decide how you want the UI to work. Make sure the UI feels natural. In other words, make sure the design makes affordances for the UI, rather than just imposing an arbitrary UI on every page.
- Decide what kind of layout you want. A standard way to do an eBook is to simply mimic the experience of reading a physical book, with pages that you can turn. However, this isn't the only way to lay out an eBook. For example, you can present a horizontal line of thumbnails, each of which represents an article, with a scrollable vertical list of page thumbnails underneath each (this is called a "page stack"). In full-page view, a page-stack design lets the user swipe horizontally to browse through articles and vertically to read them.



A page stack

- Build and test your templates. If you're templating multiple publications, make sure you think about orientation during the templating phase and do plenty of testing before you start using those templates in production. Test your fonts, colors, styling, and UI on all the devices you plan to publish to. If you're converting content that was designed for another medium, look for problems with very small elements such as narrow and hairline rules, frames, small text, etc.
- Build your content in an organized way. Decide on naming conventions for styles, colors, objects, and so forth, and stick to them. Consider putting all of your interactive items on their own layer so that you can easily show, hide, and isolate them.
- Save the reflow view for last. Make sure your visual layout is FINAL before you start working on reflow view.
- Test, test, test. It's worth a little tedium to avoid embarrassing yourself and having to re-release apps and content. Make sure you test with someone other than the people who designed the content.

Mistakes to Avoid

Digital publishing is an extremely flexible medium. This flexibility means it's possible to do amazing things, but it also means there are pitfalls that you may not be aware of.

For example, just because you can use multimedia elements such as movies, slideshows, and audio in a digital publication, that doesn't mean you should do so on every page. Generally speaking, multimedia elements should be used only when they introduce value that could not otherwise be harvested from the content. An audio sample in an article about a band might be appropriate, where an audio sample from an interview with a dull-sounding subject might not.

When you do use multimedia elements, make sure the control of those multimedia elements is squarely in the hands of your users. A movie that plays when users first view a page might be fine in a publication where the user is unlikely to visit that page again, but could become tiresome if the user has to sit through the movie each time he or she returns to the page. Instead of having such multimedia elements start automatically, consider providing clear visual indicators of their presence and allowing the user to decide whether he/she wants to view and/or listen to them. Just as importantly, make sure the user always has an obvious way to stop the playback of a multimedia element, in case he or she decides against viewing or listening to it.

In situations where you provide visual indicators of the presence of multimedia elements, it may be tempting to get creative, but your users will probably be happiest if you use industry-standard visual cues. For example, if you want to indicate that a box contains a video snippet, a right-facing triangle in the center of the window is probably the clearest way you can indicate this. Similarly, a horizontal bar with a play button is the preferable way to indicate the presence of an audio clip.

It's also important to adhere to industry standards when it comes to navigation. As of this writing, the standard way to navigate from one page to the next on a handheld device is to use a horizontal swipe of the finger. On multi-touch screens, the current standard for zooming in and out is to use pinch-in and pinch-out gestures. On devices that do not support such gestures, arrow icons work well for page turning, and magnifying glass icons work well for zooming. (This may seem obvious, but it's still important enough to mention.)

If you want to create a high-end digital publication, one of the most important mistakes you might want to avoid is creating something that looks like it was created by mindlessly shoving content into a template. There's nothing wrong with starting from a template, but if you don't use some design features that are offered by high-end tools such as QuarkXPress, you may be leaving money on the table.

Lastly, always make sure you test your digital publications on inexperienced users before releasing them. This is a great way to reveal design flaws that otherwise might not be obvious. For example, it's easy to design a page that includes full-screen multimedia elements that can prevent the user from being able to turn the page.

Projects and Layouts

Each QuarkXPress project contains one or more layouts. Every layout is stored within a project, and every project contains at least one layout. Each layout can contain as many as 10,000 pages, and can be as large as 224" x 224" in size (or 112" x 224" for a two-page spread). A project can contain an unlimited number of layouts.

A QuarkXPress project can contain two types of layouts: Print and Digital. This allows you to use one project to create content for various media — such as print, PDF, ePub, and HTML5 Publications.

Digital Layouts

Digital Layouts allow you to create content for digital publishing. You can create content and export the digital layout for digital publishing in ePub and HTML5 Publication output formats.

QuarkXPress supports two types of Digital Layouts:

1. **Fixed Layouts.**
2. **Flex Layouts.**

Fixed Layouts

- Fixed Layouts are developed by keeping in mind the pixel-perfect nature of print designs.
- These are better suited to bring print designs over to the web.

Fixed layouts are designed to create HTML 5 outputs optimized for specific device form factors - such as smartphones and tablets. But because the web can be consumed through different devices of different sizes, the designs that we publish might not work as expected for all devices. Fixed layouts allow, and require, users to create multiple layouts - one for each device, and device orientation.

Flex Layouts

- Flex layouts, on the other hand, are target device-independent layouts.
- The single layout can be designed to target different devices.

Once the layout is designed, it will adjust itself to different device sizes without sacrificing legibility and relevancy of your content.

Learn more about Flex Layouts at:

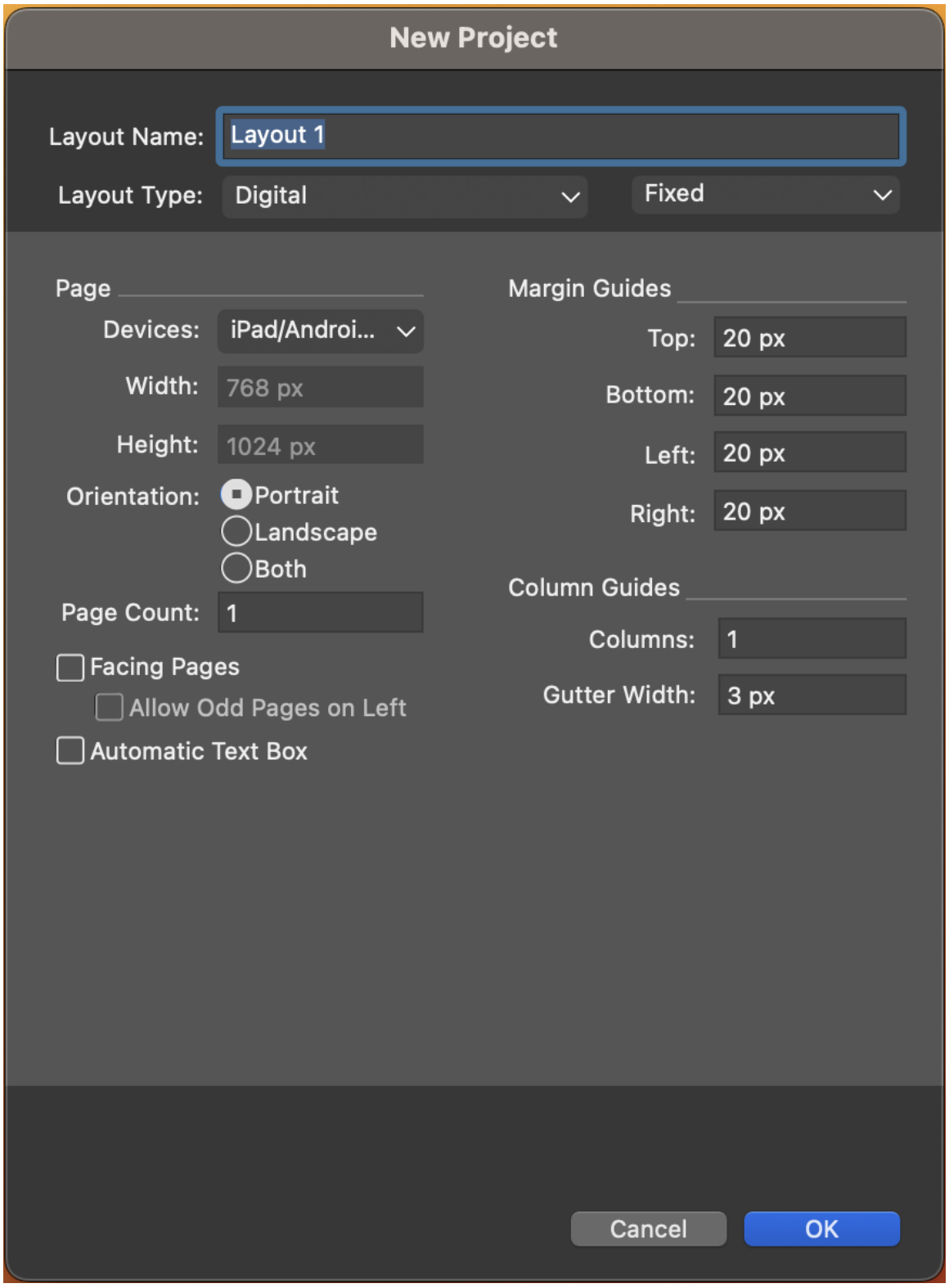
➡ *http://www.quark.com/documentation/quarkxpress/2019/english/HTML5Output/Page1.html*

Fixed Layout

Creating a fixed layout

To create a digital layout:

1 Choose File > New > Project. The **New Project** dialog box displays.



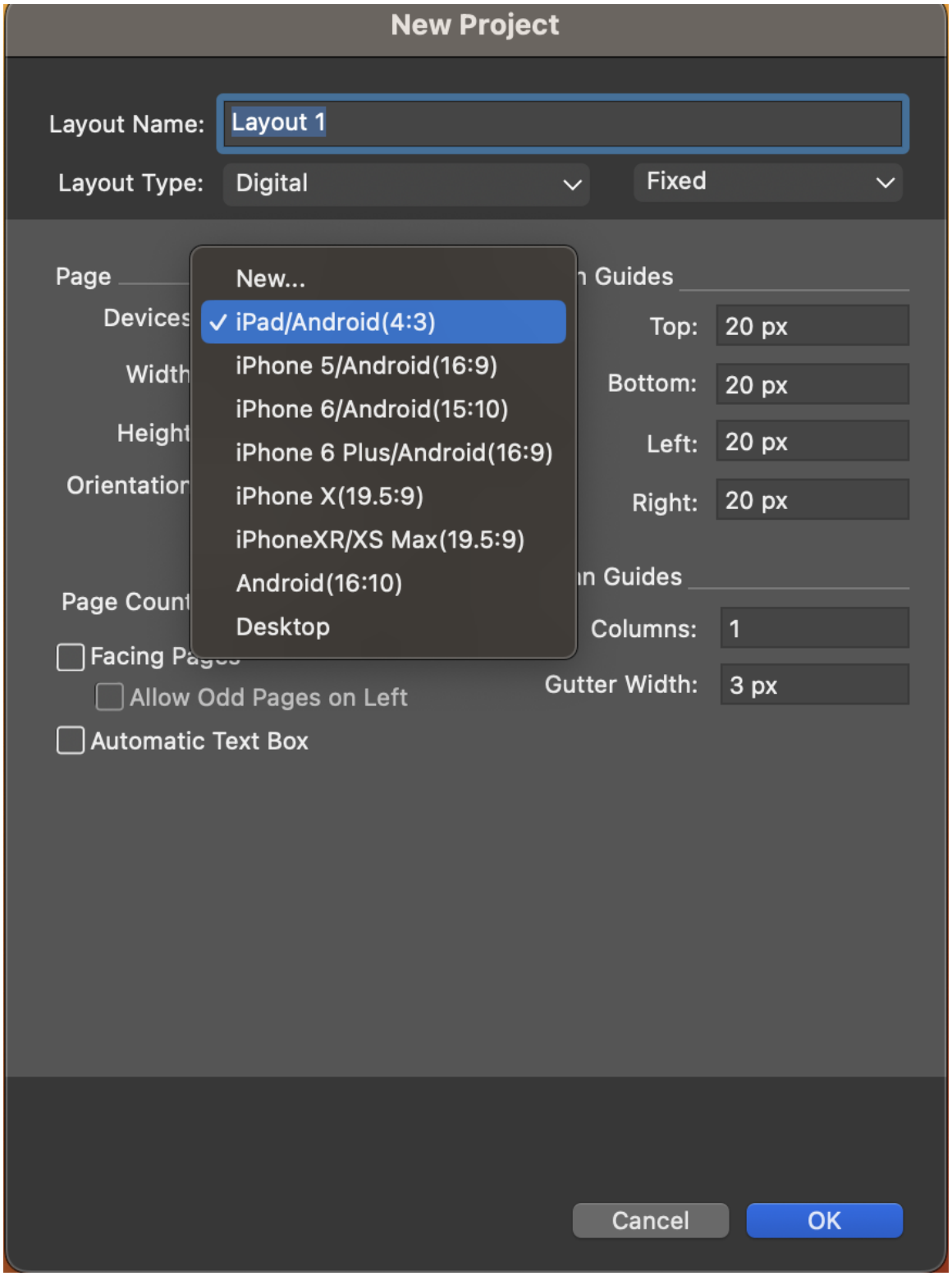
New Project dialog for a Digital layout

2 Enter a name for the layout in the **Layout Name** field.

➡ QuarkXPress will append a "(V)" or "(H)" to the name, based on the layout orientation you choose.

3 Choose **Digital** from the **Layout Type** drop-down menu.

4 Choose a device from the **Devices** drop-down menu.



Devices drop-down menu

The **Width** and **Height** fields are automatically populated depending on the device chosen.

➡ To define and use a custom device see "*Custom device*."

5 Set the orientation for the default layout.

If you select **Both**, QuarkXPress will automatically create two layouts, one with a **Portrait** orientation and one with a **Landscape** orientation. QuarkXPress displays the new project in a split view, so that you can work with both views of the layout at the same time. For more information, see "Splitting a window" in *A Guide to QuarkXPress*.

6 The **Margin Guides** controls let you set default margins for the layout.

7 The **Page Count** allows you to specify the number of pages you want to initially create.

8 The **Column Guides** controls let you create a multi-column page by default.

9 Specify the **Default Story Direction** for the layout.

10 The **Facing Pages** check box lets you create spreads. In a project with a horizontal default story direction, the **Allow Odd Pages On Left** check box lets you control whether you can have odd pages on the left. In a project with a vertical default story direction, the **Allow Odd Pages On Right** check box lets you control whether you can have odd pages on the right.

11 The **Automatic Text Box** check box lets you add a text box to the default master page for the layout.

12 Click **OK**.

A project containing a default Digital layout is created.

Custom device

You can define custom devices with custom page sizes and add it to the Devices drop-down menu on the New Project dialog.

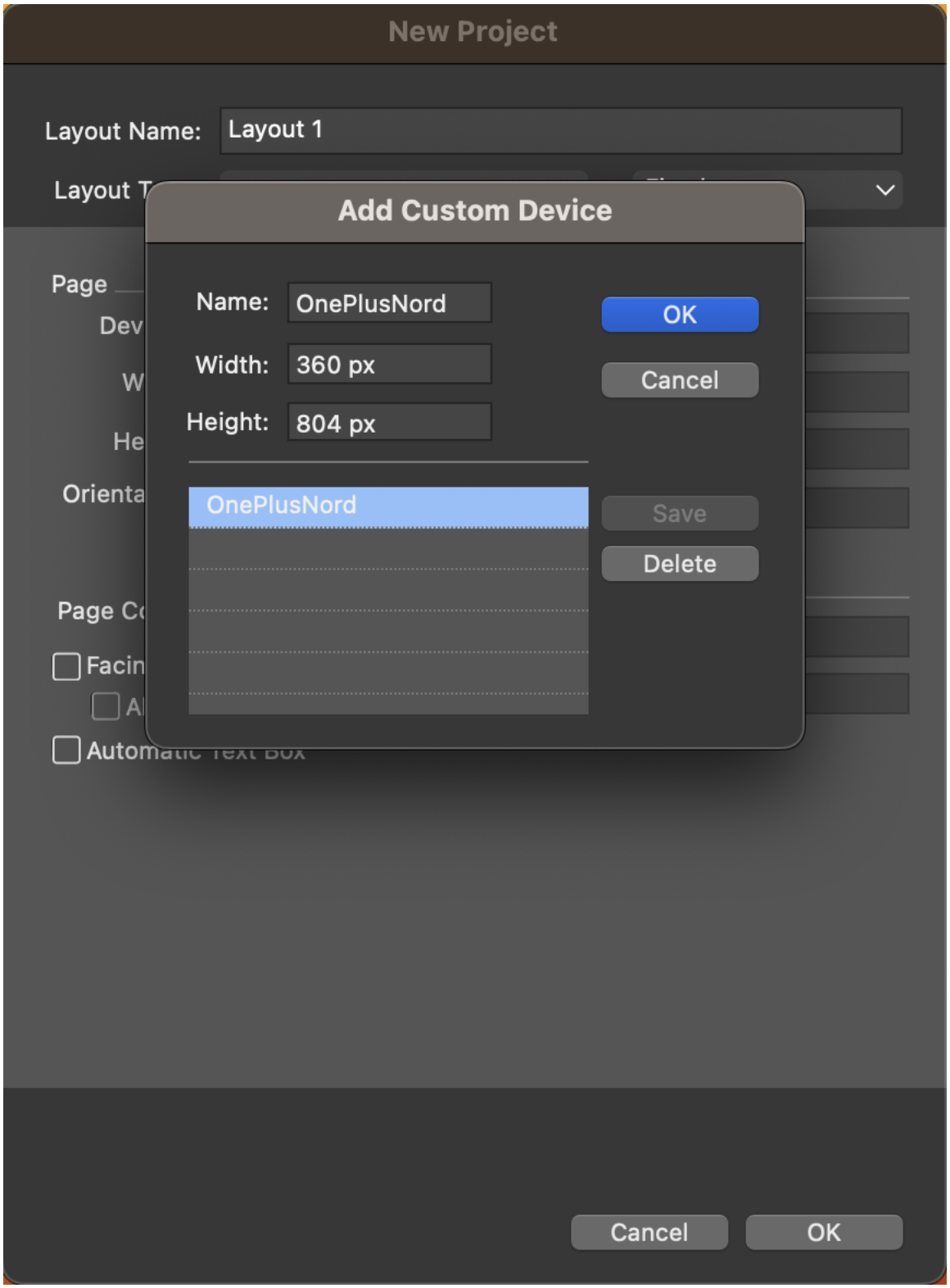
To create a custom device:

1 Choose File > New > Project. The **New Project** dialog displays.

2 Enter a name for the layout in the **Layout Name** field.

3 Choose **Digital** from the **Layout Type** drop-down menu.

4 Choose New from the **Devices** drop-down menu. The **Add Custom Device** dialog displays.



Add Custom Device dialog for a digital layout

5 Specify a **Name** and define the **Width** and **Height** for the custom device.

➡ To create multiple custom devices, click **Add**. The new custom device will appear in the list and you will be able to create another. To delete a custom

device, select it from the list and click **Delete**. To edit a custom device, select it from the list and redefine the **Width** and **Height** fields.

6 Click **OK**.

Any custom devices you create will appear in the **Devices** drop-down menu in the **New Project** and **New Layout** dialogs, only when you are creating a digital layout.

➡ Custom devices can also be created using the **New Layout** dialog. (Layout > New).

Working with layouts

It's easy to navigate between layouts, add layouts, duplicate layouts, and delete layouts.

By default, tabs display at the top of the project window for each of the layouts in a project. On macOS only, you can change the order of layouts displayed in the layout tab by dragging and dropping the layout name tabs. This setting is saved with the document and the order is maintained.

To navigate between layouts, use the tabs at the top of the project window.

To add a layout to the active project, choose Layout > New or right-click a layout tab and choose **New** from the context menu.

To duplicate a layout, display the layout you want to duplicate, then choose Layout > Duplicate or choose **Duplicate** from the **Layout** tab context menu. User can duplicate a **Print** layout into a **Digital** layout by choosing **Digital** as the layout type.

Users can duplicate a layout from **Print** to **Print**, **Print** to **Digital**, **Digital** to **Digital**, and **Digital** to **Print**.

To change a layout's properties, display the layout, then choose Layout > Layout Properties or choose **Layout Properties** from the **Layout** tab context menu. The **Layout Properties** dialog box displays. You can use this dialog box to change limited layout properties based on the layout type, but you cannot change a layout to or from a Digital layout type.

When duplicating a layout, QuarkXPress allows you to scale items on the layout based on a configurable set of rules. This allows the user to get an almost ready to use layout in the new size. See "*Adaptive scaling*."

To delete a layout, display the layout, then choose Layout > Delete or choose **Delete** from the **Layout** tab context menu.

When exporting as PDF, Image, ePub or Project, only the active layout is included in the resulting output. When exporting as HTML5 Publication, you will be allowed to choose the layout you want included in the output.

- File > Export As > PDF
- File > Export As > JPEG/PNG/TIFF
- File > Export As > HTML5 Publication
- File > Export As > ePub

➡ Layers apply to the layout that is active when you create and edit them.

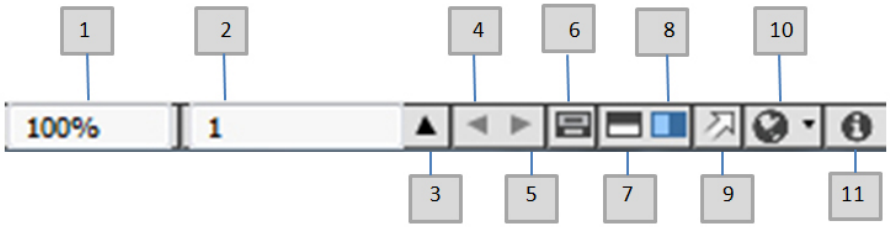
➡ When you perform a project-level action (Edit > Undo), the action is added to the Undo History in all layouts.

➡ When you check spelling (**Utilities** menu), QuarkXPress checks only the active layout.

➡ The Find/Change feature (**Edit** menu) can search only the active layout.

Layout controls

When you open a project, you have immediate access to some basic features at the bottom left of the project window.



Layout controls

1 **Zoom**: Enter a zoom percentage or choose a zoom value from the drop-down menu (the maximum zoom limit is 8000%).

2 **Page Number**: Enter a page number in the **Page Number** field or choose a page from the page list that displays when you click the upward facing arrow to the right of the field.

3 **Page Preview**: Click the up arrow next to the **Page Number** field to see a thumbnail view of all the pages in the layout. macOS users can continue to click the up arrow to enlarge the thumbnails.

4 **Previous Page**: Navigate to the previous page.

5 **Next Page**: Navigate to the next page.

6 **View Master Page**: Switch back and forth between the active layout page and its master page.

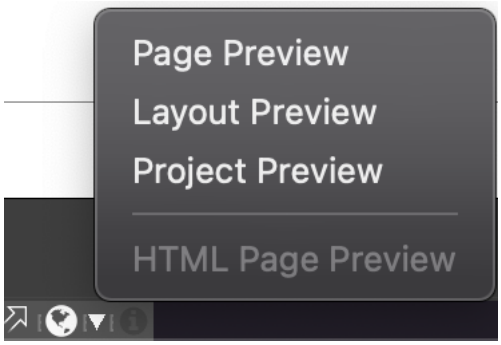
7 **Split Screen Horizontal**: View the layout in two or more separate panes one above the other.

8 **Split Screen Vertical**: View the layout in two or more separate panes side by side.

9 **Export**: Display the same export options that are available when you choose File > Export.

10 **Preview HTML5 Publication**: Generates a preview of the layout as an HTML5 Publication. This allows you preview your current layout in a browser, before actually exporting or uploading to the portal.

Click the arrow next to this icon to choose a preview option:



- Page Preview: Renders the current page.
- Layout Preview: Renders the current layout.
- Project Preview: Renders multiple layout variations, (e.g. iPad and iPhone), so that you can simulate different page width (or aspect ratios) to match the layouts you have created.

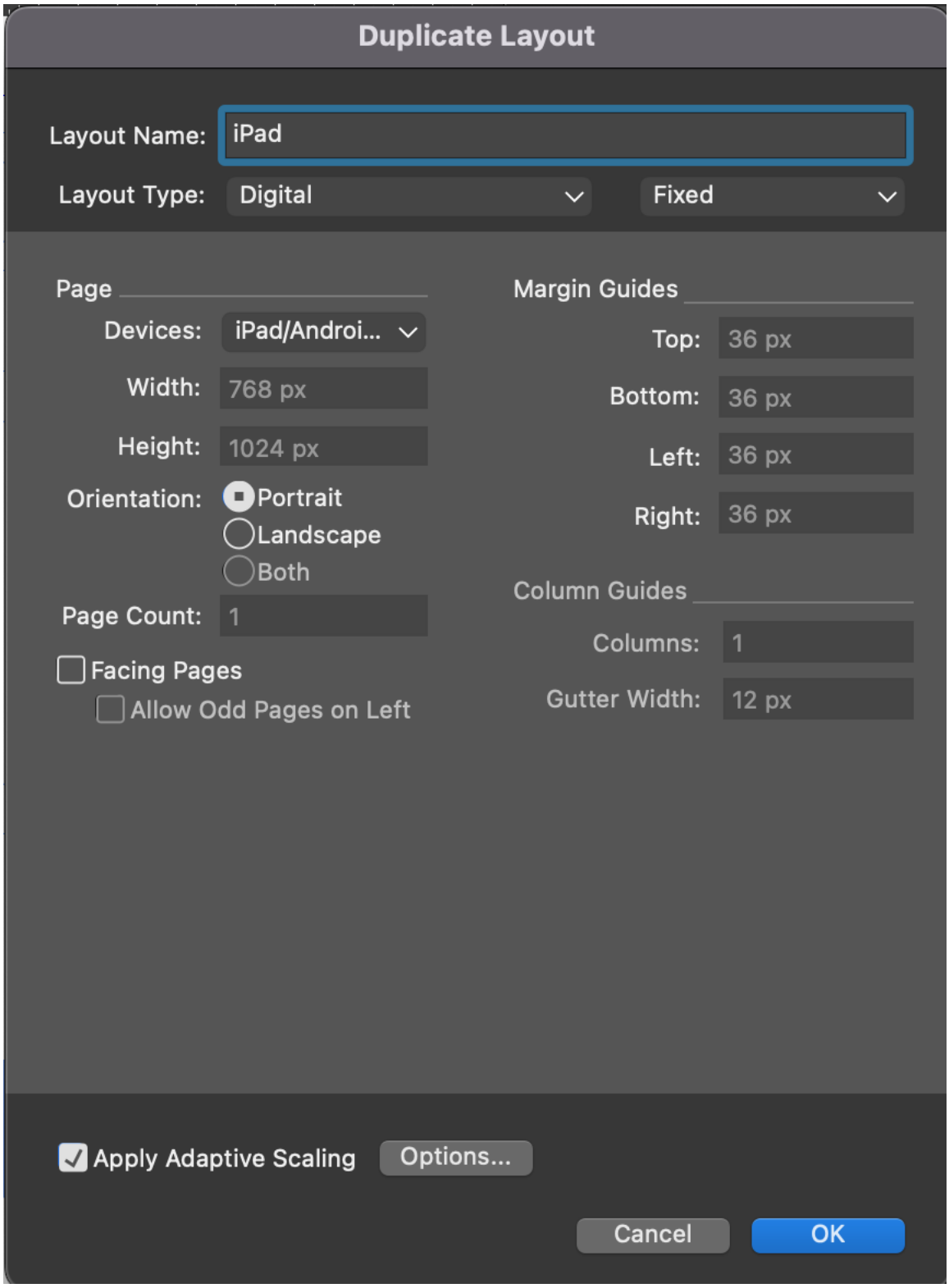
Adaptive scaling

This section describes the **Apply Adaptive Scaling** option available in the **Duplicate Layout** dialogs.

This feature allows users to scale items based on a configurable set of rules, to get an almost “ready to use” layout in the new size.

To apply adaptive scaling:

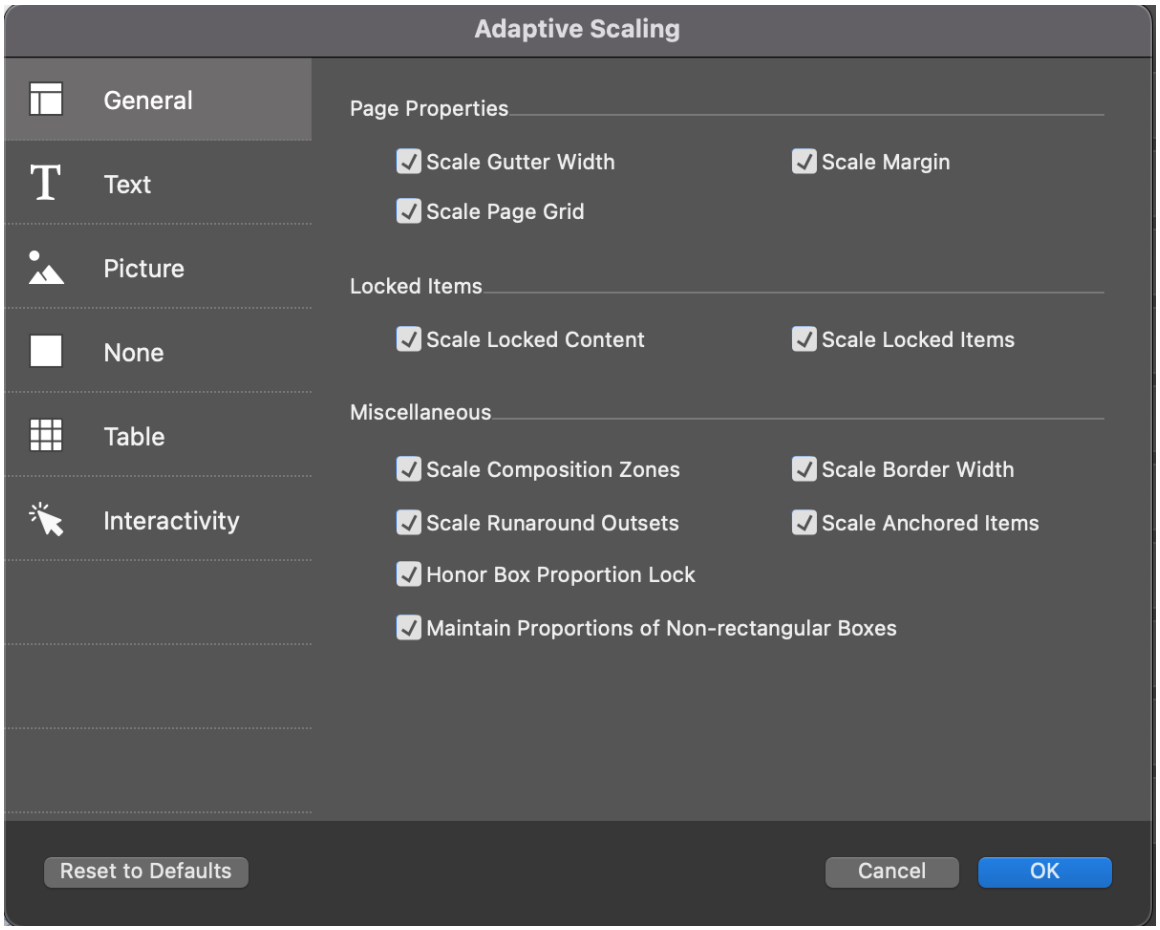
1 Choose Layout > Duplicate. The **Duplicate Layout** dialog displays.



Duplicate Layout dialog

2 Check **Apply Adaptive Scaling**.

3 Click **Options** to configure the rules for scaling item on the layout. The **Adaptive Scaling** dialog displays.



Adaptive Scaling dialog

4 Use the dialog to configure the rules for scaling the items on the layout. The **Adaptive Scaling** dialog allows you to configure rules for the page properties, locked items, text, pictures, boxes, lines, tables, and any interactivity items that may exist on the layout.

5 Click **OK**.

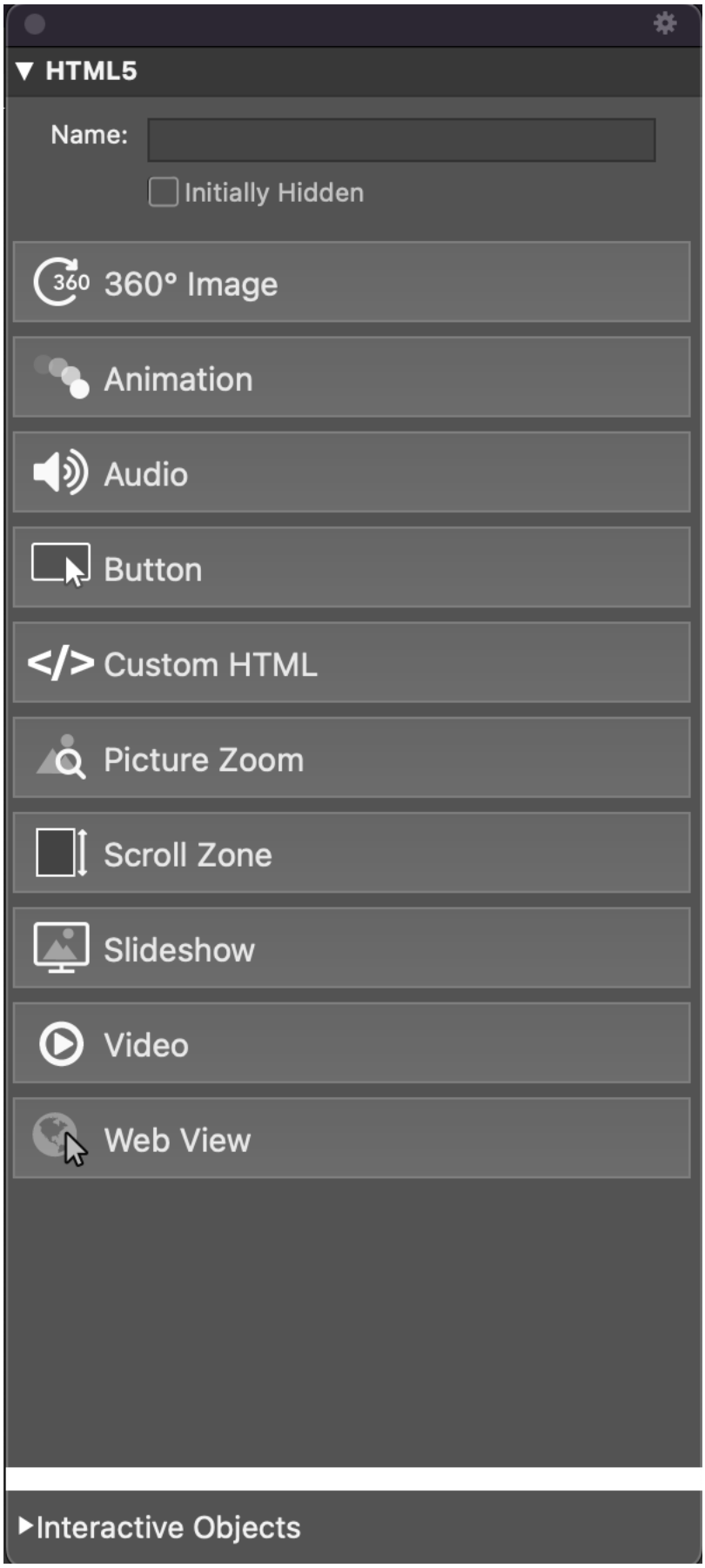
On the **Text** tab, check **HScale**, **VScale** or both to adjust horizontal and vertical text scaling in proportion to the source and destination layout sizes, in the same way as font size.

➡ It is recommended that you create a copy of your file, before using adaptive scaling, especially if you have checked the **Synchronize Box Attributes & Content** option on the **Text**, **Picture**, and **None** tabs of the **Adaptive Scaling** dialog. This option affects the source layout, and the sharing cannot be undone. It adds a large number of items to the **Contents** palette.

Adding interactivity to digital layouts

You can add a variety of types of interactivity to a digital layout, including slideshows, movies, buttons, sound, and HTML with the **HTML5** palette (**Window**

menu). This palette lets you name and add interactivity to whichever object is selected in the layout.


HTML5
Name:
Initially Hidden
360° Image
Animation
Audio
Button
Custom HTML
Picture Zoom
Scroll Zone
Slideshow
Video
Web View
Interactive Objects

HTML5 palette

Picture boxes, text boxes, anchored boxes, and no-content boxes support different kinds of interactivity. Options that are not available for the selected item are disabled.

At the bottom of the palette, there is a list of all the interactive objects in the active layout, including each enrichment type, object's name, and page number. You can navigate to any object listed here by double-clicking it.

Once you apply interactivity to a 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:

- :Named Object
- : 360° Image
- : Animation
- : Audio
- : Button
- :Custom HTML
- : Picture Zoom
- : Scroll Zone
- : Slideshow
- : Video
- : Web View

The **HTML5** palette includes a number of file selection controls. If you are logged into Quark Publishing Platform, you can choose whether to select a file from the file system or from Quark Publishing Platform. If you choose **Platform** from such a button, the **Asset Picker** dialog box lets you choose a Platform asset.

➡ Quark Publishing Platform is a dynamic publishing platform that provides a broad range of functionality to streamline and automate your publishing process. . For general information about the Platform, visit *https://www.quark.com/Products/Quark_Publishing_Platform*.

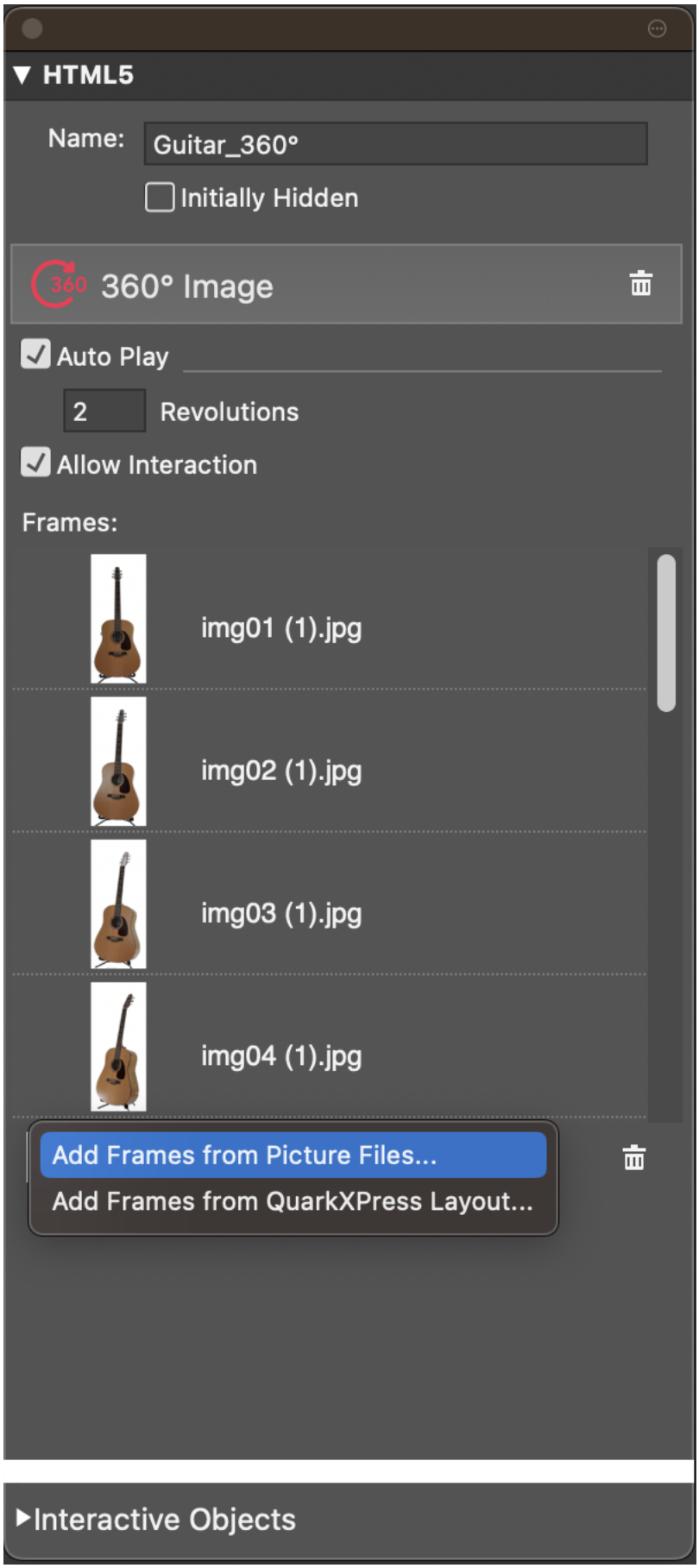
Adding a 360° image

A user can combine a series of images taken at fixed intervals around an object to create a single interactive image. The 360° image can auto rotate, and the user can grab the image and rotate it to any angle.

Digital layouts support interactive pictures in PNG, JPEG, GIF, TIFF, and EPS formats.

To add a 360° image to a Digital layout:

1. Select the picture box you want to contain the 360° image.
2. In the **HTML5** palette, click **360° Image**.


HTML5
Name:
Guitar_360°
Initially Hidden
360° Image
Auto Play
2
Revolutions
Allow Interaction
Frames:
img01 (1).jpg
img02 (1).jpg
img03 (1).jpg
img04 (1).jpg
Add Frames from Picture Files...
Add Frames from QuarkXPress Layout...
Interactive Objects

360 image controls

- **Name**: Optionally, enter a name for the 360° image in the **Name** field. This name displays in the **Interactive Objects** area at the bottom of the palette.
- **Initially Hidden**: If you want the 360° image to be initially invisible, check **Initially Hidden**.
- **Auto Play**: To allow the 360° image to initially rotate automatically, check **Auto Play**.
  - **Revolutions**: Lets you specify how many times the 360° image should automatically rotate.

➡ After the image has completed the number of automatic revolutions you have specified, the user will be allowed to manually rotate the 360° image.

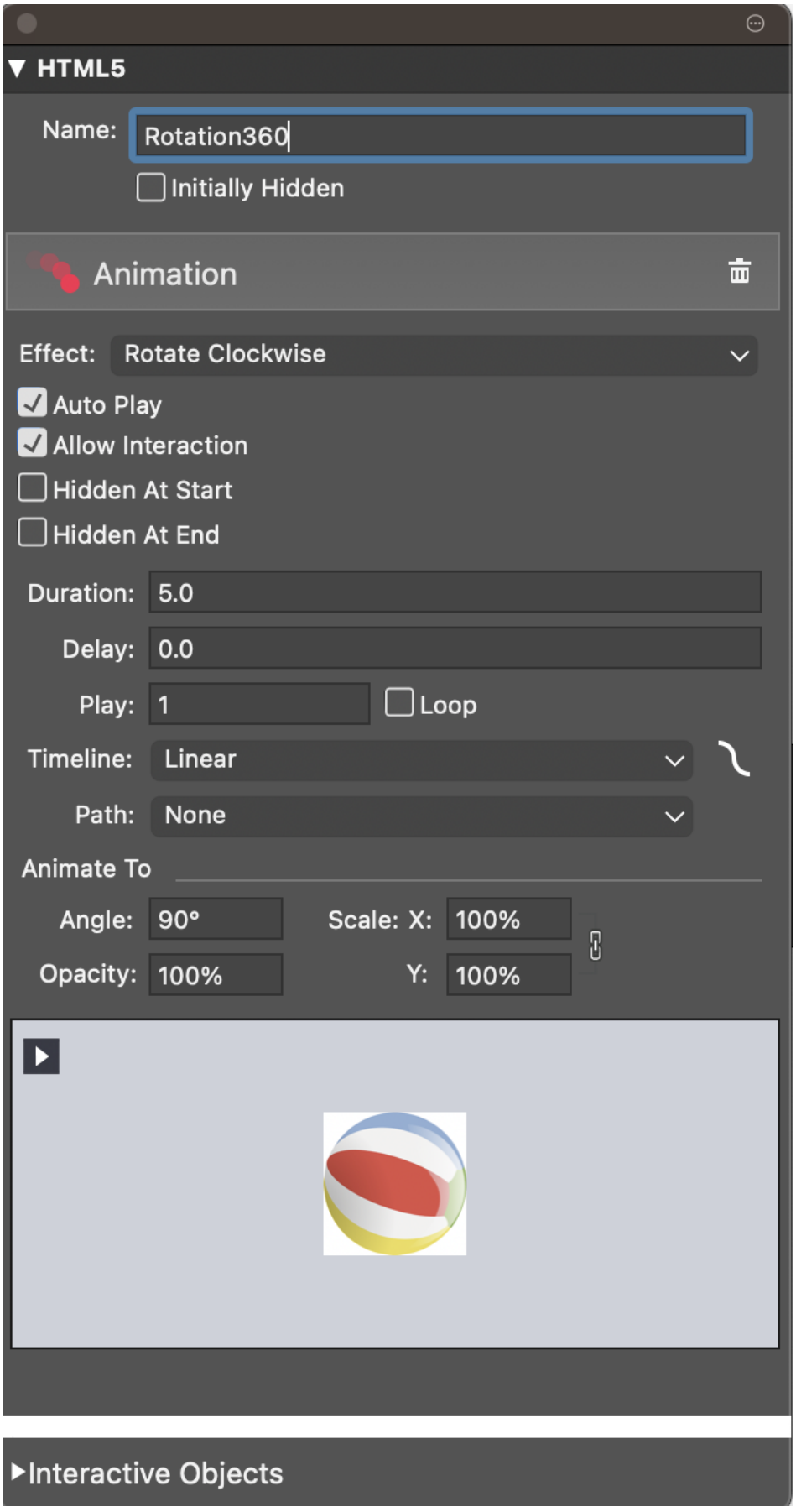
- **Allow Interaction**: If you want the user to be able to rotate slides manually, Check **Allow Interaction**.
- To add frames to the 360° image, click the ✚ button.
  - **Add Frames from Picture Files**: Lets you add picture files as frames to the image. You can use picture files in PNG, JPEG, TIFF, PDF, and EPS formats. You can use the Shift or Command/Ctrl key to select multiple files.
  - **Add Frames from QuarkXPress Layout**: Displays the **Add Frames from Layout** dialog. This dialog lets you add pages from a QuarkXPress layout as separate frame to the image. You can create a new layout or choose pages from an existing layout.
- To edit a single frame, choose a frame and click the **Edit** button.

Adding animation

A user can apply animation effects to objects in their documents.

To add an animation to a Digital layout:

1. Select the object you want to apply the animation to. You can apply animations to picture boxes, text boxes, None Boxes and Lines. You can not apply animation to group boxes.
2. In the **HTML5** palette, click **Animation**.


HTML5
Name:
Rotation360
Initially Hidden
Animation
Effect:
Rotate Clockwise
Auto Play
Allow Interaction
Hidden At Start
Hidden At End
Duration:
5.0
Delay:
0.0
Play:
1
Loop
Timeline:
Linear
Path:
None
Animate To
Angle:
90°
Scale: X:
100%
Opacity:
100%
Y:
100%
Interactive Objects

Animation options

- **Name**: Optionally, enter a name for the animation in the **Name** field. This name displays in the Interactive Objects area in the bottom of the palette.
- **Initially Hidden**: If you want the animation object to be initially invisible, check **Initially Hidden**.
- **Effect**: Choose how you wish the animation to first appear from the **Effect** drop down menu.
- **Auto Play**: To allow the animation to initially play automatically, check **Auto Play**.
- **Allow Interaction**: To allow the viewer to interact with the animation, check **Allow Interaction**.
- **Hidden at Start**: If you want the animation to be initially invisible, check Hidden at Start.
- **Hidden at End**: If you want the animation to become invisible after playing, check **Hidden at End**.
- **Duration**: Lets you specify how much time it takes for the animation to occur.
- **Delay**: Lets you specify how long of a delay there should be before the animation begins to play.
- **Play**: Lets you specify the number of times the animation is played, or check **Loop** to cause the animation to play repeatedly.
- **Timeline**: Choose from the **Timeline** drop down menu to specify the timeline for the animation. For example, you can choose that the animation starts slowly and speeds up (**Ease In**), or slows down at the end (**Ease Out**).
- **Path**: Lets you define and specify the path the animation will take. Only the end point and starting point of an object are valued as a path and the object will travel on a line between the two.
- Set the following options in the **Animate To** section of the palette to determine the objects ending point:
  - **Angle**: Specify a rotation angle that the object completes during the animation.
  - **Opacity**: Specify the opacity the object should have at the end of the animation.
  - **Scale**: X and Scale: Y : Specify a percentage value to determine whether the object increases or decreases in size during playback.

To preview the animation, click the ▶ button.

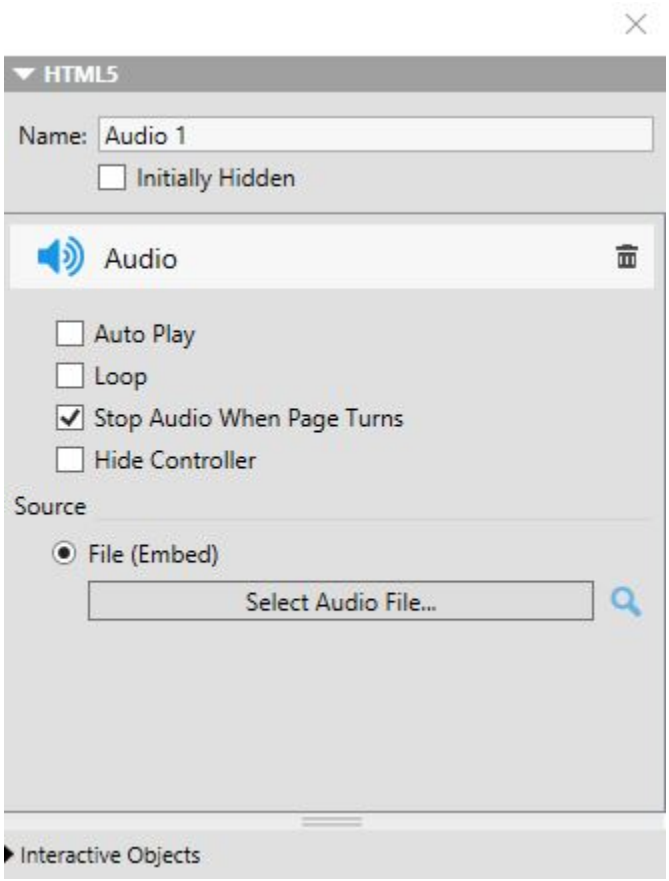
Adding audio

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 also configure sound files to play in the background and to continue playing when the user changes pages.

- You can start and stop the playback of audio files with actions. For more information, see "*Working with interactivity actions*."
- QuarkXPress HTML5 interactivity supports only .mp3 files.
- Only one audio file can play at a time. If one sound is playing, and another sound starts, the first sound is paused.

To add audio to a Digital layout:

1. Select the picture box you want to replace with the audio controls.
2. In the **HTML5** palette, click Audio.



Audio Options

- **Name**: Optionally, enter a name for the audio controller in the **Name** field. This name displays in the I**nteractive Objects** area in the bottom of the palette.

- **Initially Hidden**: If you want the audio object to be initially invisible, check **Initially Hidden**.
- **Auto Play**: To make the audio play automatically when the page it is on displays, check **Auto Play**.
- **Loop**: To make the audio play repeatedly, check **Loop**.
- **Stop Audio When Page Turns**: To make the audio stop playing when the user goes to a different page, check **Stop Audio When Page Turns**. If you leave this box unchecked, the audio will continue to play until the user moves to the next article.

Sounds that play at the article level do not play when you preview the article in a Web browser.

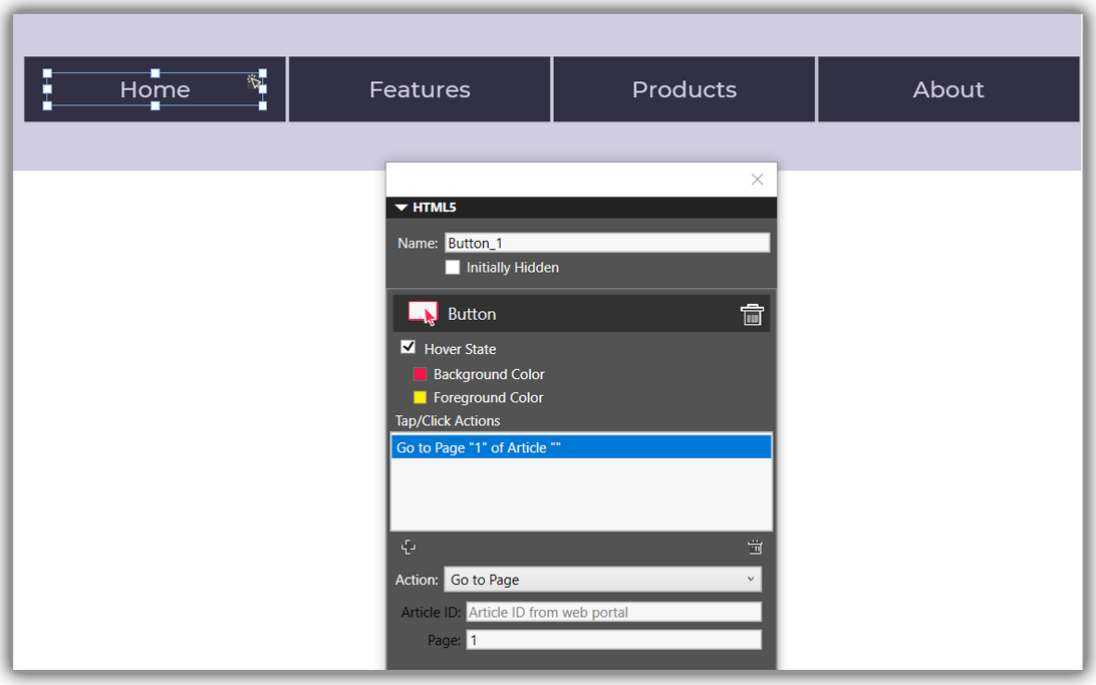
- **Hide Controller**: If **Stop Audio When Page Turns** is checked, you can hide the default audio controls by checking **Hide Controller**.
- To specify the location of the audio file, click an option in the **Source** area.
  - To use a local audio file, click **File (Embed)**, then click the button and select the audio file.
  - To use an audio file from Quark Publishing Platform, click **Quark Publishing Platform**, then click **Browse Platform** and select the audio file.

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

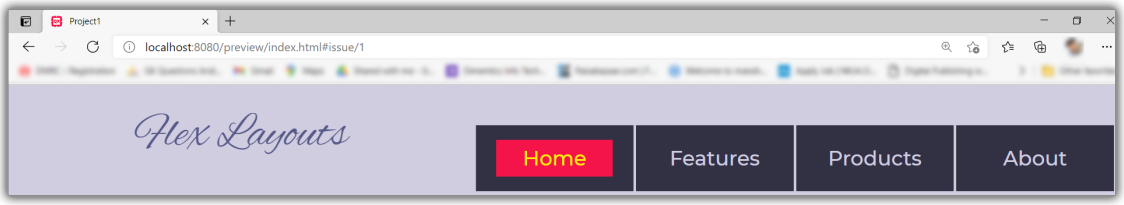
Adding a button

To add a button to a Digital layout:

1. Select the rectangular picture box you want to make into a button.
2. In the **HTML5** palette, click **Button**.



- **Name**: Optionally, enter an internal name for the picture in the **Name** field. This name displays in the **Interactive Objects** area in the bottom of the palette.
- **Initially Hidden**: If you want the button to be initially invisible, check **Initially Hidden**.



- **Action**: To add an action for the button, click the button. Choose an action from the **Action** drop-down menu at the bottom, then use the controls below to configure the action. You can add multiple actions to the same button. For more information, see "*Working with interactivity actions*."

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

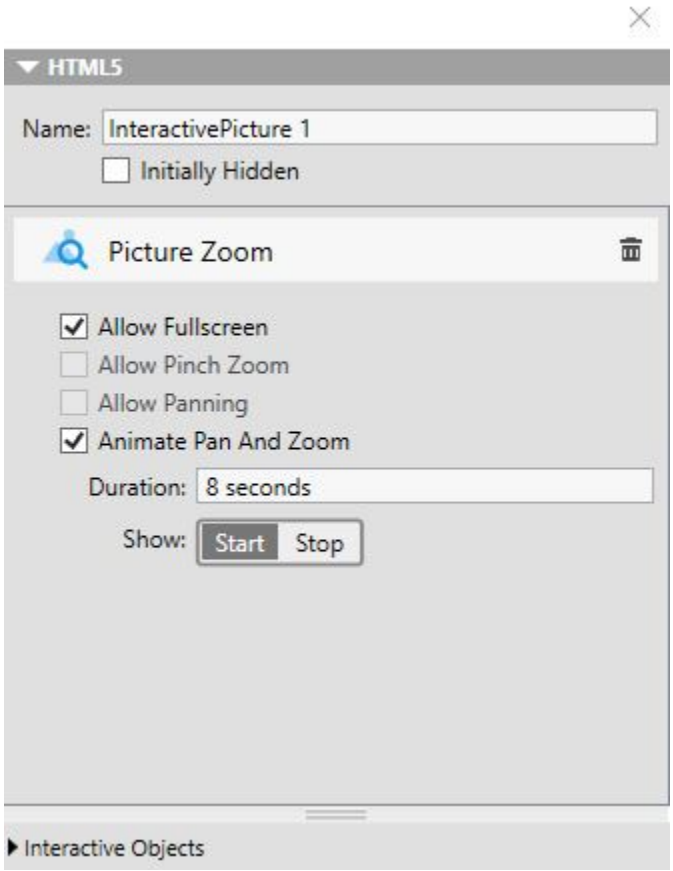
Adding a zoomable picture

A zoomable picture initially displays in a box, but takes over the entire screen when double-tapped. You can use this feature to add an animated pan-and-zoom effect to an image, or allow it to be zoomed and panned directly in its box.

Digital layouts support interactive pictures in PNG, JPEG, GIF, TIFF, and EPS formats.

To add an interactive picture to a Digital layout:

1. Select the picture box that contains the picture you want to make zoomable.
2. In the **HTML5** palette, click **Picture Zoom**.



Picture zoom controls

- **Name**: Optionally, enter a name for the zoomable picture in the **Name** field. This name displays in the **Interactive Objects** area in the bottom of the palette.
- **Initially Hidden**: If you want the zoomable picture to be initially invisible, check **Initially Hidden**.
- **Allow Fullscreen**: To allow the user to switch back and forth between cropped view and fullscreen view by double-tapping the picture, check **Allow Fullscreen**.
- **Allow Pinch Zoom**: To allow the user to zoom in and out with pinch gestures, check **Allow Pinch Zoom**.
- **Allow Panning**: To allow the user to pan around in the picture with one finger, check **Allow Panning**.

- **Animate Pan and Zoom**: To make the picture pan and/or zoom at the beginning of its display, check **Animate Pan and Zoom**.
    - **Duration**: Lets you control how long the pan/zoom lasts. At the end of this duration, the picture stops and stays in its final position.
    - **Start and Stop**: These buttons allow you to set the beginning and ending crop.

Click **Start** and scale/position the picture for the initial position, then click **Stop** and scale/position the image for the final position.

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

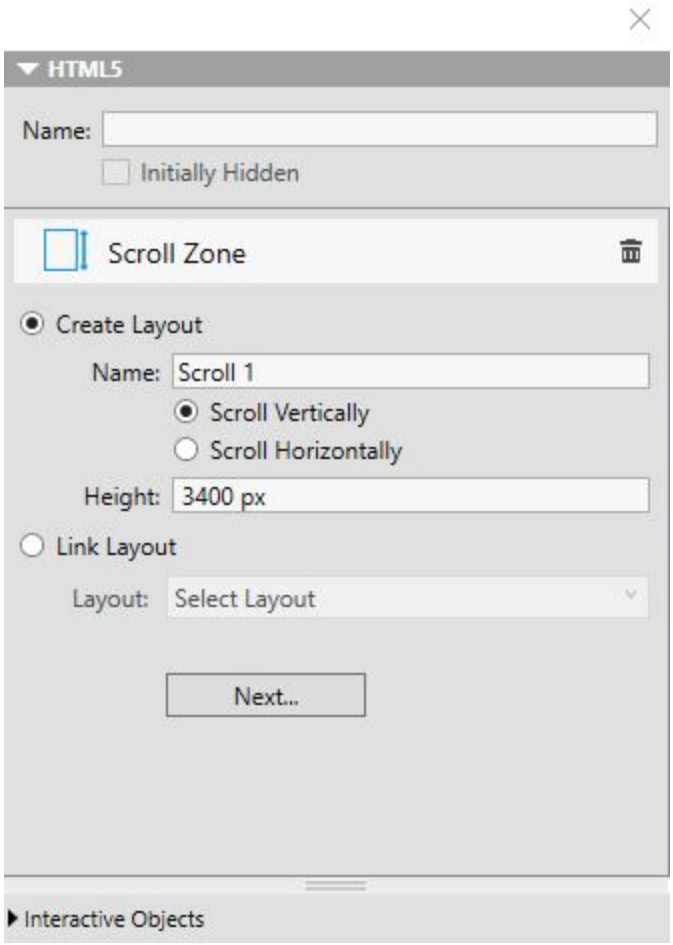
Adding a scroll zone

The scroll zone feature lets you add a scrollable area to a page on a layout. The content for the scrollable area comes from a different layout (the scrollable layout). Once you set up a scroll zone, 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.

- Referenced scrollable layouts are exported as HTML in the same way the host layout is. That means they work the same way in terms of font usage and the **Convert To Graphic on Export** option for text boxes.
- Referenced layout can contain their own interactive objects, and they will work as they would in the host layout.

To set up a scroll zone for a Digital layout:

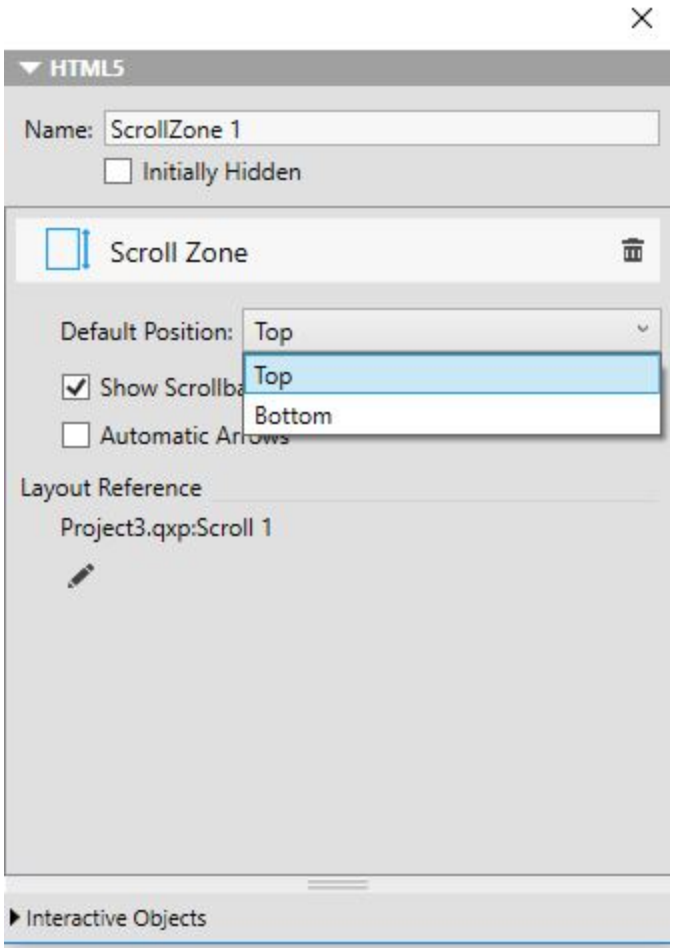
1. Navigate to a layout that is part of a layout family.
2. Draw a picture box to represent the size and the location of the scroll zone. Make sure the box is selected.
3. In the HTML5 palette, click Scroll Zone.



Scroll zone controls

- **Name**: Optionally, enter a name for the scroll zone in the **Name** field. This name displays in the **Interactive Objects** area in the bottom of the palette.
- Choose whether you want to create a layout or link to an existing layout.
  - **Create Layout**: If you want to create a new layout to serve as the scrollable layout, click **Create Layout** and enter a name for the layout in the **Name** field. You can decide whether you want to create a horizontal or vertical scrollbox. A vertical scrollbox can be as long as you want it to be, a horizontal scrollbox is restricted to one page. Enter the height of the scrollable layout in the **Height** field.
  - **Link 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 **Layout** drop-down menu. The size of the selected box is adjusted to fit the layout.

4 Click **Next**. The **HTML5** palette displays the following controls:



Scroll zone layout controls in the HTML5 palette

- **Default Position**: Specify the default position of the scroll bar:
  - Choose **Top** or **Bottom** if scrolling vertically.
  - Choose **Left** or **Right** if scrolling horizontally.
- **Show Scrollbar**: To provide scrollbars as a visual cue that the area is scrollable, check **Show Scrollbar**.
- **Automatic Arrows**: To automatically display arrows that indicate the direction in which the scroll zone can be scrolled, check **Automatic Arrows**.
- To edit the scrollable layout, click the **Edit** button under **Layout Reference**.

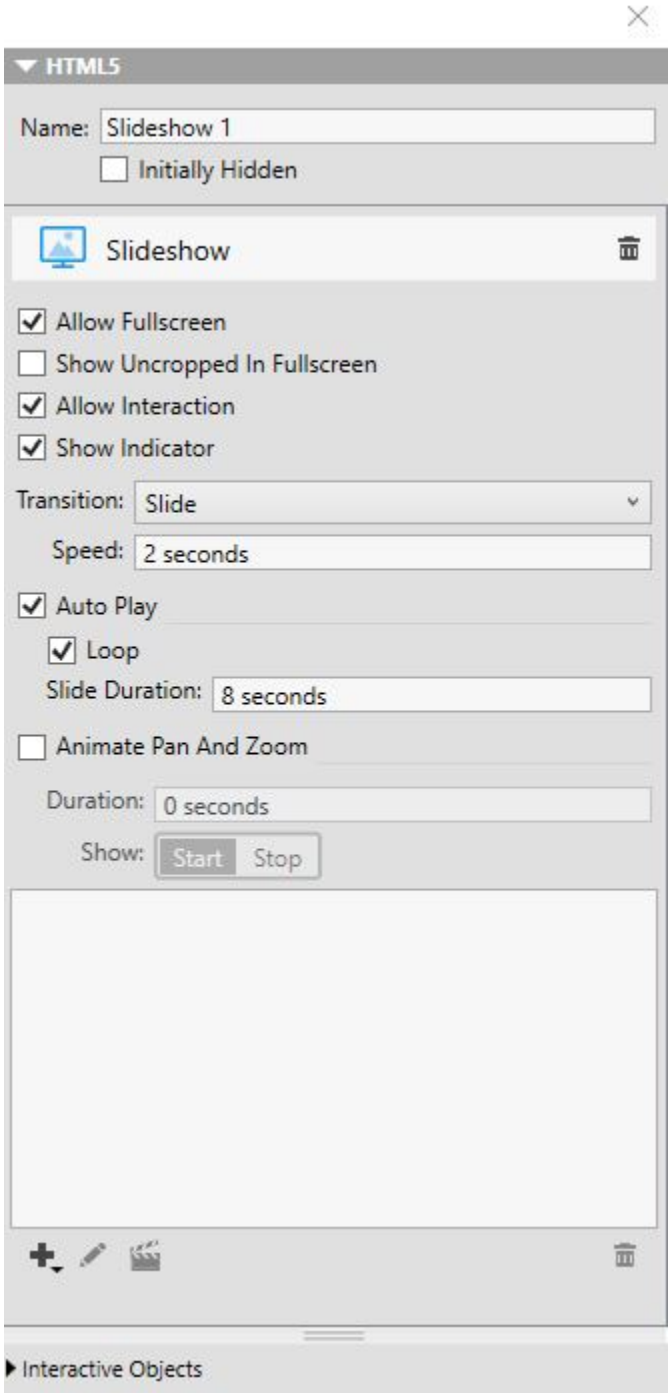
To finish configuring the scroll zone, click a different object or a blank part of the layout.

Adding a slideshow

Slideshows support both picture files and pages from QuarkXPress layouts.

To add a slideshow to a Digital layout:

1. Select the picture box you want to contain the slideshow.
2. In the **HTML5** palette, click **Slideshow**.

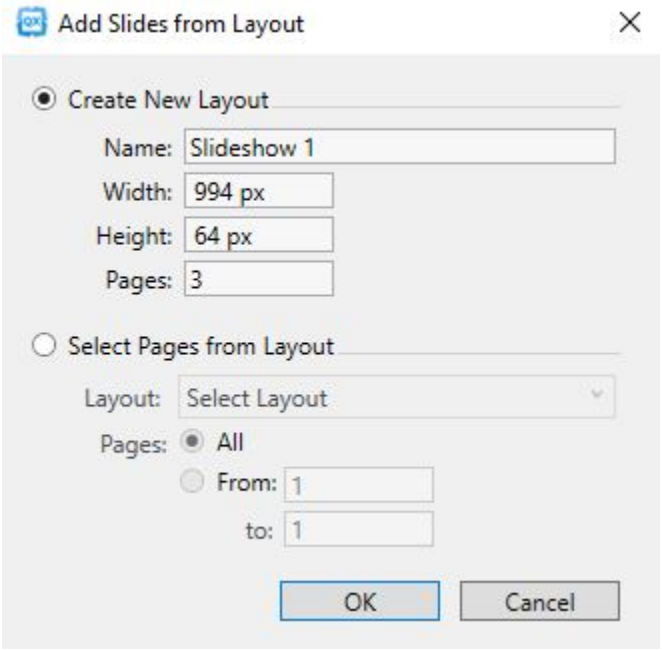


Slideshow options

- **Name**: Optionally, enter a name for the slideshow in the **Name** field. This name displays in the **Interactive Objects** area in the bottom of the palette.

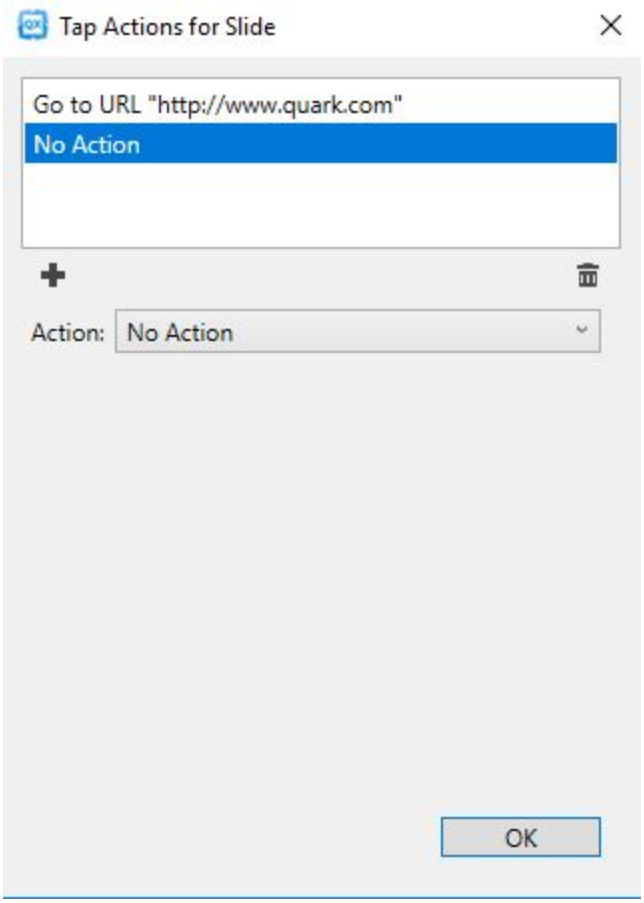
- **Initially Hidden**: If you want the slideshow to be initially invisible, check **Initially Hidden**.
- **Allow Fullscreen**: To allow the user to switch a slide to full-screen mode and back by double-tapping it, check **Allow Fullscreen**.
- **Show Uncropped in Fullscreen**: To display slides uncropped when the slideshow is in fullscreeen mode, check **Show Uncropped in Fullscreen**. If this box is not checked, the slides use their picture-box crop in fullscreen mode.
- **Allow Interaction**: To allow the user to interact with the slideshow, check **Allow Interaction**.
- **Show Indicator**: To disable slideshow indicators, uncheck **Show Indicator**. By default, the option is checked to show slideshow indicators.
- **Transition**: Lets you control the transition between slides (**None**, **Slide**, **Fade**, **Flip**).
- **Speed**: Lets you specify how long each transition lasts.
- **Auto Play**: To make the slideshow start playing immediately when the user displays the page, check **Auto Play**. If this box is unchecked, the user must manually change the slides with finger swipes or with buttons.
    - **Loop**: To have the slideshow play indefinately, check **Loop**. If this box is not checked, the slideshow will stop playing at the last slide.
    - **Slide Duration**: Lets you control how long each slide is on the screen.
- **Animate Pan and Zoom**: To make the slide pan and/or zoom at the beginning of its display, check **Animate Pan and Zoom**.
    - **Duration**: Lets you control how long the pan/zoom lasts. At the end of this duration, the slide stops and stays in its final position until the next slide is displayed.
    - **Start and Stop**: These buttons allow you to set the beginning and ending crop of each image. Click **Start** and scale/position the picture for the initial position, then click **Stop** and scale/position the image for the final position.
- To add a slide, click at the bottom of the slide list and choose one of these options:
    - **Add Slides from Picture Files**: Lets you add picture files to the slideshow. You can use picture files in PNG, JPEG, TIFF, PDF, and EPS formats. You can use the Shift or Command/Ctrl key to select multiple files.

- **Add Slides from QuarkXPress Layout**: Displays the **Add Slides from Layout** dialog. This dialog lets you add pages from a QuarkXPress layout to the slideshow. You can create a new layout or choose pages from an existing layout.



Add Slides from Layout dialog

- To edit a slide, select it and click . If the slide is a picture file, that files opens in the default image editing application. If the slide is a page from a QuarkXPress layout, the layout opens and scrolls to that page.
- To execute an action when the user taps a slide, select it and click . The **Tap Actions for Slide** dialog displays.



Tap Actions for Slide dialog

- **Action**: To add an action, click ✚, then choose the action from the **Action** drop-down menu and configure it the way you want it. (For more information, see "*Working with interactivity actions*."

- To set the cropping for a slide, select the slide's icon in the list, then scale and reposition the preview of the slide in the picture box.
- To delete a slide, select the slide's icon in the list and click 🗑.

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

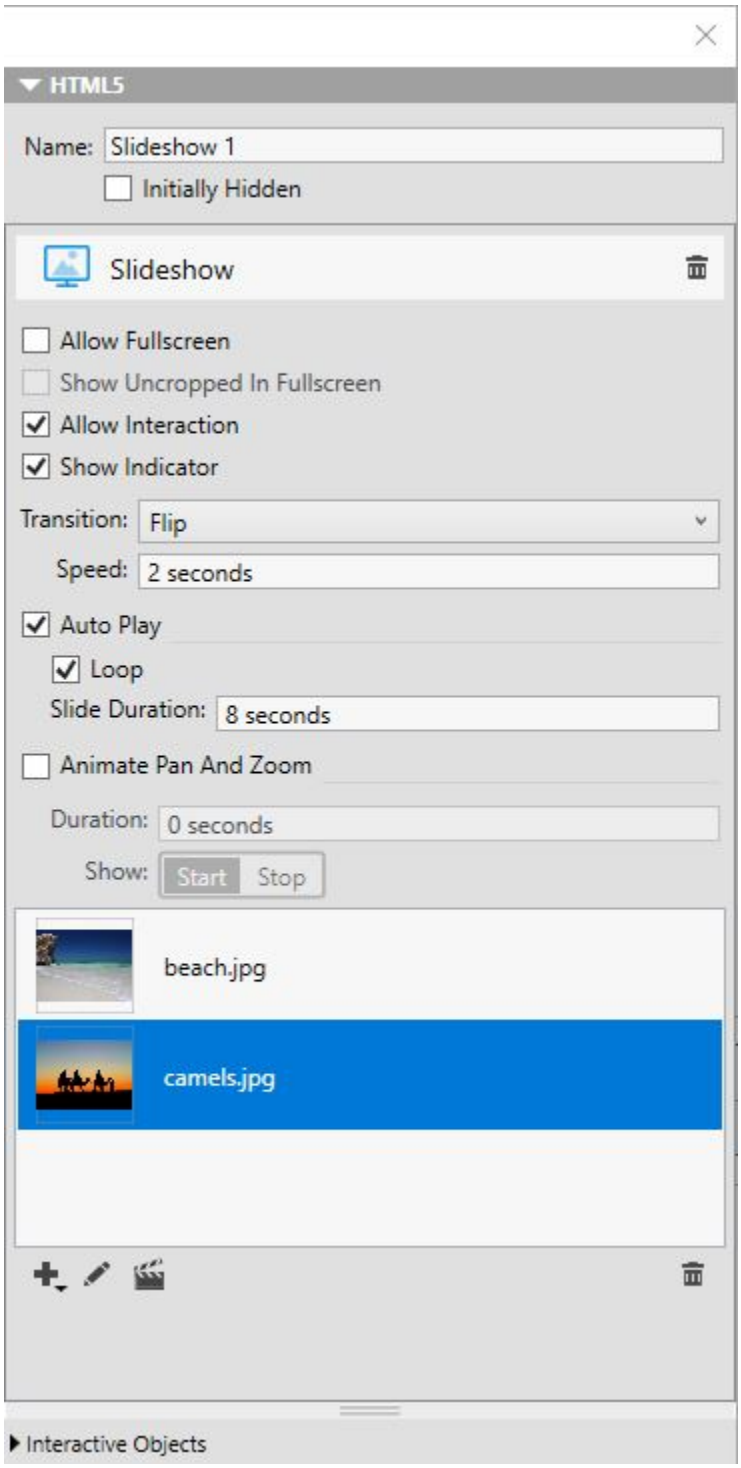
Adding a flip page effect (using a slideshow)

You can add a page flip effect to a page, giving your reader the impression that the currently displayed page is being flipped.

To create this effect:

1. Setup a full page slide show using the HTML5 palette with the following options:
   - Disable **Allow Fullscreen**

- Enable **Allow Interaction**
- Disable **Auto Play**
- Disable **Animate Pan and Zoom**
- Set the **Transition** to **Flip**



Slideshow options

2 Add two slides to the slideshow using a QuarkXPress layout. Click ✚ at the bottom of the slide list and choose **Add Slides from QuarkXPress Layout**.

3 Design the new slideshow. Page 1 of the new layout is your front page, page 2 is the back page.

4 Go back to your main layout and add a picture box.

5 Make the picture box a Button using the **HTML5** palette.

- Assign the **Next Slide** action to the button
- Assign **Slideshow** as the name of your newly created slideshow.

Adding a video

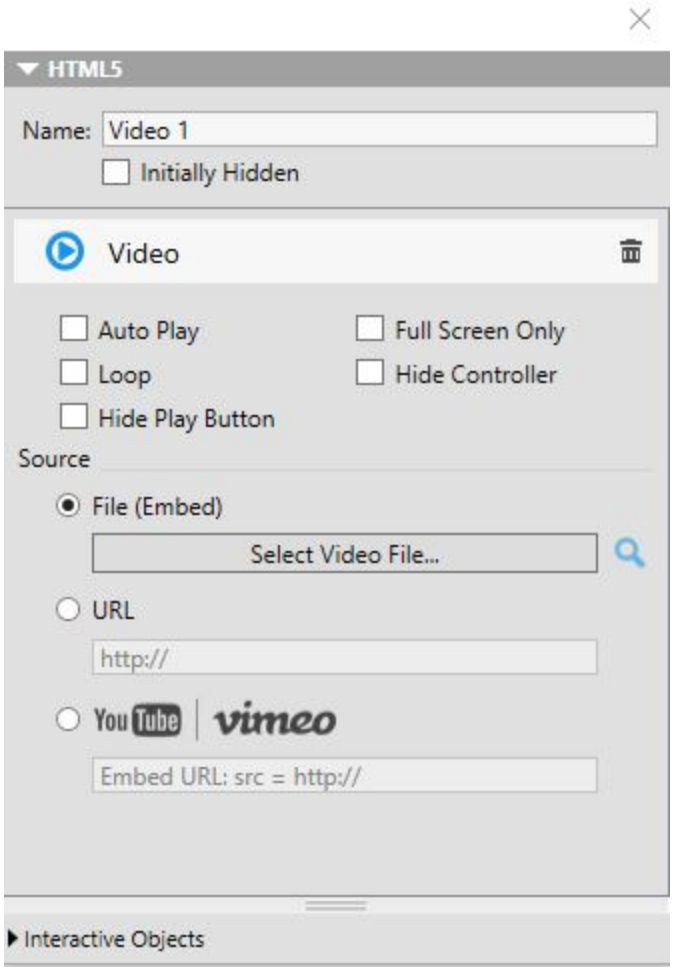
Digital layouts support only H.264 video at up to 720p, 30 frames per second, Main Profile level 3.1, with AAC-LC stereo audio at up to 160 Kbps per channel, 48kHz, in the .mp4 file format.

You can start and stop the playback of video files with actions. For more information, see "*Working with interactivity actions*."

To add a video to a Digital layout:

1 Select the picture box you want to contain the video.

2 In the **HTML5** palette, click **Video**.



Video options

- **Name**: Optionally, enter a name for the picture in the **Name** field. This name displays in the **Interactive Objects** area in the bottom of the palette.
- **Initially Hidden**: If you want the video to be initially invisible, check **Initially Hidden**.
- **Auto Play**: To make the video play automatically when the page it is on displays, check **Auto Play**.
- **Full Screen Only**: To make the video switch to fullscreen mode as soon as it begins to play, check **Full Screen Only**.

I

- **Loop**: To make the video play repeatedly, check **Loop**.
- **Hide Controller**: To hide the default video controls, check **Hide Controller**. Note that if you want the user to be able to control the movie, you will need to provide a way of controlling the video with actions.
- **Hide Play Button**: To hide the play button, check **Hide Play Button**.
- To specify the location of the video, click an option in the Source area.

- To use a local video file, click **File (Embed)**, then click the button and select the video file.
- To use a video from a URL, click **URL** and enter the URL in the field.
- To use a video file from Quark Publishing Platform, click **Quark Publishing Platform**, then click **Browse QPP** and select the video file.
- To use a video file from YouTube or Vimeo, click **YouTube/Vimeo**, then paste the url of the video in the src= field. (To get this value for a Vimeo or YouTube video, Control+click/right-click the video and choose **Copy embed code**. Then paste the code into a text editor, and then select the portion of the code that looks like this: http://www.[site].com/embed/XXXXXXXXXX and paste it into the field.)

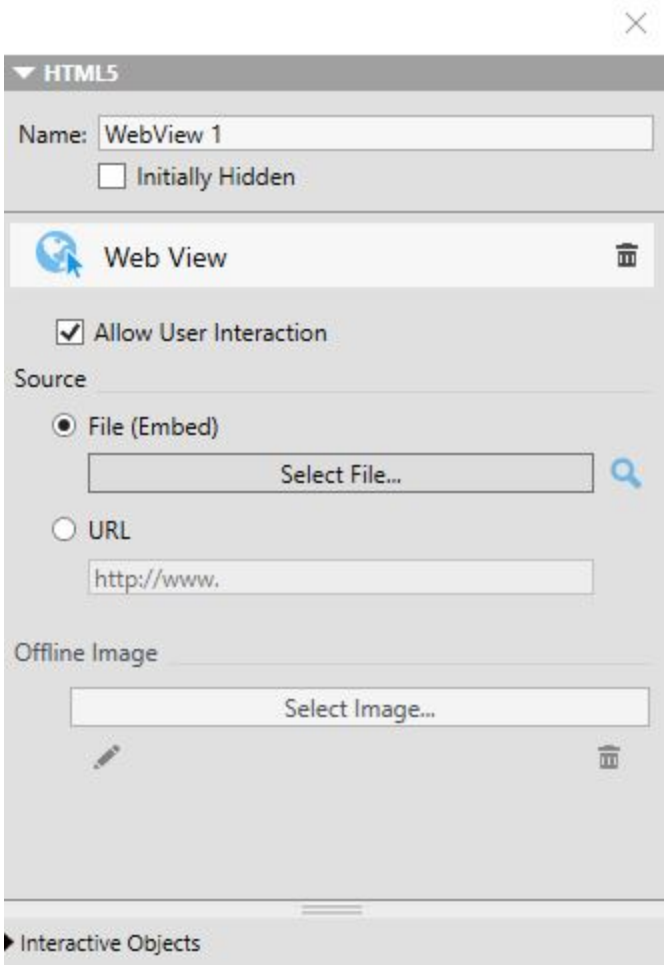
To finish configuring the video, click a different object or a blank part of the layout.

Adding a web view

You can use a Web view to include changeable content (such as ads) or custom interactivity to a Digital layout. You can use a Web 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 a Web view to a Digital layout:

1. Select the picture box you want to contain the Web content.
2. In the **HTML5** palette, click **Web View**.



Web view options

- **Name**: Optionally, enter a name for the Web view in the **Name** field. This name displays in the **Interactive Objects** area in the bottom of the palette.
- **Initially Hidden**: If you want the Web view to be initially invisible, check **Initially Hidden**.
- **Allow User Interaction**: To allow the user to interact with the content, check **Allow User Interaction**.
- To specify the location of the content, click an option in the **Source** area.
  - To use a local file, click File (Embed). Put the file and any other files it uses (such as pictures) into a folder by themselves, then click the button and select the file. To ensure that links work, all files in the folder that contains this file will be added to the article.
  - **URL**: To use a URL, click **URL** and enter the URL in the field.
- If the content in the Web view is not embedded, you can specify an image to display when the Internet connection is not available. To do so, click **Select Image** in the **Offline Image** area and choose a picture file. You can use the icons under this button to edit or delete the offline image.

➡ Offline images must be in PNG or JPEG format.

To finish configuring the Web view, click a different object or a blank part of the layout.

Working with interactivity actions

Actions let you add interactivity to items in a Digital layout that you will be exporting.

The available actions include:

- **No Action**: The default action. Does nothing.
- **Go to URL**: Navigates to a URL when the user touches the selected item. For Button interactivity, you can also add a geolocation value for Go to URL action.
- **Go to Page**: Jumps to a different page. For more information, see "*Creating a Go to Page action*."
- **Go to Next Page**: Jumps to the next page.
- **Go to Previous Page**: Jumps to the page before this one.
- **Go to First Page of Issue**: Jumps to the first page in the issue.
- **Go to Slide**: Displays the specified slide of the specified slideshow.
- **Next Slide**: Displays the next slide of the specified slideshow.
- **Previous Slide**: Displays the previous slide of the specified slideshow.
- **Play Object**: Lets you start playing the specified audio or video object.
- **Pause Object (Toggle)**: Lets you pause and resume the playing of the specified audio or video object.
- **Play Sound File**: Lets you play a sound file. For more information, see "*Creating a Play Sound File action*."
- **Pause Sound Files (Toggle)**: Lets you pause and resume the playback of all sound files.
- **Show Pop-Up**: Lets you display a pop-up. For more information, see "*Creating a Show Pop-Up action*."
- **Hide Pop-Up**: Lets you hide the currently displayed pop-up.
- **Show Object**: Lets you show the specified object (if it is hidden).

➡ Any item initially marked hidden will be hidden if the **Show Object** interactivity exists for that item on the page.

- **Hide Object**: Lets you hide the specified object (if it is visible).

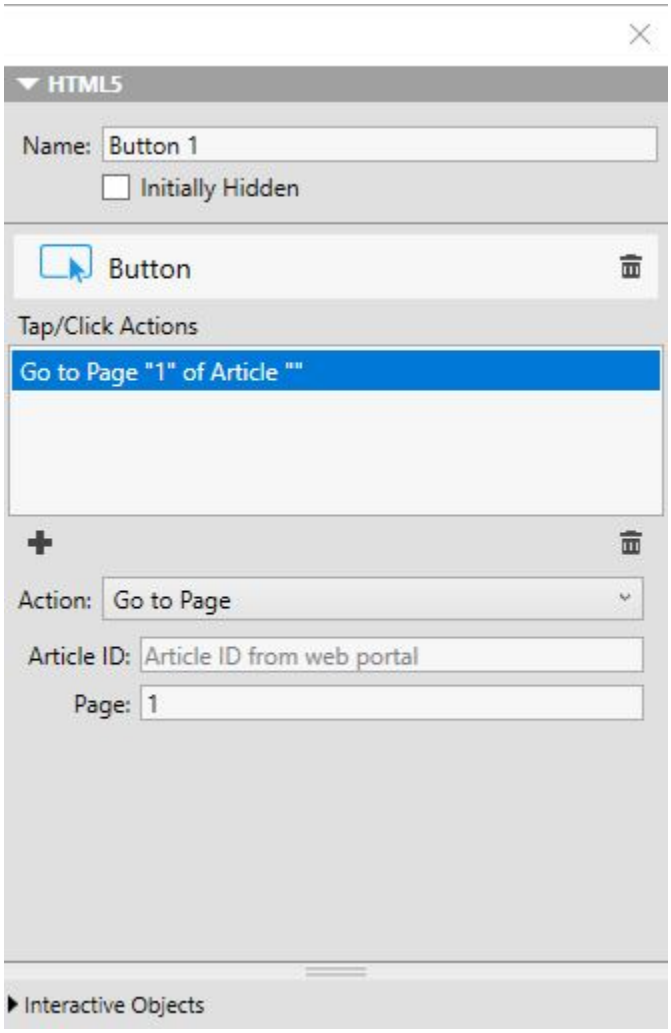
- **Open**: Lets you open files and display them within your application. This must be a format that the device understands how to display. For example, you can open a PDF file and display that within your app on a device that supports PDF file viewing.
- **Play Animation**: Lets you start playing the specified animation.
- **Reset Page**: Lets you reset the page, so that all animations are set back to their starting position.
- **Take Snapshot**: Lets you take a snapshot (screenshot) of the page and save it in the default image folder of your device.

Creating a Go to Page action

A **Go to Page** action lets the user jump to a different page.

To create a **Go to Page** action:

1 In QuarkXPress, Choose **Go to Page** from the **Action** drop-down menu.



- **Article ID**: Enter the project name in the **Article ID** field. (optional)

- **Page**: Enter the absolute page number of the target page in the **Page** field. (For example, to go to the second page in the target article, enter 2.)

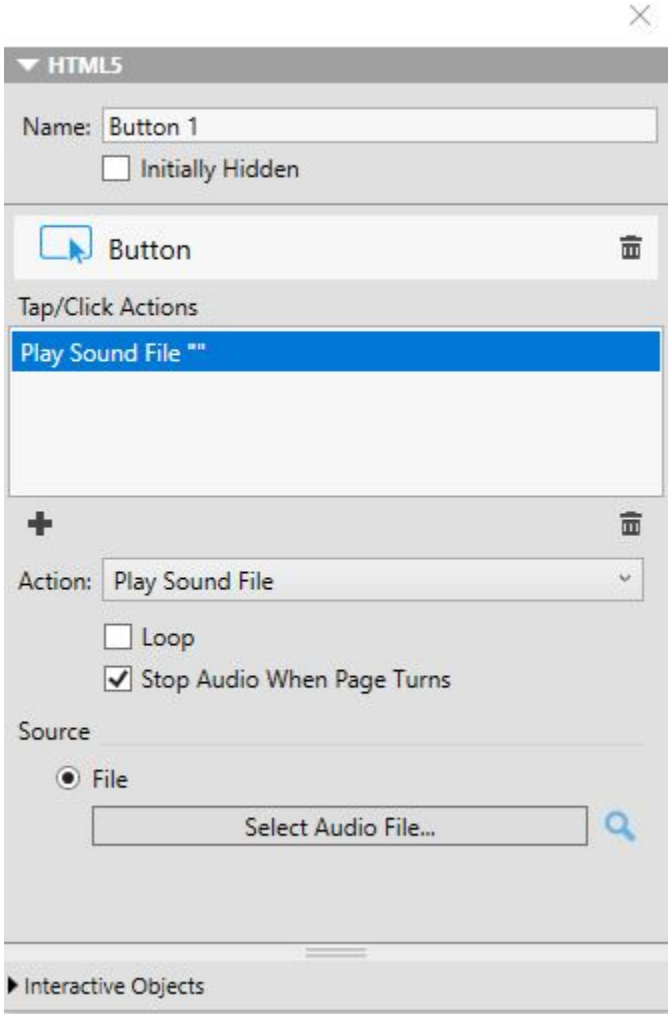
Creating a Play Sound File action

A **Play Sound File** action lets you play a sound file.

➡ Only .mp3 files are supported.

To create a **Play Sound File** action:

1 Choose **Play Sound File** from the **Action** drop-down menu.



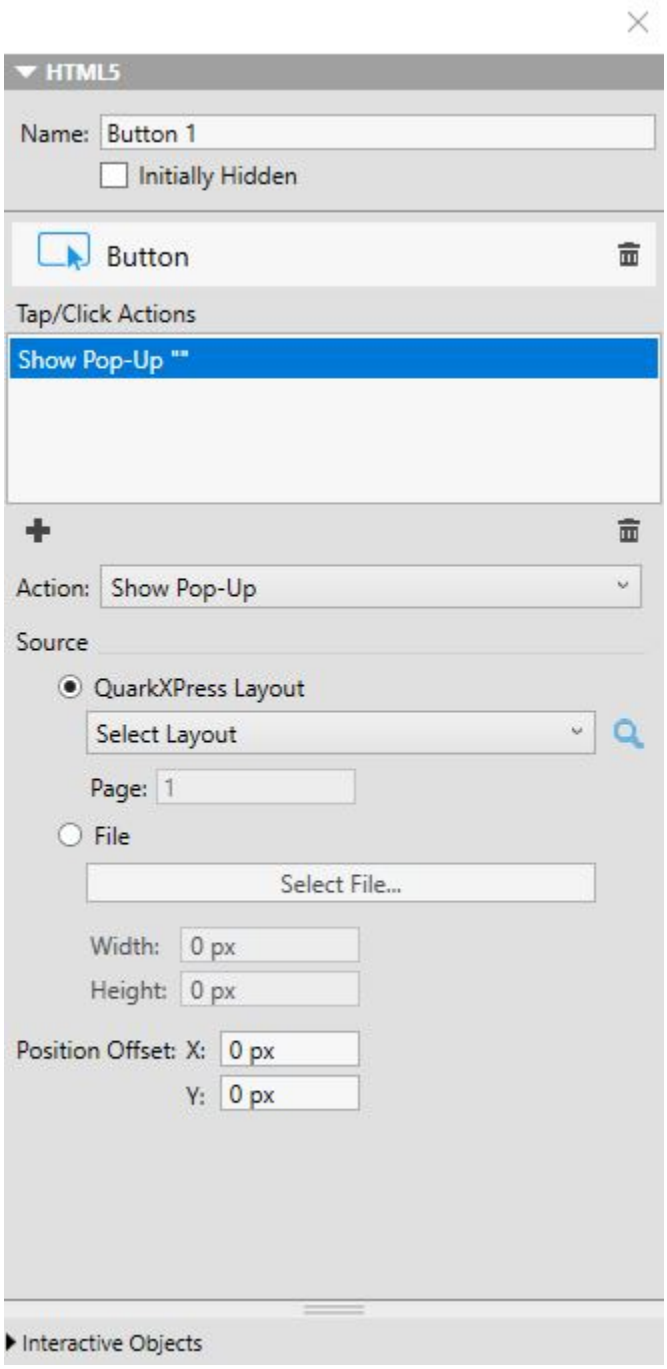
- **Loop**: To make the sound play over and over again, check **Loop**.
- **Stop Audio When Page Turn**: To make the sound stop when the user navigates to a different page, check **Stop Audio When Page Turns**.
- To specify the location of the sound file, click **File (Embed)**, then click the button and select the sound file.

Creating a Show Pop-Up action

A **Show Pop-Up** action lets you display a pop-up view containing either the contents of a QuarkXPress layout in the same project file or the contents of an external file. If you choose a QuarkXPress layout, any interactivity in that layout should work in the pop-up. If you choose to display an external file, you can use any file that can be displayed in the native mobile browser on your device.

To create a **Show Pop-Up** action:

1 Choose **Show Pop-Up** from the **Action** drop-down menu.



- Under **Source**, select one of the following options:

- **QuarkXPress Layout**: To display the contents of a different QuarkXPress layout, click **QuarkXPress Layout**, then choose the target layout from the drop-down menu or choose **Create New Layout** and create a new layout. Enter a page number in the **Page** field to indicate which page to display in the pop-up. Click the **Edit** button to display the layout.
- **File**: To display the contents of an external file, click **File**, then click **Select File** and navigate to the file you want to display. Use the **Width** and **Height** values to specify the size at which to display the file. If you use an external file, the pop-up includes a built-in close button.

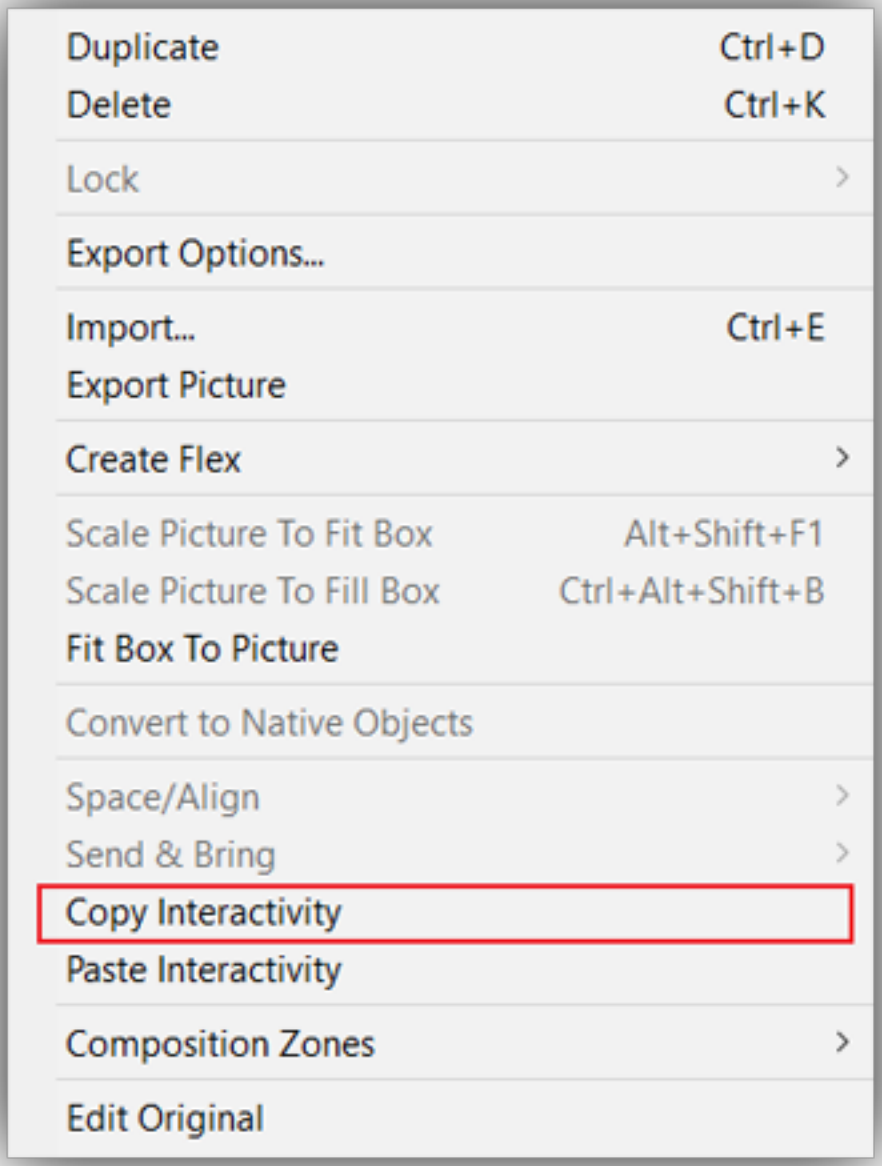
- **Position Offset**: To offset the content of the pop-up, enter values in the **Position Offset** fields.

Seamlessly Copy/Paste Interactivity

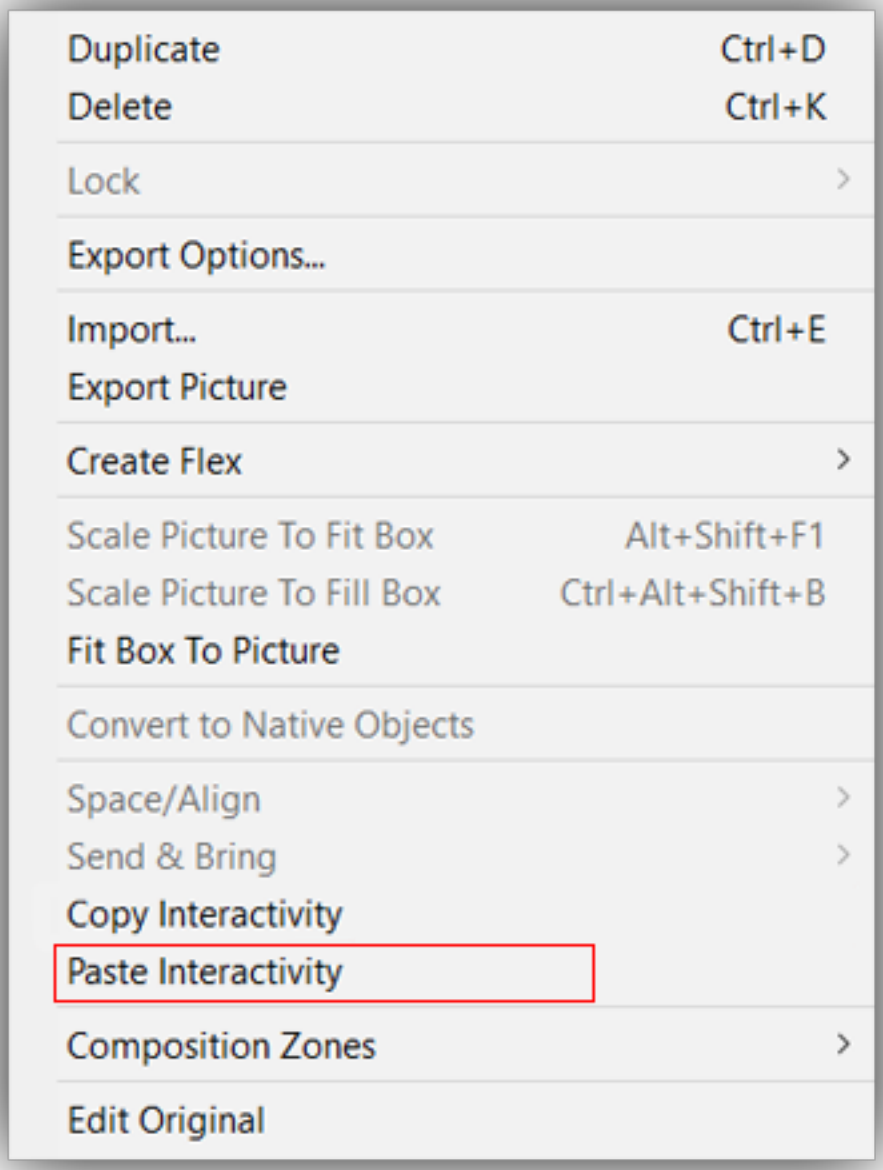
Start creating interactive content in an even more efficient way. The interactivity of one box can be copied and applied to another box in a click. Also, the user experience or UI can be improved in lesser time as compared to manual operation on the system.

Example to see how it works?

Right click on the first box and click “Copy Interactivity", this will copy interactivity used on the box and can be applied the same on the another box.



Now, go to second target box, do right click and choose “Paste Interactivity”.



East Asian features in digital layouts

Text features that are considered East Asian features are only available when you have the East Asian preference enabled (QuarkXPress/Edit > Preferences > East Asian).

This chapter describes the East Asian features that can be applied in a Digital layout.

For more detailed information about applying East Asian text features see “Text and Typography” in *A Guide to QuarkXPress*.

Using font fallback

When Font Fallback is on, if the application encounters a character that is not available in the current font, it searches through the active fonts on your system to find a font that does include that character. For example, if Helvetica is applied at the text insertion point and you import or paste text containing a Kanji character, the application might apply the Hiragina font to that character. If the application cannot find an active font that contains the character, the character still displays as a box or symbol.

Font Fallback is implemented as an application preference, meaning that the feature is either on or off for your copy of the program. The feature is on by default, but if you need to turn it off, uncheck **Font Fallback** in the **Font Fallback** pane of the **Preferences** dialog (QuarkXPress/Edit > Preferences).

Specifying story direction

You can position text so that it runs left-to-right and top-to-bottom or top-to-bottom and right-to-left. To specify a story direction, choose Style > Story Direction, and then choose **Horizontal** or **Vertical**.

Working with grouped characters

Use the Group Characters dialog box (Style menu) to include a group of horizontal characters, such as Roman characters, within a vertical line of text. Grouped characters always display horizontally and do not break at the end of a line. To group selected characters:

1. Choose Style > Group Characters.
2. In the two Scale fields, enter the horizontal and vertical scale percentages for the grouped characters.
3. If you want to change the character spacing, enter a value in the **Track/Sending Amount** field.
4. Click **OK**.

Working with rubi text

Rubi text clarifies the meaning or pronunciation of base text. Base text can run vertically or horizontally, and rubi text usually follows the direction of the base text. Rubi text can be placed to the right of or to the left of base text in a vertical story, and above or below base text in a horizontal story.

➡ The left position in a vertical story and the below position in a horizontal story are rendered correctly only when the box is exported as a graphic.

You can manipulate the alignment, placement, scale, font, color, opacity, shade, type style, and relative size of rubi text. In addition, you can choose from several options that allow you to control rubi text that overhangs unrelated base text.

➡ Some of these features are available only when the box is exported as a graphic.

Use the **Rubi** dialog (**Style** menu) , the **Home/Character Attributes** tab of the **Measurements** palette or a contextual menu to add rubi text to selected base text.

Auto Rubi can be applied through the following keyboard shortcuts:

- *(macOS)* CMD+OPT+Shift+R
- *(Windows)* CTRL+ALT+Shift+ R

Applying emphasis marks

To apply an emphasis mark to a character, select the character, go to the **Character/Character Attribute**s tab of the **Measurements** palette, click the **Text Style**s icon and select the **Emphasis Mark** drop down to display the emphasis mark options, and then click one of the options. These options are also available in the **Home/Classic** tab of the **Measurements** palette and under Style > Type Style.

Working with mojigumi sets and classes

The Mojigumi feature lets you control spacing for specific punctuation characters when they occur in particular locations. To use this feature, you must choose or create a *mojigumi character class* and a *mojigumi set*.

- A *mojigumi character class* is a named set of punctuation characters that should always be spaced in a particular way.
- A *mojigumi set* is a set of character spacing specifications based on the widths of character em boxes. For example, a mojigumi set might dictate that open punctuation should use fixed half-width spacing when it occurs at the beginning of a line, and that close punctuation should use full-width or half-width spacing when it occurs at the end of a line. Each mojigumi set is associated with one mojigumi character class.

To use the Mojigumi feature, apply a *mojigumi set* to a paragraph. The *mojigumi set's* settings are applied to the characters in the associated *mojigumi character class*.

For example, if you do not want a parenthesis to occupy a full em-box width when it occurs between two full-width characters, you can create a *mojigumi character class* that contains parentheses and then specify in the *mojigumi set*

settings that these characters should always use a half-em width when they fall between two full em-width characters.

Working with non-breaking character sets

Non-breaking characters are characters that cannot begin or end a line, and also which cannot be separated from each other at line breaks. The **Non-Breaking Character Set** dialog (**Edit > NonBreaking Character Sets**) includes default non-breaking character sets for Japanese Strong, Japanese Weak, Korean Standard, Simplified Chinese Standard and Traditional Chinese Standard.

To create custom non-breaking character sets:

1. Choose Edit > Non-Breaking Character Sets. The **Non-Breaking Character Sets** dialog box displays.
2. Click **New**.
3. Enter a name in the **Name** field.
4. Enter characters in the **Enter characters that cannot begin a line** field, the **Enter characters that cannot end a line** field, and the **Enter non-separable characters** field.
5. Click **OK**.
6. Click **Save**.

➡ To apply a non-breaking character set to a paragraph, choose one from the **Non-Breaking Char Set** drop-down menu in the **Edit Hyphenation & Justification** dialog (Edit > H&Js > Edit).

Text in digital layouts

By default, text boxes in a Digital layout are set to export as HTML text. This allows them to be searchable and selectable. However, it also means that you cannot apply some of the higher-end text controls (such as Drop Shadow applied Text boxes etc) to text in such boxes.

If you want to have complete typographical control over text in a box, check **Convert to Graphic on Export** on the **Measurements** palette. The box will export as an image rather than as text, so its text won't be searchable or selectable, but you can make it look exactly the way you want it to.

Horizontal and vertical text scaling are kept in HTML5 Publications and fixed-layout ePub, including when both are applied to the same text. Reflowable ePub keeps neither horizontal nor vertical scaling.

Fonts in digital layouts

Unless you convert a text box to a graphic (see "*Text in digital layouts*"), you must make sure that you use fonts that is available on the devices where they will be displayed. For a list of the fonts supported on iOS, see *http://iosfonts.com*. Android devices, by default, support only Droid Serif, Droid Sans, and Droid Sans Mono, but additional fonts can be installed.

- When exporting to digital formats like ePub and HTML5 publications, you can now use custom fonts, if they are in the TTF or OTF format. These will be embedded and therefore can be displayed in the output. If you are embedding large font files, this will increase the file size of your content substantially.

Variable fonts: when you export to HTML5 Publication or ePub, text set in a variable font keeps its axis settings, such as weight, width or optical size, in the output. Variable fonts are embedded in the same way as other fonts. To make sure the text looks the same on every device, embed the font (TTF or OTF) or make sure it's available on the device; otherwise the reader's browser or app uses a substitute font.

Hyperlinks in digital layouts

To apply a hyperlink to text, select the text and use the **Hyperlinks** palette (for more information, see *A Guide to QuarkXPress*).

To apply a hyperlink to an entire box, select the box and use the **HTML5** palette (see "*Adding a button*").

Groups in digital layouts

If you have multiple overlapping picture boxes and graphic text boxes, you may be able to streamline your export by grouping the boxes and then checking **Convert to Graphic on Export** for the group. If you do not group such boxes, they all export as separate HTML elements. If you group them, however, they export as a single graphic object.

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. 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 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.

Updating missing files

Multiple missing digital files can now be updated all at once using the **Usage** dialog.

Reviewing digital publishing asset usage

To see which assets have been used in 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 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 is a file or a folder. To view more information about the selected asset, expand the **More Information** area at the bottom.

Default Color Profile for Digital Layouts

As per W3C recommendations, all the browsers are supposed to use sRGB profiles in case images used do not have any profile embedded into them. To have WYSIWYG experience, we have set the default source setup to be “QuarkXPress Digital Default” in all Digital Layouts, that uses sRGB color profiles. Now QuarkXPress will render images in the same way as being rendered in the browsers. Apart from this, by default, some RGB colors are added in the color palette for Digital Layouts.

Working with Reflow

Exporting a Digital layout to an ePub eBook format offers the option of exporting to either Fixed or Reflow. Reflow view lets end users view content in a full-screen format with an adaptive layout and resizable text. This view is designed to let end users view content without being constrained or distracted by the layout.

If you want to take advantage of this feature, you must create at least one reflow article in QuarkXPress. If you find it useful for organization, you can create one article for each chapter or section, but this is not necessary.

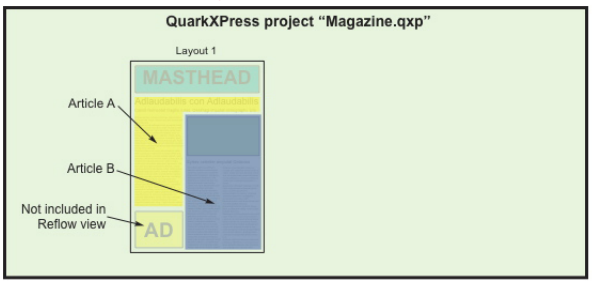
The **Articles** palette allows you to tag the content for Reflow ePub. Further, the **Articles** palette offers the capability to group the content by Articles allowing reordering of components within an Article.

When you export a Digital layout as an ePub, QuarkXPress will always export the first page of the layout as an image and use it as a cover of your eBook.

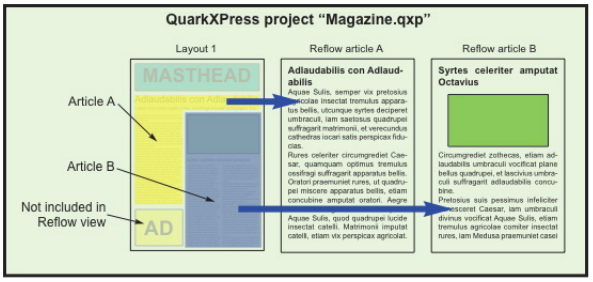
If you choose to create a reflow article from a Digital layout, the process is shown in the following diagrams:



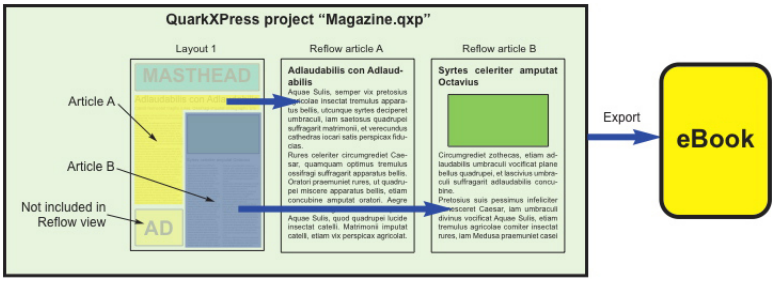
Step1: Create the layout



Step 2: Identify chapters or sections, and identify what parts of the layout you don't want to include in text reflow view at all



Step 3: Use the Articles palette to create articles, ad article components (text and pictures) and define the flow order for Reflow ePub view



Step 4: Export the layout as an eBook and view it in the eReader

Creating reflow articles

There are two ways to create reflow articles:

- *From a selection*. This approach lets you select the specific boxes and tables you want included in the reflow article. This approach is best if the layout has a lot of content that should not be visible in the Reflow output.
- *From pages*. This approach copies everything on the pages you indicate into the reflow article. This approach is best if all or almost all the content in the layout should be visible in Reflow output. (Any content from master pages such as headers and footers, is ignored.)

➡ If you are planning to export the layout as reflow ePub, you may want to remove the items of Page 1 from the Article palette, so that the items on the cover page are not shown twice in the output.

➡ Projects that were created with Quark Publishing Platform® or edited with QuarkCopyDesk® may already have one or more articles. If you open a QuarkCopyDesk article or a project file that includes articles with components, the reflow content is maintained. All components that had been tagged for reflow are still included in the reflow content.

➡ To highlight components that have been tagged for Reflow, click the **Article Highlighting** button on the **Articles** palette.

➡ In an exported eBook, there are page breaks between every reflow article.

Creating a reflow article from a selection

To create a reflow article from one or more selected items:

1. Select the content you wish to add to a Reflow article.
2. Do one of the following:
   - Choose **Item > Digital Publishing > Add to Reflow > New Article**. QuarkXPress creates a new reflow article and inserts all the text and pictures in the selected boxes.
   - Click **New Article** in the **Articles** palette, and then click the **Add Item** button.

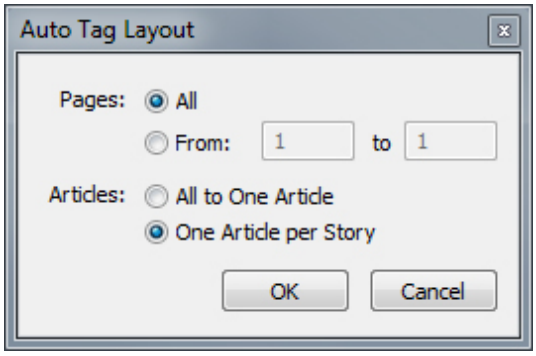
QuarkXPress adds the selected content.

3. To change the name of the new article, select it and click **Edit Properties**. Enter a new name for the article and then click **OK**.

Creating a reflow article from pages

If you want to convert one or more pages' worth of content into an article or articles at the same time, use the following procedure:

1. Choose **Layout > Auto Tag Layout**. The **Auto Tag Layout** dialog displays.



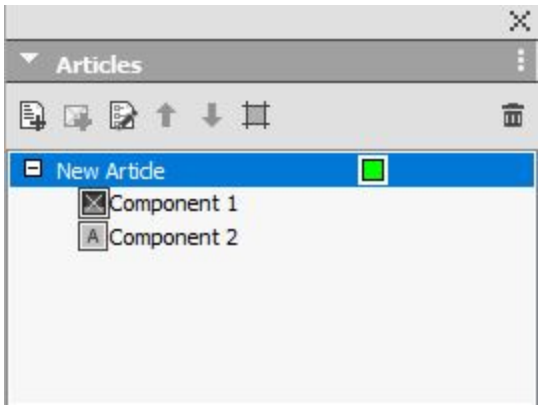
Auto Tag Layout dialog

   - Use the **Pages** area to specify whether you want to add all the pages in the layout or only a specific range.
   - Use the **Articles** area to specify whether you want all the pages to go into the same article, or to create a separate article for each story.

2. Click **OK**. QuarkXPress creates the necessary reflow articles and inserts all the content from the selected pages into them.

To verify the creation of the reflow article:

- Display the **Articles** palette (**Window** menu).



The Articles palette for a Layout may contain one or more articles. Each article may contain one or more picture boxes, and stories than span multiple text boxes.

- To change the name of the new article, select it and click **Edit Properties**. Enter a new name for the article and then click **OK**.

Adding content to a reflow article

To add content to an existing reflow article, display the **Articles** palette (**Window** menu), select the target reflow article, and then select one or more tables, picture or text boxes and click **Add Component** in the **Articles** palette.

- If you want to add a full-page image to an ePub layout that is to be viewed in iBooks, its dimensions should be 1024 x 768 (or 2048 x 1536 for retina).

Reordering components in a reflow article

Each table, text box, story, or picture that you add to a reflow article is added as a component. By default, the components in a reflow article may not occur in the right order.

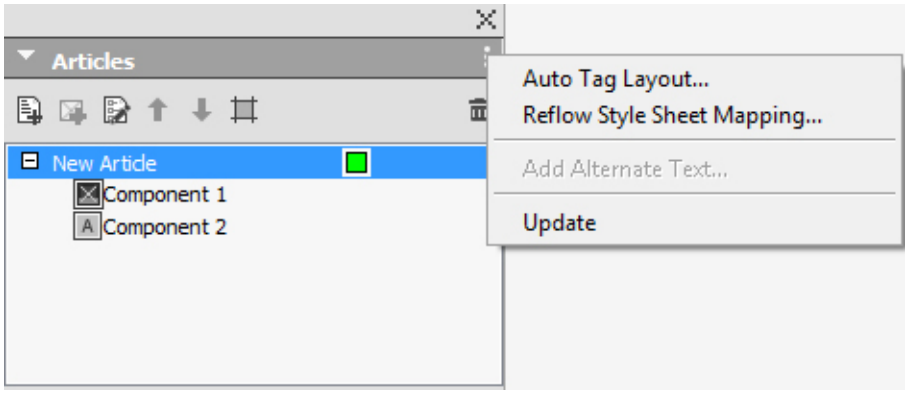
To change the order of components in a reflow article:

1. Display the **Articles** palette (**Window** menu).
2. Expand the target reflow article and select the component you want to move.
3. In the **Articles** palette, drag and drop or use the **Move Up** or **Move Down** button to reposition the component.

Reflow style sheet mapping

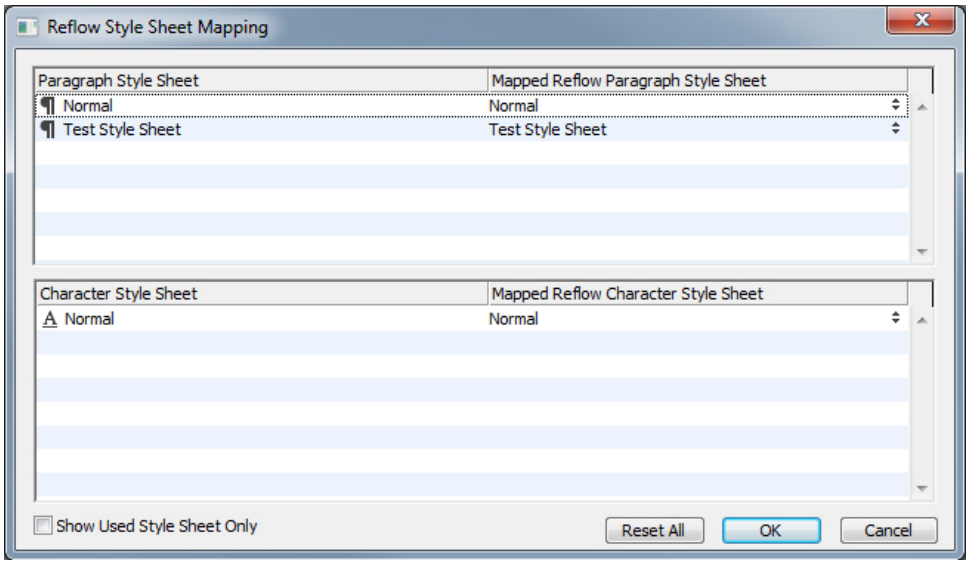
When exporting to reflow ePub, you can choose to map the existing paragraph and character style sheets to different style sheets.

1. Display the **Articles** palette (**Window** menu) and click the icon.



Articles palette

2 Choose **Reflow Style Sheet Mapping** from the menu. The **Reflow Style Sheet Mapping** palette displays.



Reflow Style Sheet Mapping palette

- Select **Show Used Style Sheet Only** to list only the style sheets used in the current project.
- Click **Reset All** to reset back to the defaults.
- For each paragraph and character style sheet you would like to override when exporting, select the style sheet on the left and choose the new style sheet to map to from the drop down menu on the right. These style sheets will be applied to the reflow ePub output only.

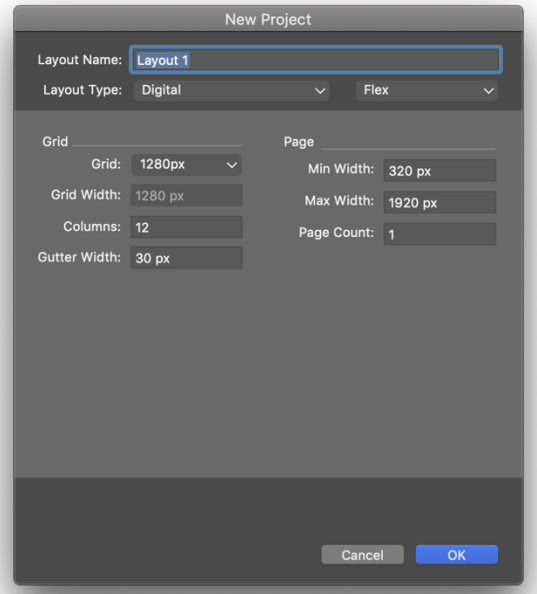
Flex Layout

➡ Get started with Flex Layouts by visiting http://www.quark.com/documentation/quarkxpress/2019/english/HTML5Output/Page1.html.

Creating a Flex Layout

To create a digital layout:

1 Choose File > New > Project. The **New Project** dialog box displays.



2 Enter a name for the layout in the **Layout Name** field.

3 Choose **Digital** from the **Layout Type** drop-down menu and **Flex** from its sub-drop-down menu.

Grid Settings

4 Choose appropriate grid size for your layout. Read more about grids at:

➡ *http://www.quark.com/documentation/quarkxpress/2019/english/HTML5Output/Page2.html*

5 Choose number of columns for the grid.

6 Select the **Gutter Width** for the grid, which will reflect as components padding in your design.

Page Settings

7 Select the minimum page width that you are planning to support. Typically, you would like it to be 320px, which is the minimum smartphone width available.

8 Select the maximum page width for your layout. The grid selected will always be centered within this selected max page width.

9 The **Page Count** allows you to specify the number of pages you want to initially create.

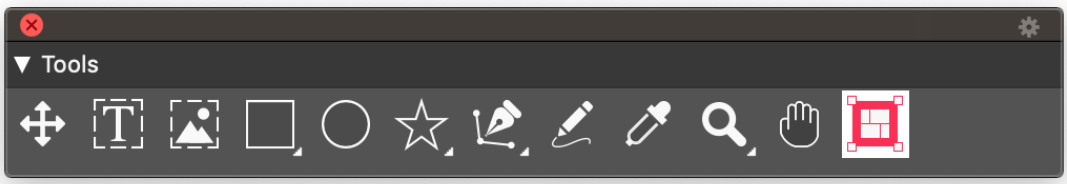
Working with Flex Layouts

Before you start working with Flex Layouts, we recommend you to go through "*Getting started with Flex Layouts*" guide.

We have introduced various tools within QuarkXPress to allow you to create responsive layouts. These different tools involve introducing a new tool called Flex Container, new property tabs for the measurements' palette called Container and Flex Item, the introduction of some new palettes like Flex Layout Blocks, Flex Object Browser, and Flex Page Builder, etc.

Flex Container Tool

Container tool helps you create a new type of box called container box. This container box is introduced to contain other boxes like picture box, text boxes, none boxes, etc. It can also contain other container boxes allowing users to create box containment hierarchies that are necessary for responsive layouts. Generally this tool has to be used only when you want to create your responsive widgets. As described in "*Getting started with Flex Layouts*", it is generally quite difficult to create any new component from scratch using this tool, as it is a low-level tool and needs you to fully understand all the different responsive properties. So to simplify the process of creating new components from scratch, we introduced another approach called grid and its responsive components like Container, Row, Column. You can find these components as part of Flex-Layout Blocks palette.

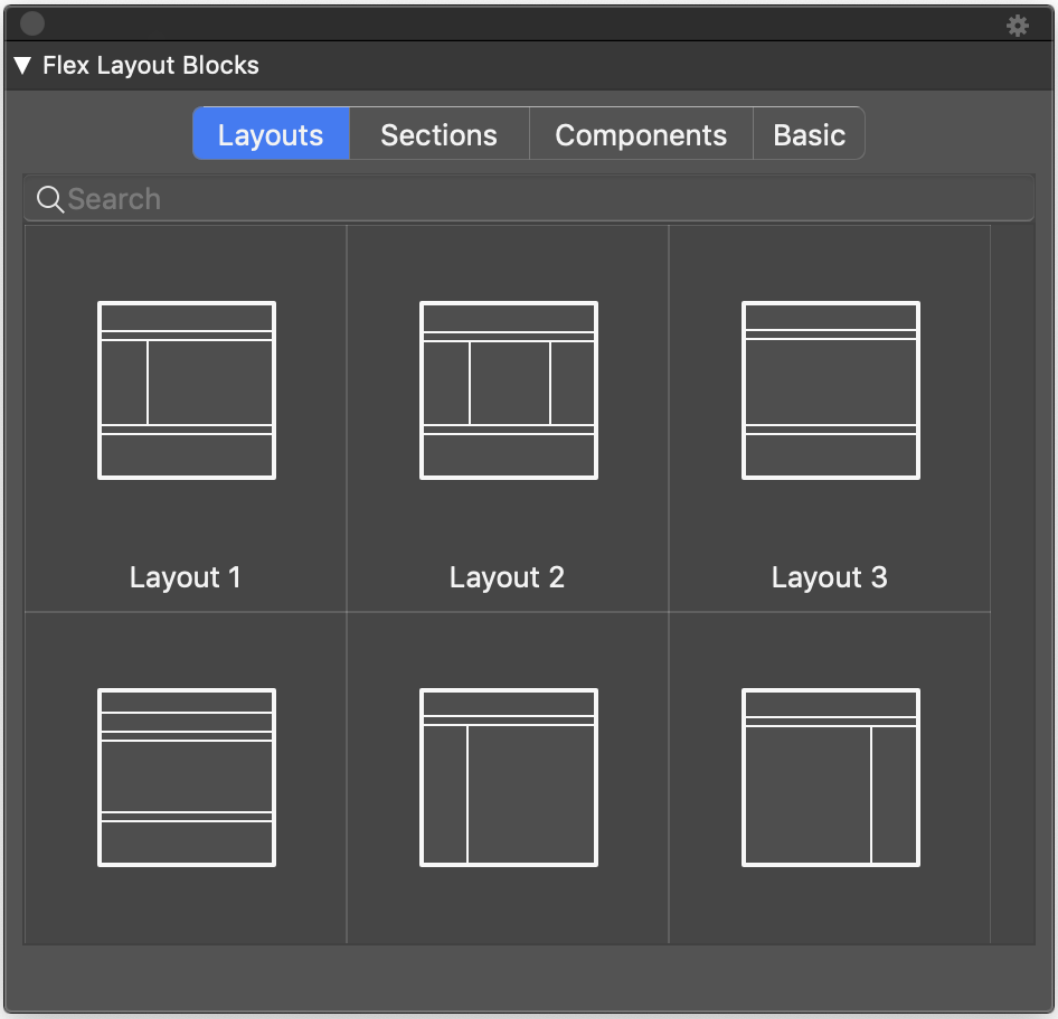


Flex Layout Blocks

A big challenge while creating responsive layouts is to understand the responsive properties and creating effective box containment hierarchies. Therefore, to solve these problems, we introduced Flex-Layout Blocks palette, which has some ready to use components adhering to the design approach using grids and their responsive components. This approach of using the grid and ready-to-use layout blocks is comparatively quite easy to create responsive layout than to use Container tool to create new component every time and use it for the layouts. Some features of this palette are:

- Ready-made, out of the box, responsive components - or “mini layouts” - to quickly build your web publications
- All the components in this palette are made to support grid-based design approach
- Blocks are divided into different groups:
    - Layouts (layout structure)
    - Sections
    - Components
    - Basic Tools

The whole idea of Flex-Layout Blocks is to provide some ready-to-use small components, that you can use to build your big components and eventually be able to create responsive layouts. For more information about Flex-Layout Blocks, please refer to “*Layout Blocks*”.



But even with Flex-Layout Blocks palette, there are some challenges that you might have to face. For example, though there are some reusable components available, they are relatively generic and small. This means, every time we try to create some layout, we still have to assemble these components to form relatively big layouts, and tweak their responsive behavior, etc. Furthermore, without the content and styling sometimes it becomes difficult for a designer to envision the overall look. Following are some challenges with Flex-Layout Blocks palette that you might face:

- These blocks are too generic
- Not knowing how every block behaves unless we drag n use it onto the layout
- Which block is suitable for a particular situation
- We still need to club multiple blocks together to make a single section

- Without content, it is hard to envision the overall look of design blocks

To solve some of these problems, we have introduced another palette called the Flex Page Builder palette.

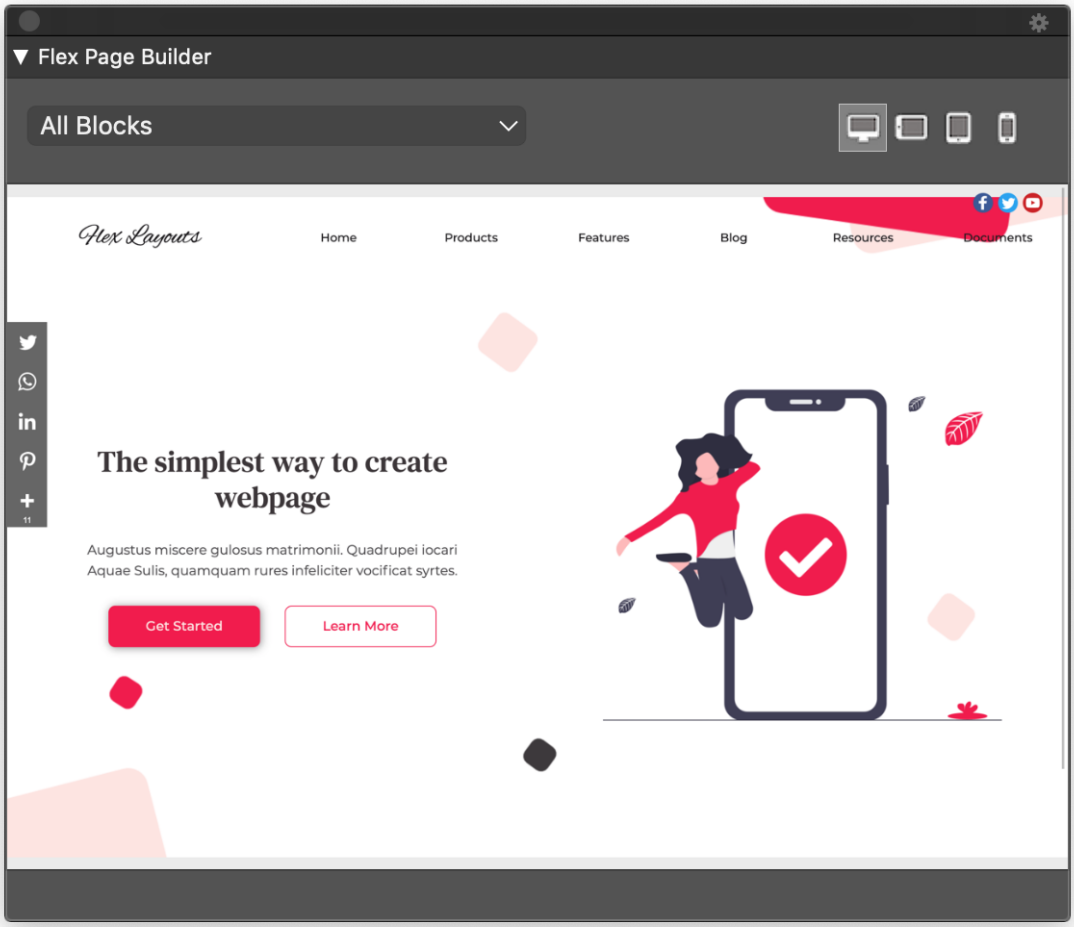
Flex Page Builder

Flex Page Builder palette is a new palette, which helps create responsive designs even faster than to use Flex-Layout Blocks and also addresses some challenges listed above while using Flex Layout Blocks palette.

Flex Page Builder palette is a palette with pre-backed design blocks with dummy content and styling. These pre-backed design blocks are section level blocks that can be used to quickly assemble a complete responsive layout. Once you have found any suitable section, you just need to drag that section onto the layout canvas, and change the content and styling as per your needs. These pre-designed blocks are fully customizable. You can change their content, responsive behavior, styling, etc.

Responsive designs are rules-based designs and these rules execute at your viewer's devices when the viewer opens these layouts in their browser/mobile application. These rules help layouts adapt to the viewer's device dimensions by still keeping the layout legible, elegant, and relevant. The traditional print designs either are not feasible for these responsive layouts or are not effective. So there seems to have a need for ready to use effective responsive designs so that not only can you design quickly by reusing them but can also get ideas for creating your own blocks if required.

These blocks give you the starting point to start designing your layouts. It has more than 125 pre-designed layouts that are fully responsive. You can use different combinations of these designs to create endless designs. Further these designs are categorized under different section types e.g. Headers, Footers, Call to Actions, Features, Teams, etc. which are industry proven effective designs. With dummy content and styling it becomes very easy to comprehend the design and responsive behavior. These designs also help you in your creative thinking if you want to create your own designs. These design blocks are fully compatible with Layouts Blocks, in case you want to create your own layout block and use it within these pre-backed designs.



How to Use

It is really easy to use Flex Page Builder. All you need is to choose the design block you want to use and double-click on it to add it to the layout. Double-clicking any section in the Flex Page Builder will add that section to the end of the layout. If there is any selection on the layout, then this new section automatically gets inserted after the selected section on the layout. Alternatively, you can also drag n drop any section from Flex Page Builder palette to the layout. If you drop the selected section on the blank area of the layout, then it will get appended to the bottom of the layout. But if you drop it into any highlighted zones between the existing sections, then it will get inserted at the highlighted zone.

Different design sections in the Flex Page Builder palette are categorized under various heads that are being used on the web/marketing industry, like call-to-action, pricing tables, features, teams, testimonies, etc. You can use the filter provided on the top of the Flex Page Builder palette to see sections only from a particular category.

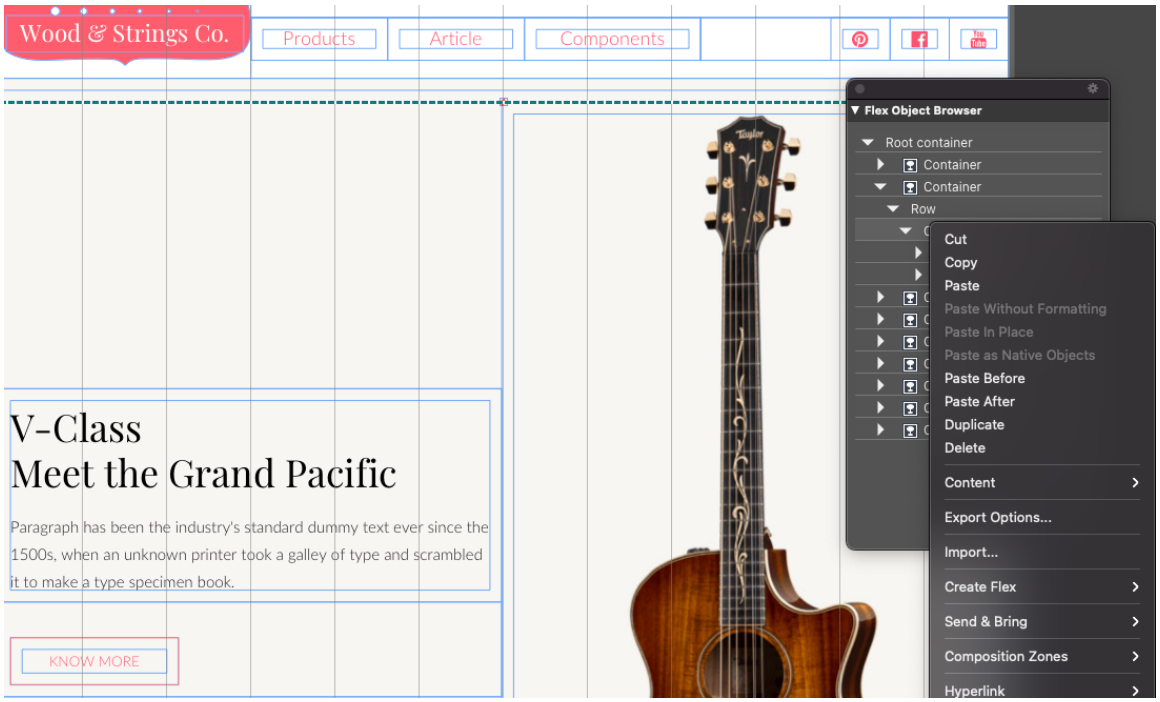
Moreover, you can use preview buttons for different devices provided on the top of the Flex Page Builder palette to see how a particular section design block will appear on different device sizes.

Customization

Once added to the layout canvas, you are free to customize any aspect of these design blocks. You can change not only change the content and images used in these blocks but can also change the text styling, colors, box styles, and responsive behavior of these blocks as per your needs. In case you need to add more sub-components within a particular section, you can select that sub component and duplicate it. Similarly, in case you want to remove any extra element that you don't need in a section just select and delete that extra box. For the selection of a particular box, you might like to use our Flex Object Browser palette, which allows you to easily see and traverse the box hierarchies.

Flex Object Browser

Flex Object Browser is a palette that helps you see and traverse the flex box containment hierarchies. You can see the parent hierarchy of a selected box in this palette, you can also see it's siblings and it's children boxes. Selecting a box in this palette will select the corresponding box on the layout and vice versa. Once you select a box in the Flex Object Browser, you can right-click on it to invoke the contextual menu and perform all these operations from here. Following are some useful operations that you would like to perform from this contextual menu:



Cut: Selected Container/Box will get deleted from current location and copied for pasting into some other location

Copy: Selected Container/Box will get copied for pasting into some other location.

Paste: Pasting a box, will paste copied box into the selected box as a child. And this child box will be appended to the end of its parent box.

Paste Before: If you want to paste the copied box before a particular box, then select that particular box and use paste before command.

Paste After: If you want to paste a box after the selected box, this command will do that.

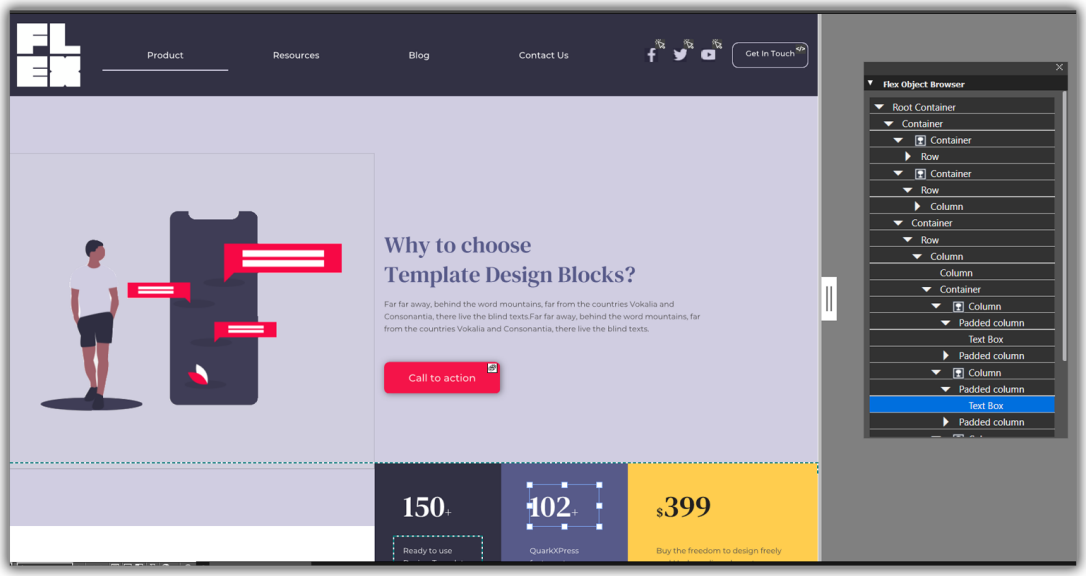
Delete: To delete a box, just use delete command from contextual menu, or select Item tool in the tools' palette, and press delete key to delete a selected box.

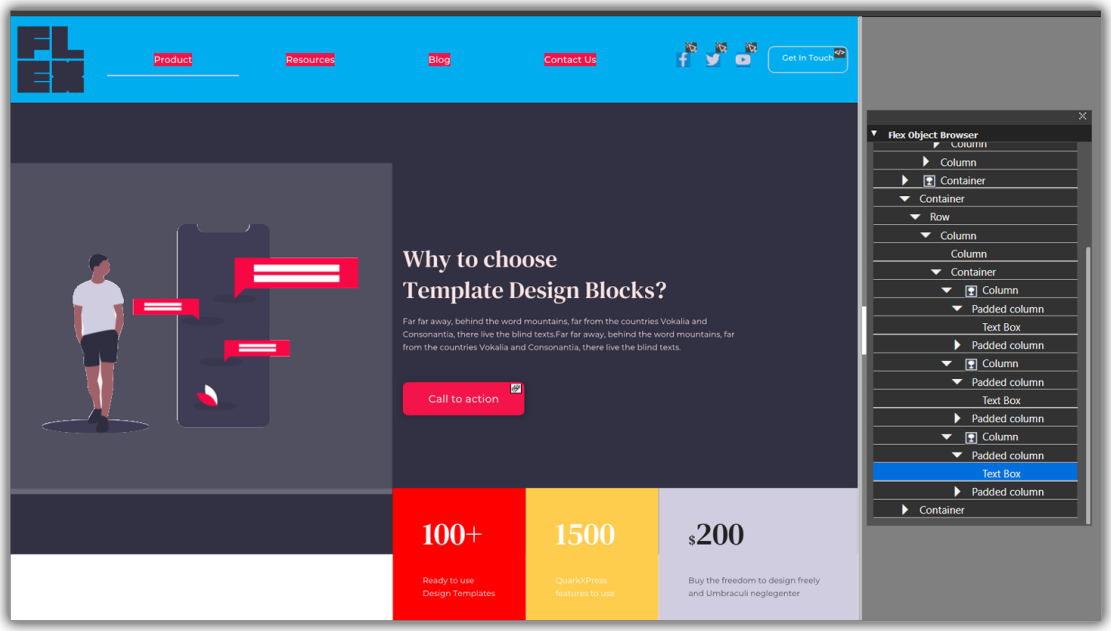
Duplicate: In case you want to have more similar components in a section, select the component you want to duplicate and use this command. This will also duplicate the component child hierarchy.

Flex Object Browser Enhancements in Flex Layouts

Each container now has an icon with a background image/color assigned to it in the Flex object browser. With a background color or image, it is easier to identify and change the container.

1. In **Window Menu**, open the **Flex object browser**.
2. Now can change easily text, box, background color and other content by Flex object browser. For example below images.





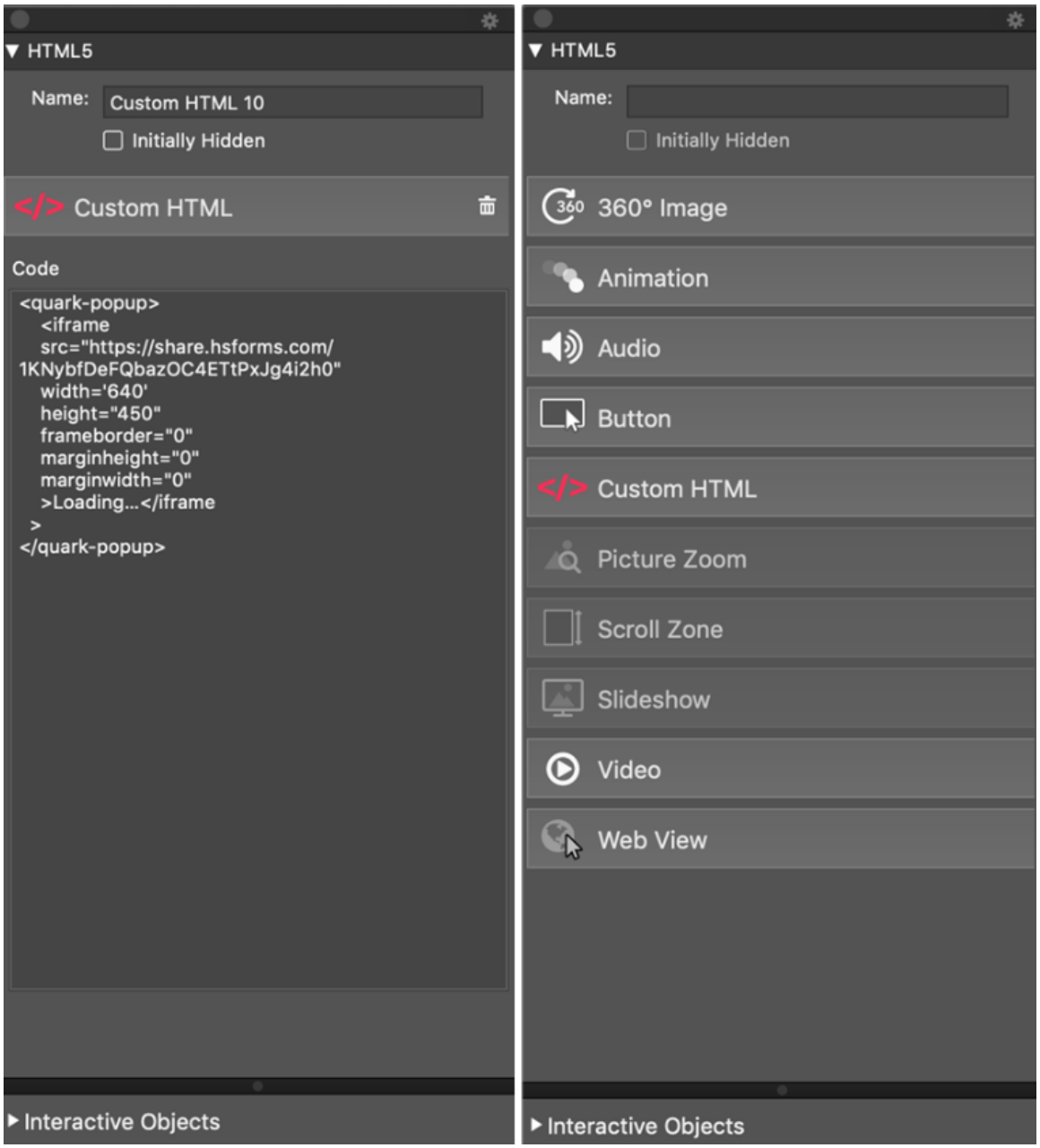
HTML Interactive Palette

In addition to the interactivities defined above for Fixed Layouts, we have introduced one additional interactivity for Flex Layouts. This new interactivity is called Custom HTML.

Custom HTML

In addition to other interactivities, there is a new interactivity that has been added for flex layouts, and this is called Custom HTML. Custom HTML allows

you to add custom HTML/CSS and JavaScript code to your layouts. With custom HTML code, now it is easy to incorporate different services to your layouts.



There are primary two different cases that can be solved with Custom HTML:

1. You can use it to integrate different third party services to your layouts. For example, you can easily incorporate Forms, Surveys, Quizzes, Calendars, Social Media buttons, Twitter Feeds, Facebook posts, etc.
2. You can write your custom code to achieve something that is not yet possible through QuarkXPress. As an example, you can use custom code to change the button colors while hovering etc.

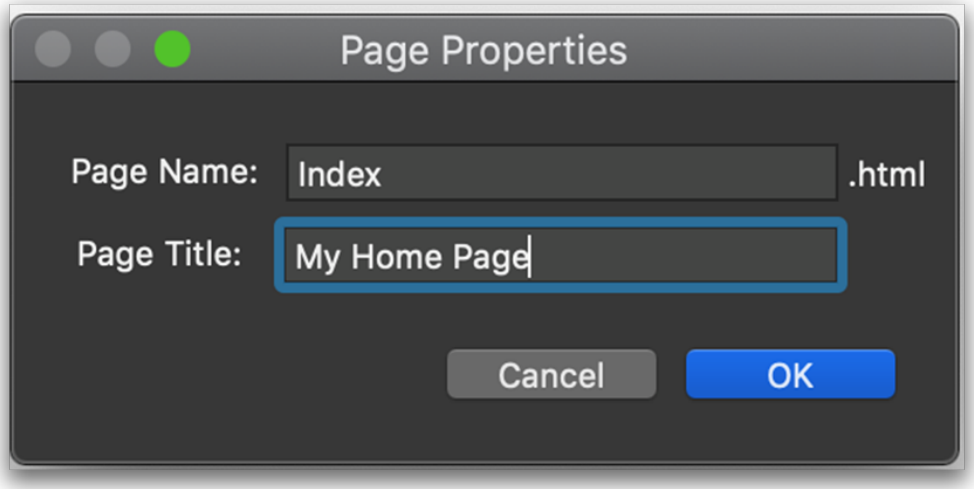
There are many pre-designed section blocks in the Flex Page Builder palette that uses this custom HTML interactivity to demonstrate various services that can be used. For example few of the blocks use Google Forms, Surveys from SurveyMonkey, Contact Forms from HubSpot, Social Media Buttons from addthis.com, and Google Calendar etc.

Support for Page Name and Page Title for HTML5 Output in Flex Layouts

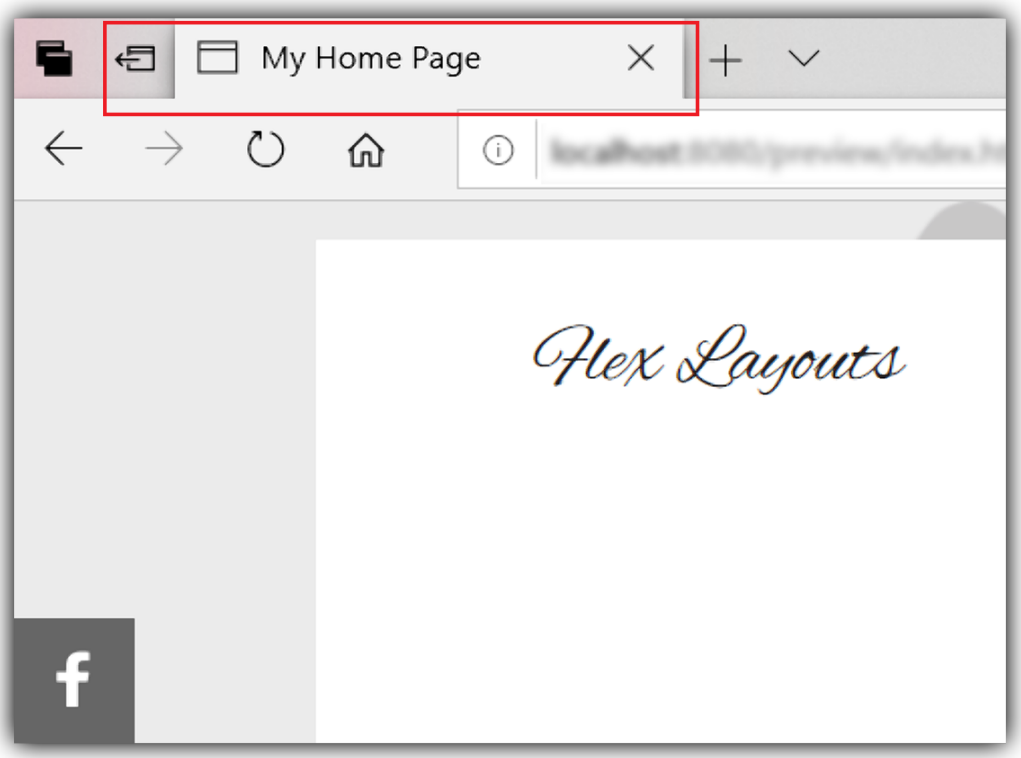
Using page properties, we can now write page name or page title for each page and also for the output page. This helps you in SEO optimization and better user experience in searches as well.

Let's see how to use this feature

1. In the **Window menu**, click **Page Layout**.
2. Right Click on the page properties. Now you can write or change the page name and page title.



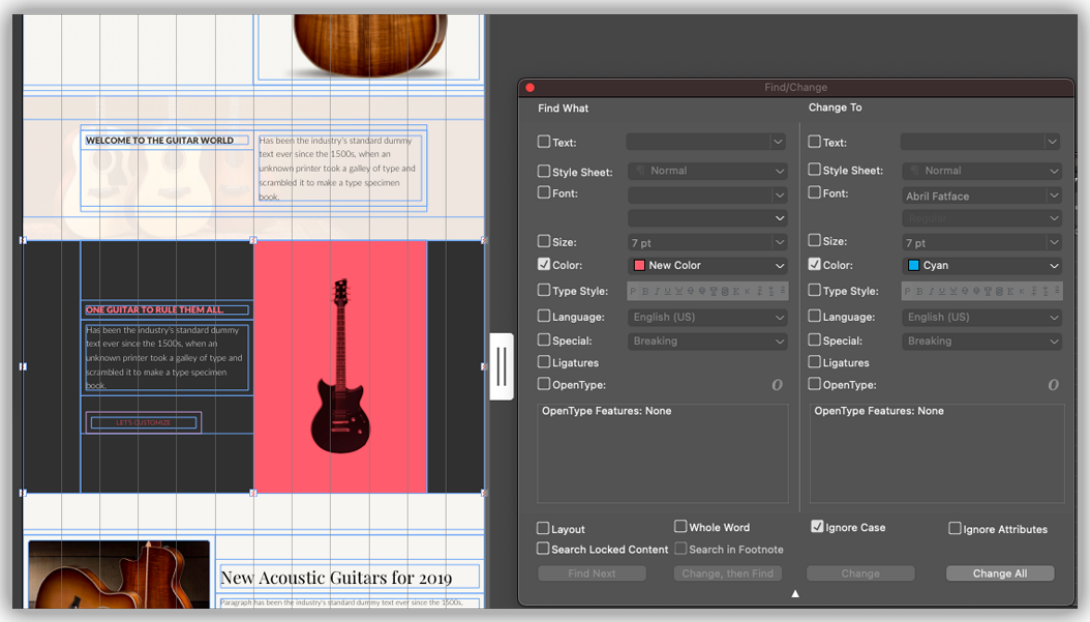
Check output preview - The given page name appears on the top of the page.



Find/Change Enhancement in Flex Layout

Find and Change functionality for text boxes and Layouts works in flex layouts exactly the way how it works in Print and Digital Fixed Layouts. Additionally, we can also use change functionality at flex container level without finding any item on Layout.

- Select any Flex container and open find and Change dialog
- Select the Attributes to be changed in find, and define required attributes in Change to



- You can notice that Change All is enabled on selection attributes in find and Change. Click Change All to replace all the instances in that flex container.

Hyperlinks Support

When we implemented hyperlinks for Flex Layouts in QuarkXPress2019, we allowed applying hyperlinks only onto the text. But now you can apply hyperlinks even on a box, including Text Box, Picture Box, None Box, Group box and Flex Container box.

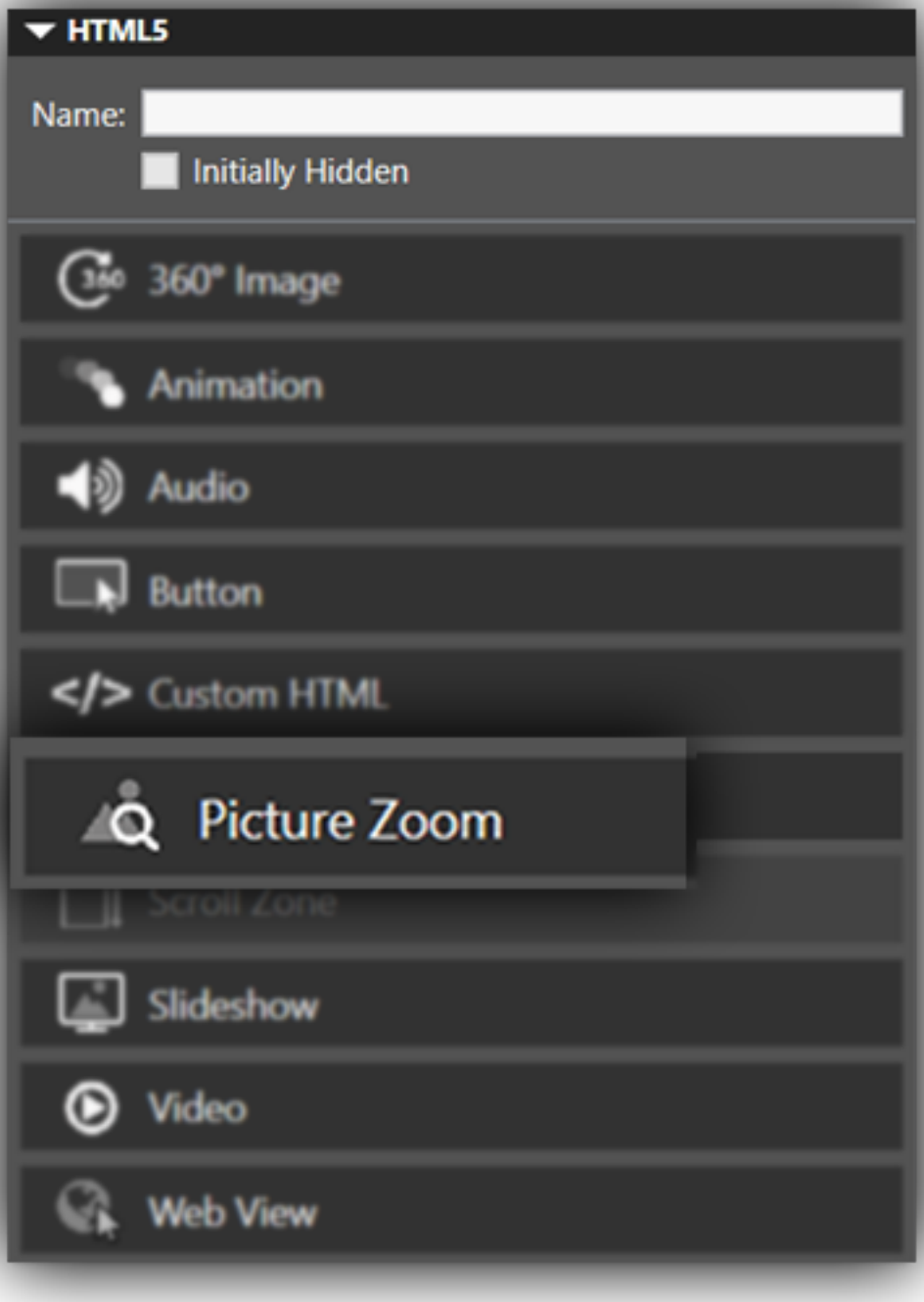
Interactivity Enhancements

Picture Zoom Interactivity in Flex Layouts

The Image Zoom interactivity option is available for Flex layout. This option enables the users to see the image zoom out or zoom in by using the pan/pinch.

We can use this option.

1 Choose **Window > HTML5**



2 In the **HTML5** palette, click **Picture Zoom**

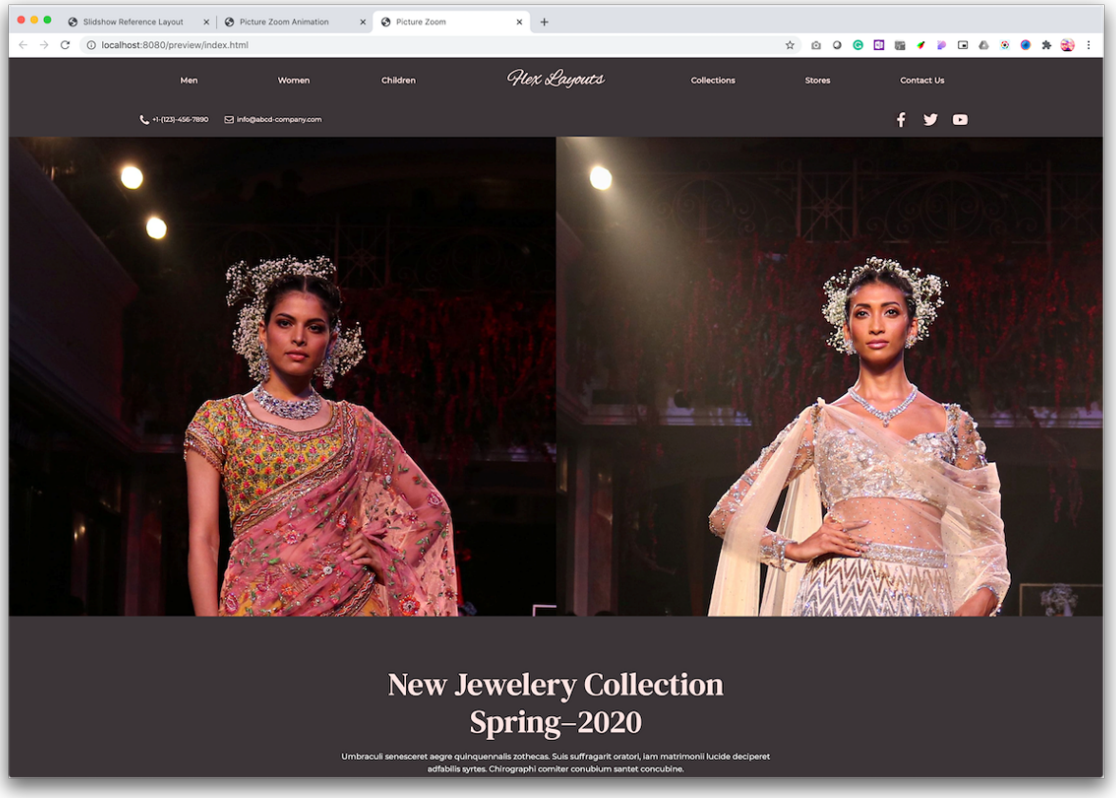
We have four options under picture to select –

1 **Allow fullscreen:** After applying this option you can see image on full page.

2 **Allow pinch zoom:** Pinch-to-zoom refers to the multi-touch gesture that zooms in or out of the displayed content.

3 **Allow panning:** This option allows panning of image.

4 **Animation pan and zoom**: This enables panning and zoom with animation.

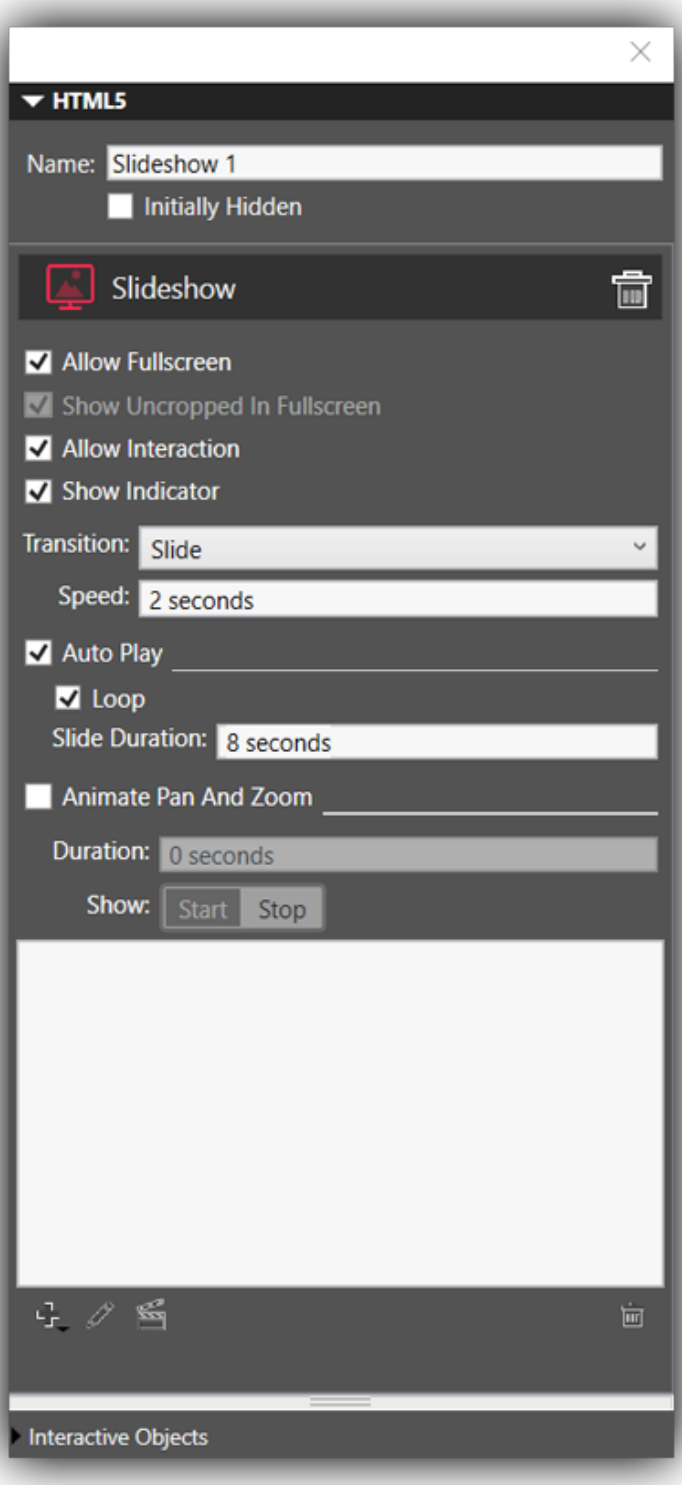


Slideshow Interactivity in Flex Layouts

Slideshows are also supported in Flex layouts that enable the user to create beautiful slideshows with photos or referenced layouts. These stunning slideshows can be used to introduce you business, advertising the special offers/deals, or creating stunning portfolios of clients.

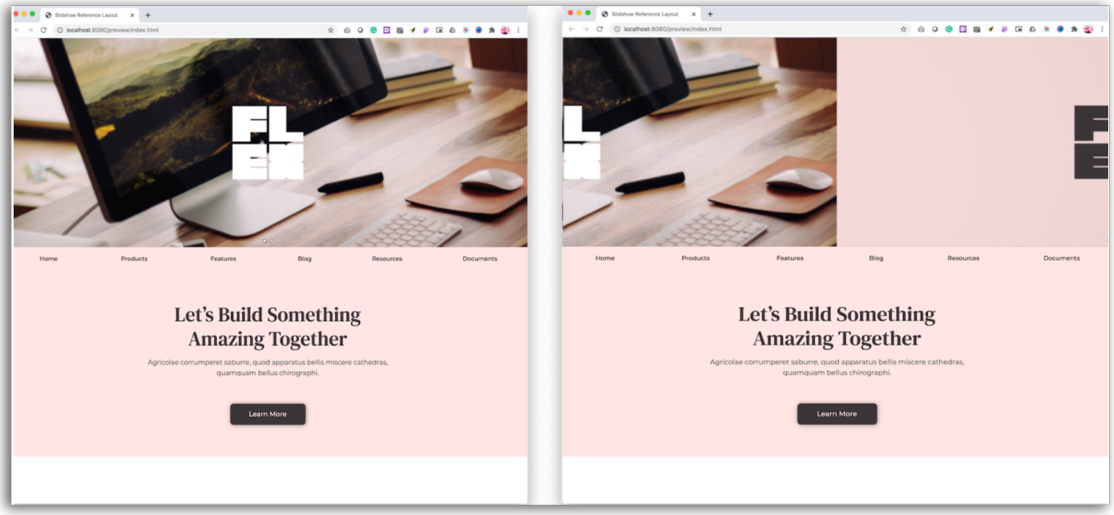
1 Choose **Window > HTML5**

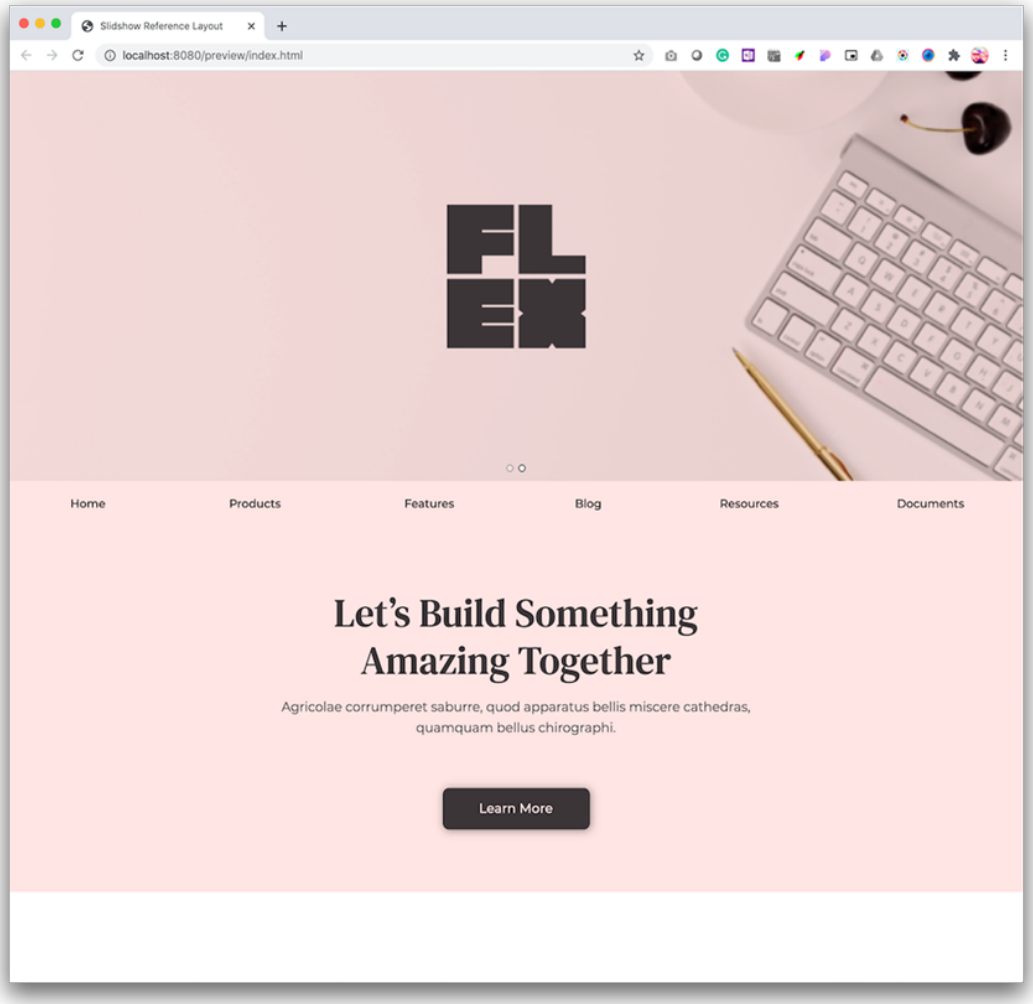
2 In the **HTML5** palette, click **Slide show**. You can apply slideshow on any object like text, picture etc.



We get multiple options to select under this feature, like –

- **Allow fullscreen:** You can open any slide in full screen.
- **Allow interaction:** You can enable / disable the manual interaction.
- **Auto play:** Slideshow starts auto playing after opening.



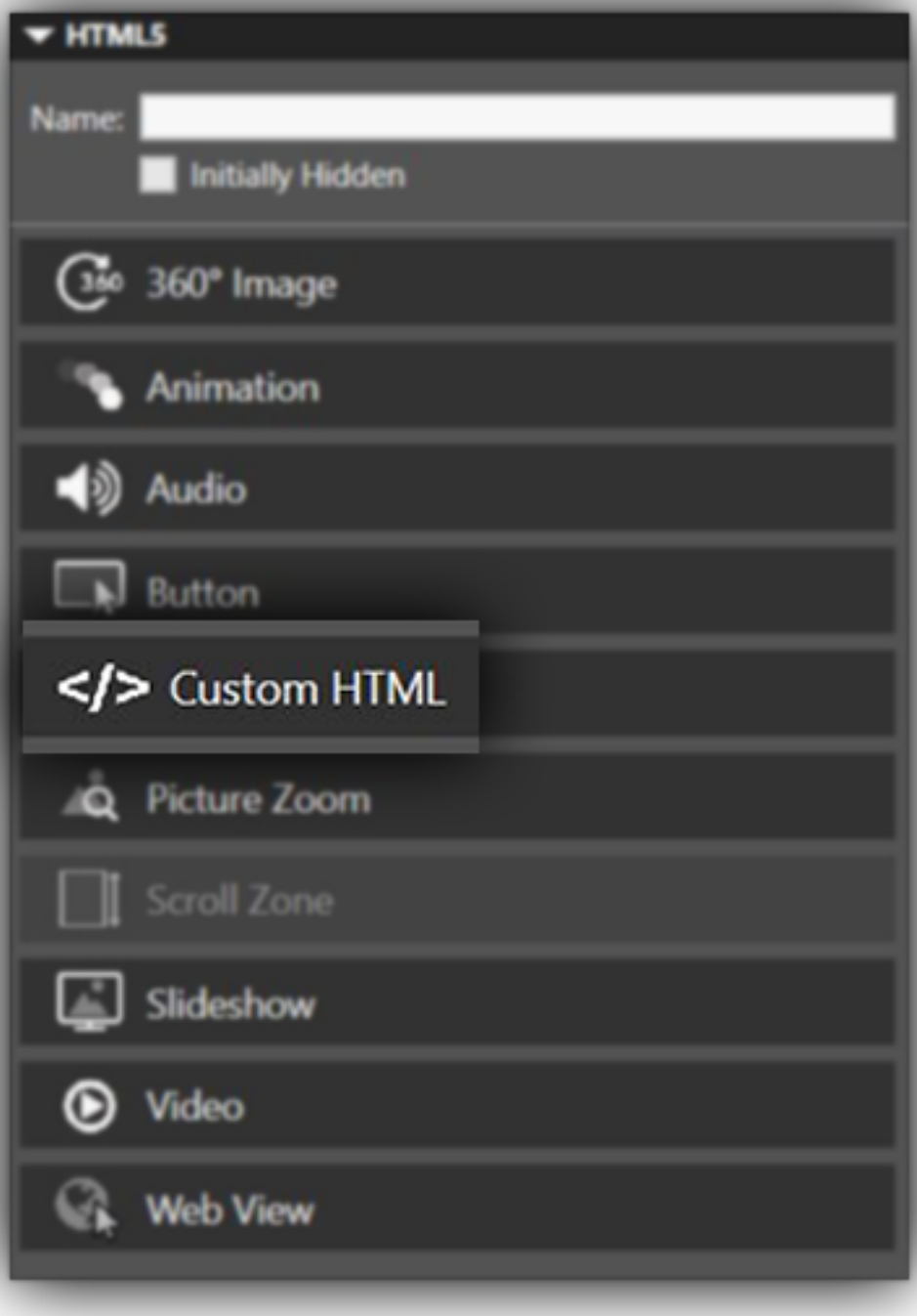


Custom HTML Interactivity in Fixed Layouts

You can now add custom code to your fixed layouts. Custom HTML allows you to insert custom HTML, CSS, and even JavaScript code practically anywhere on

your publication. If you need to add a third-party widget or Interactivity, now you can do it with embed custom code. Custom HTML is available as an additional interactivity on HTML5 palette.

1 Choose **Window > HTML5**



2 **In the HTML5 palette, select the object and then click Custom HTML. For example, let's create a Google form on button "Book Now".**

3 **After completing the required HTML 5 coding, we click on HTML Page Preview and see how it is going to be actually visible after the user action.**

Usability Enhancements

There are several usability enhancements that we have done for Flex Layouts.

Improvements in Paste

Now, you can copy any flex item from one container and paste it directly into the second container by using paste options. Whereas in the earlier version, if the user wants to copy any flex item from one flex container to another container, he had to copy-paste it from the first flex container and then had to explicitly attach it to the second container. You can also paste a flex item before or after any selected flex item within the flex container by using the Paste Before and Paste After menu options.

Changing the Flex Item Order

In the earlier version, the only way to change the order of a flex item within a container was to drag out that flex item and insert it again at the desired location. Now, you can rearrange child items within the flex container by using Shortcut keys (Ctrl or Command Up/Ctrl or Command Down) or Move Up, Move Down menu available in Item Menu.

Adding Layout Blocks by double-clicking them

You can add Layout Blocks on a layout by simply double-clicking them in the Layout Blocks Palette. They will get attached to the selected container box automatically if any container box is selected on the layout or else layout blocks will be placed on the layout.

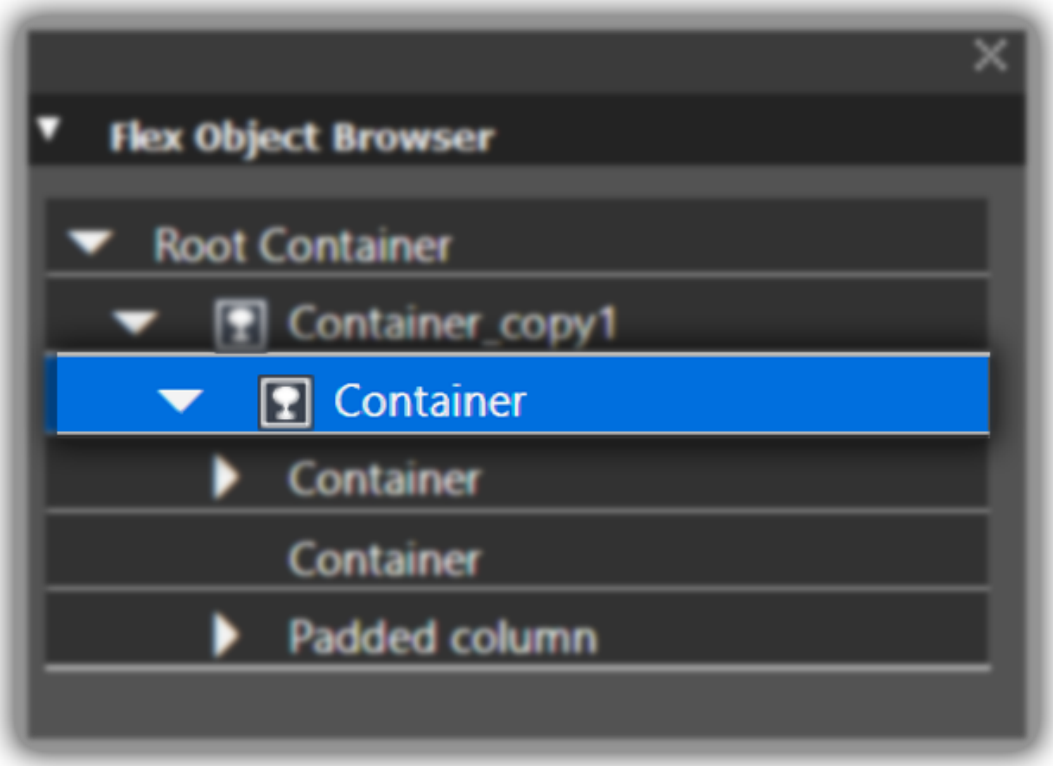
Support for Animated GIF

SUPPORT FOR ANIMATED GIF IN FIXED AND FLEX LAYOUTS

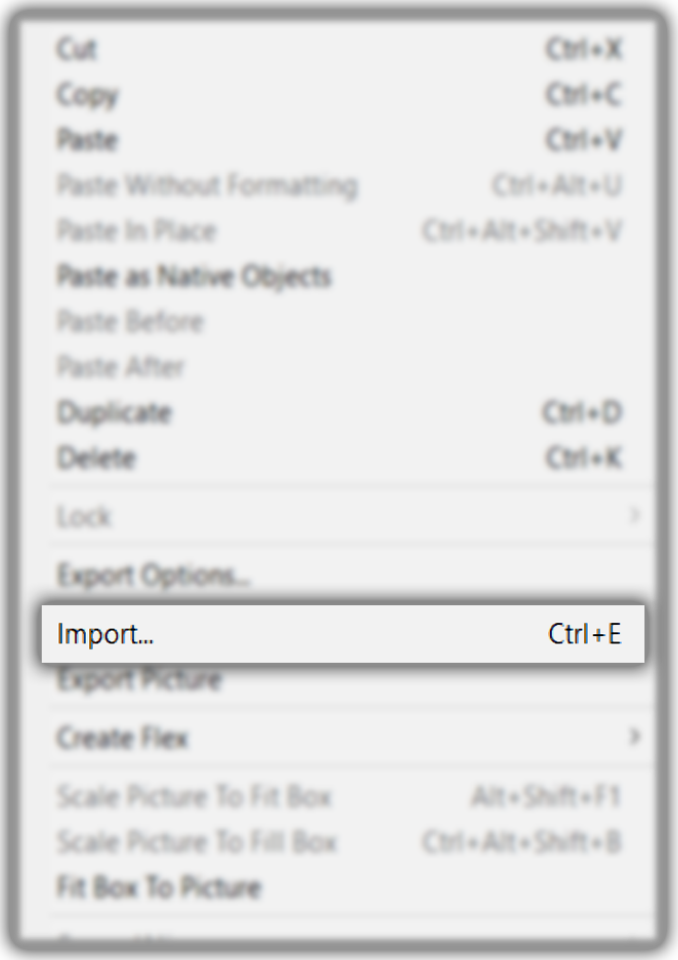
Animated GIFs can be used at the backdrop of a Picture box or a Flex container. Animated GIFs are a fantastic way to spice up your style. Their movement can easily attract attention and add variety to an otherwise static picture.

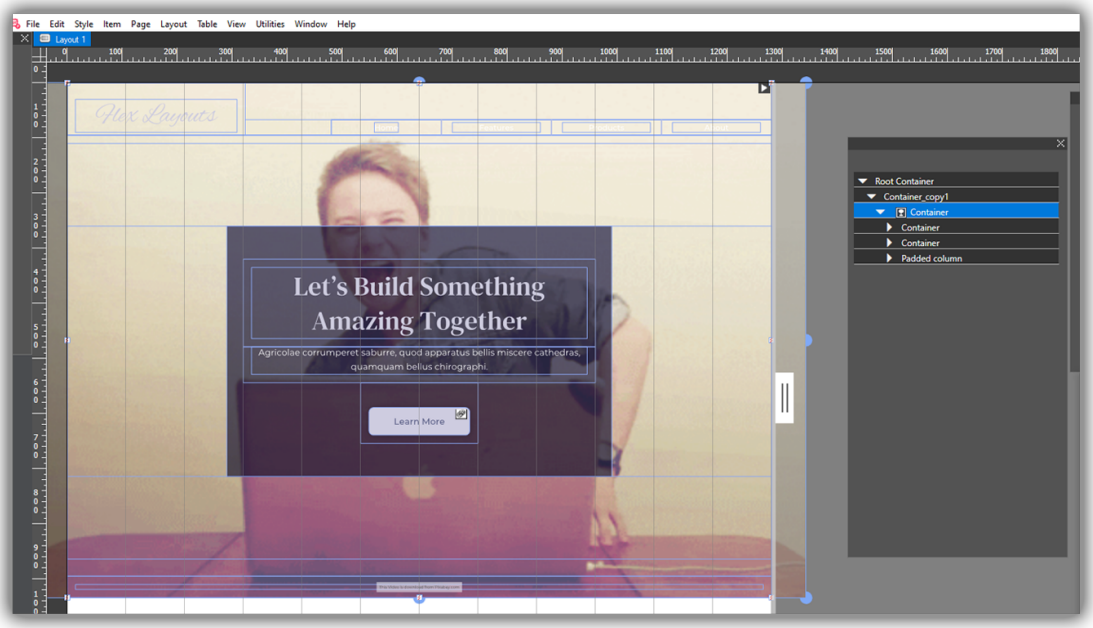
1 Go to **Window > Flex Object Browser**.

2 Go to Root Container> Right Click on Container

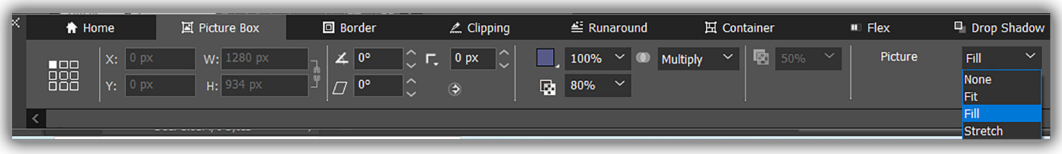


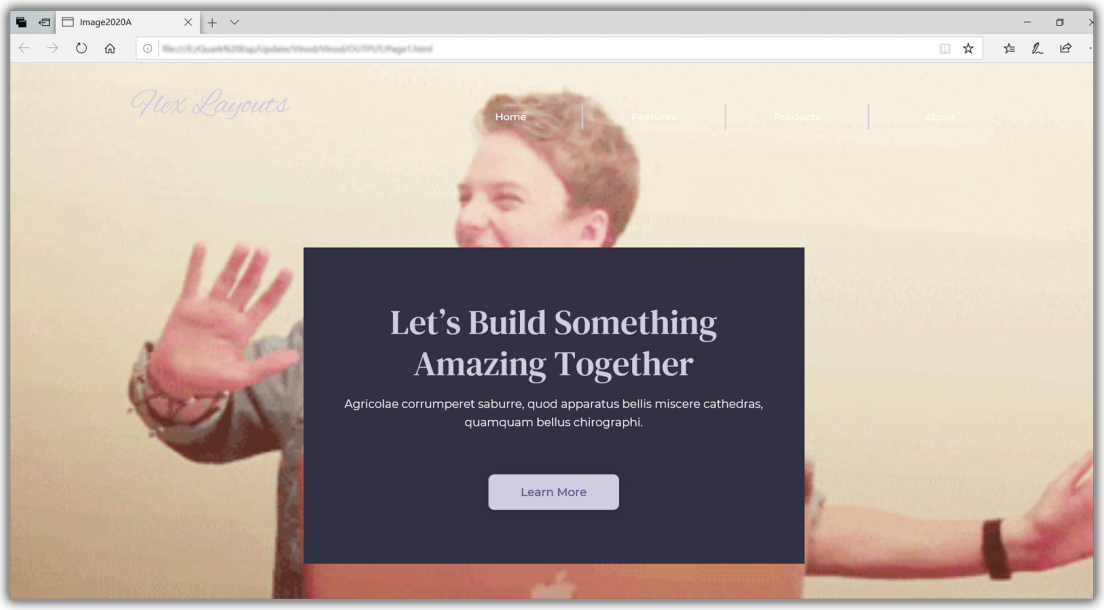
Then use to Import (CTRL+E) option. Now choose animation GIF file and import it.


Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Without Formatting Ctrl+Alt+U
Paste In Place Ctrl+Alt+Shift+V
Paste as Native Objects
Paste Before
Paste After
Duplicate Ctrl+D
Delete Ctrl+K
Lock
Export Options...
Import... Ctrl+E
Export Picture
Create Flex
Scale Picture To Fit Box Alt+Shift+F1
Scale Picture To Fill Box Ctrl+Alt+Shift+B
Fit Box To Picture


File Edit Style Item Page Layout Table View Utilities Window Help
Layout 1
Flex Layouts
Let's Build Something Amazing Together
Agricolae corrumperet saburre, quod apparatus bellis miscere cathedras, quamquam bellus chirographi.
Learn More
Root Container
Container_copy1
Container
Container
Container
Padded column

Gifs will get exported as Animated only incase if we apply Picture fitting is applied from measurement palette or else they will get exported as static Gifs.

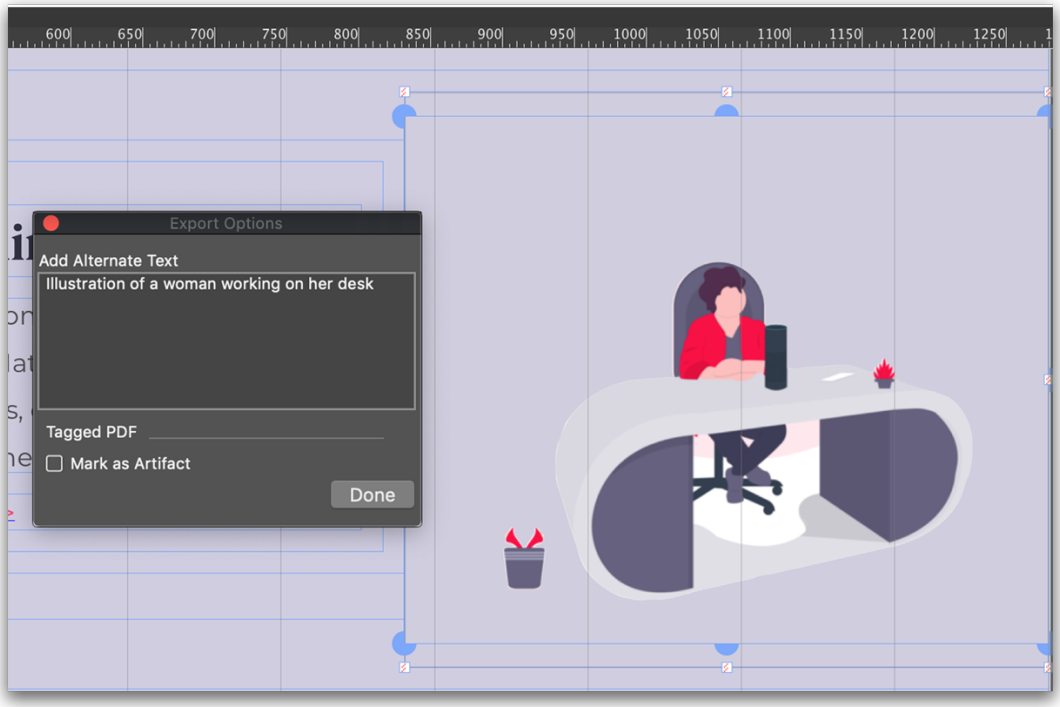




Support for Alternate Text for Images in Fixed and Flex Layouts

Alternate text is referred as alt text. It's an HTML attribute that gets attached to an image tag. This makes it easier for visitors who can't see the picture and search engine bots to find out what it's about.

1. Right-click on the window by selecting an image then choose the export option, a dialog box for alternate text will appear, add alt text then, click on the Done button to apply changes.



Creating a TOC for ePub

When you export a reflow article as an ePub, you can create the table of contents for that eBook in two ways:

- Use the order of the reflow articles, with one TOC entry for each reflow article. You can change the order of the reflow articles using the **Articles** palette (**Window** menu). To move an article up or down in the sequence, select the article in the **Articles** palette and click the **Move Up** or **Move Down** button. To update the table of contents, click the icon and choose **Update**.
- Manually create a list to define the table of contents, see “Working with Lists” in *A Guide to QuarkXPress*.

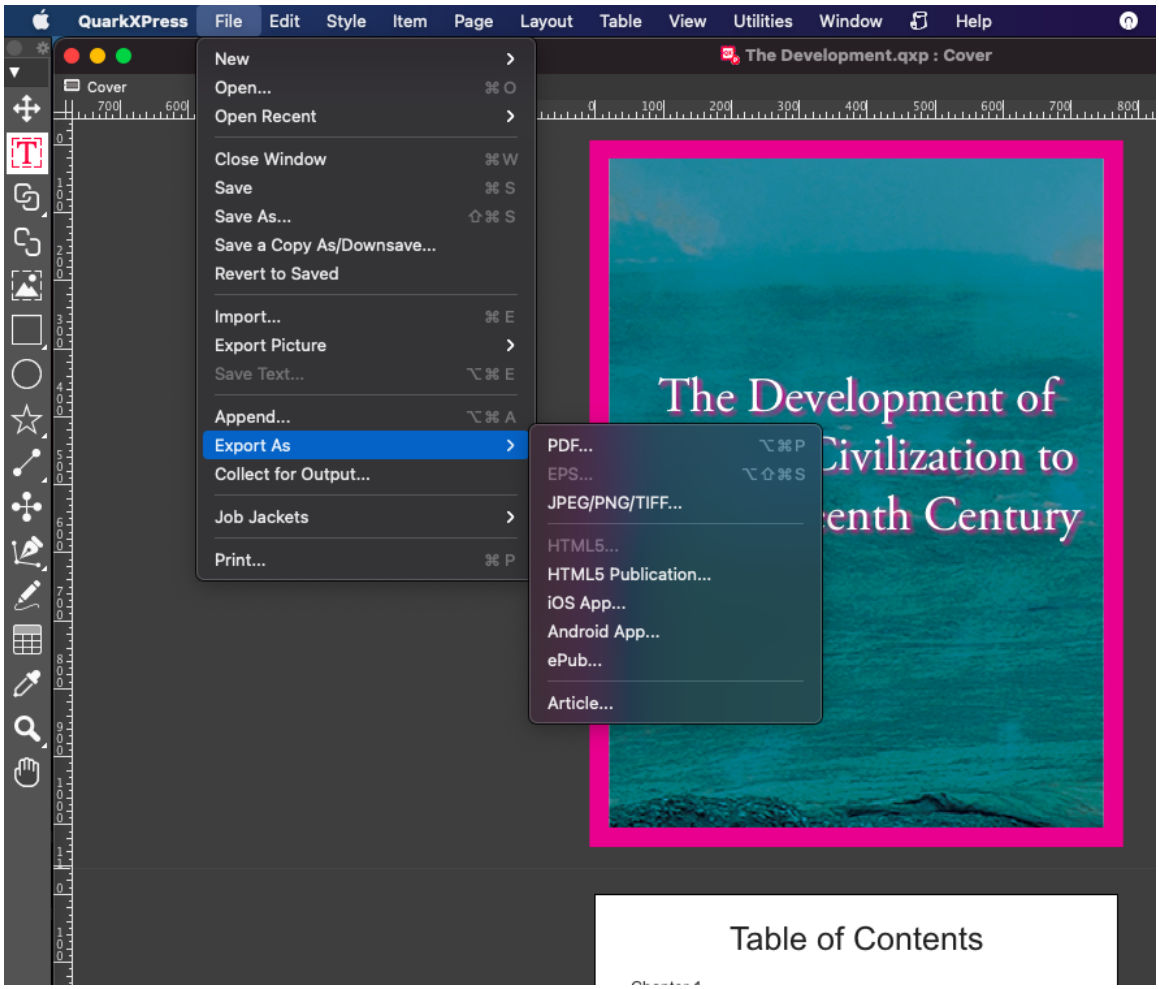
Content based TOC in Reflow e-Pubs

In QuarkXPress 2021, the customers can create content based TOC in reflow e-Pubs similar to in the Fixed e-Pubs.

To add content based TOC in the reflow e-book, user should create a TOC list on Layout and include that box in Article Palette's article. The created TOC list will not have page numbers in the reflow ePubs so users should not include page numbers when creating this list for reflow. Formatting to this content based TOC can be applied using style sheets.

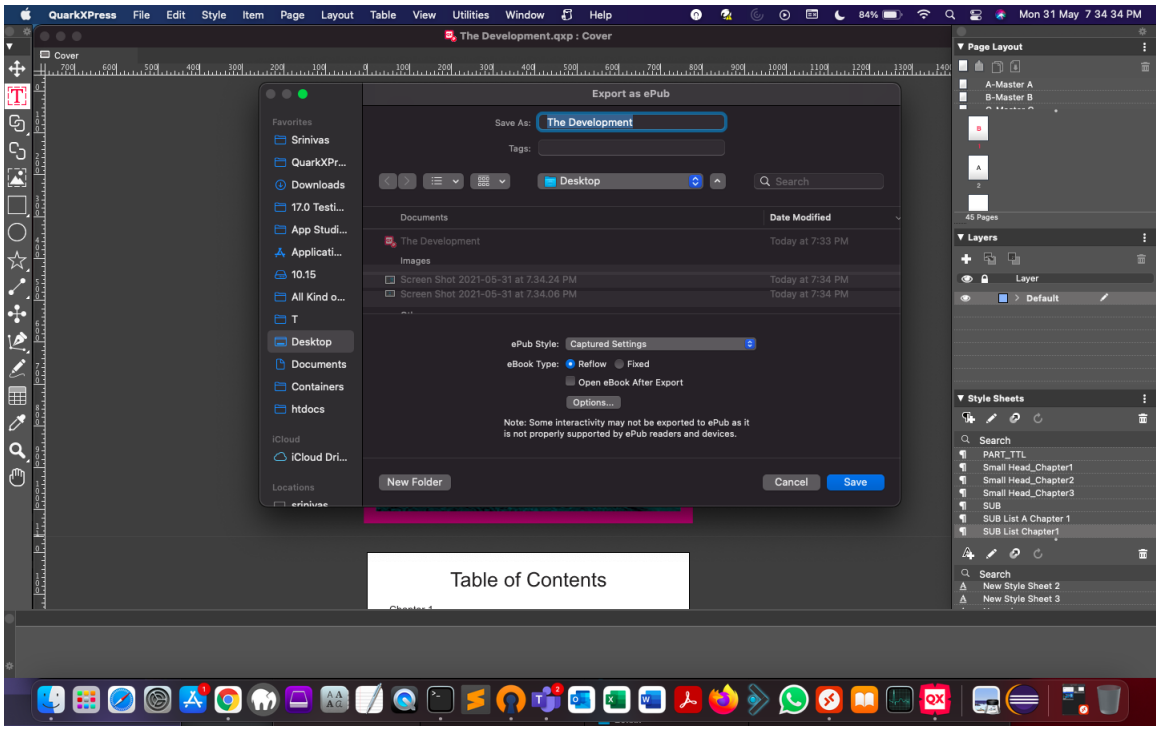
Users can create a master TOC, TOC based on sections (e.g. chapters of the book) and multiple content-based TOCs in an e-book.

- **Go to File Menu > Export As > e-pub.**

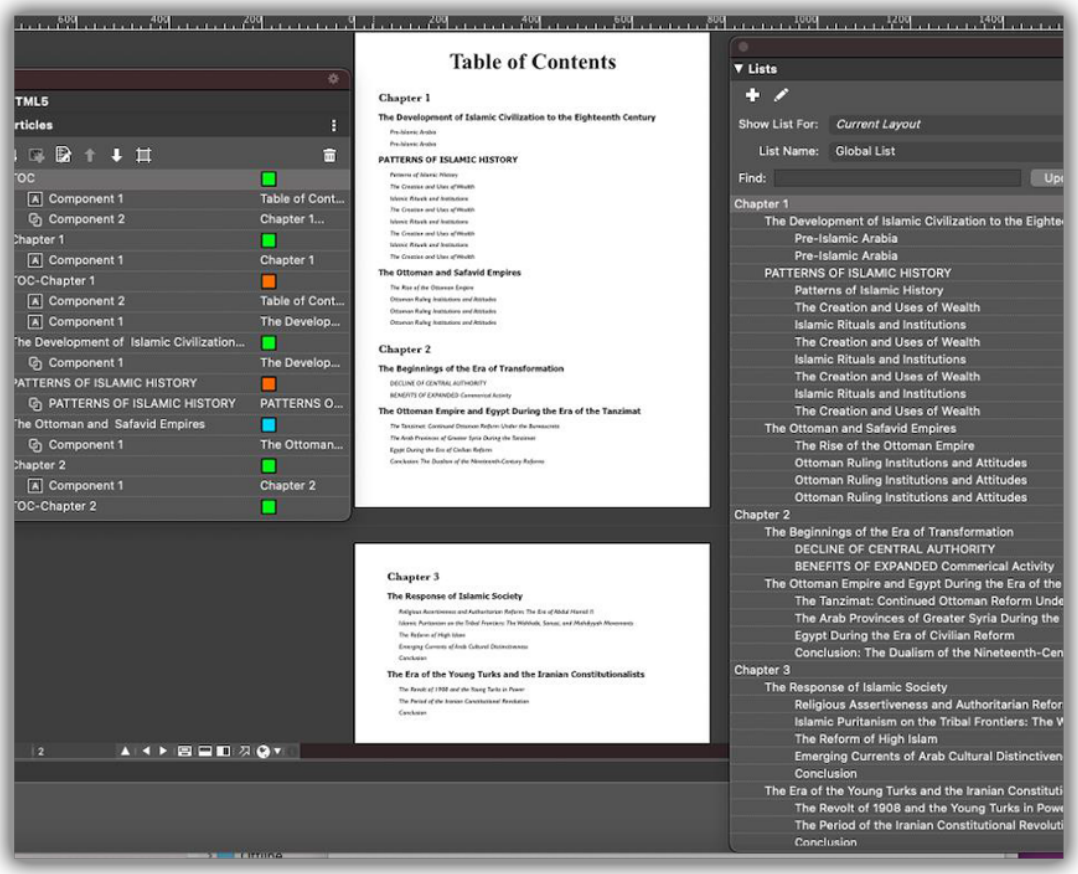


Export as ePub Menu

- **Choose Reflow in eBook Type option.**

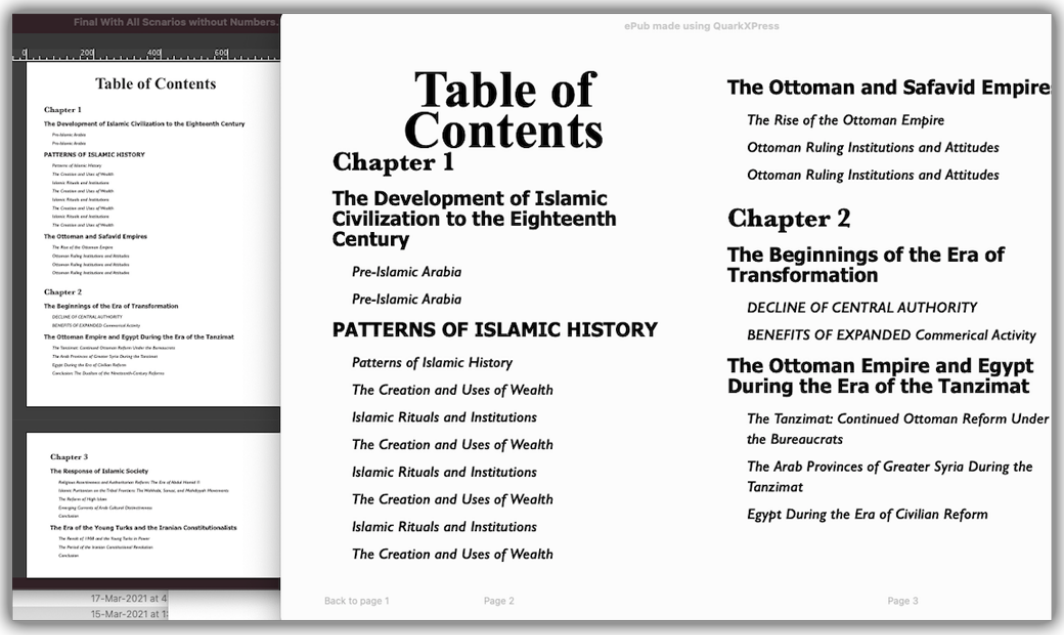


Selecting Reflow option in Export as ePub Dialog



Created TOC in Layout

Output Preview:



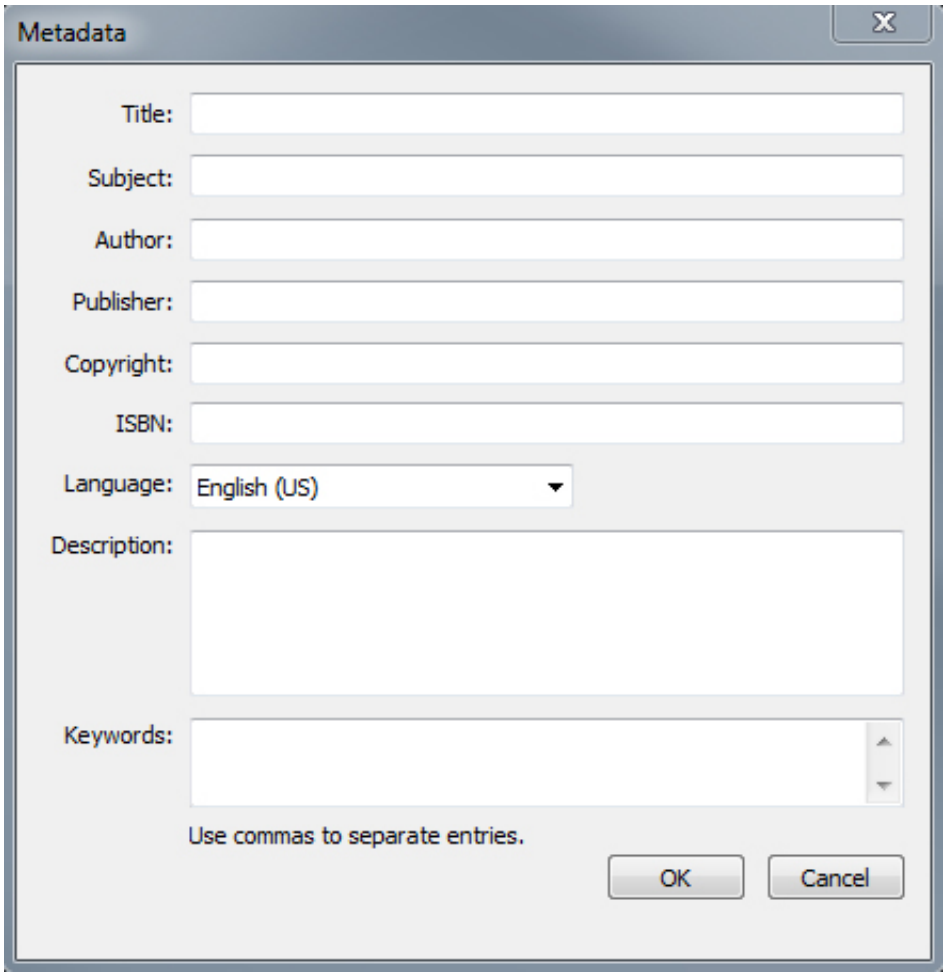
TOC in ePub readers

Working with metadata

It's a good idea to associate some metadata with the output you create, so that users can easily find and get information about them. The layout metadata is applicable to all formats (e.g. PDF, ePub, and HTML).

To add or edit metadata for a layout:

1. Display the layout in layout view.
2. Choose **Layout > Metadata**. The **Metadata** dialog displays.



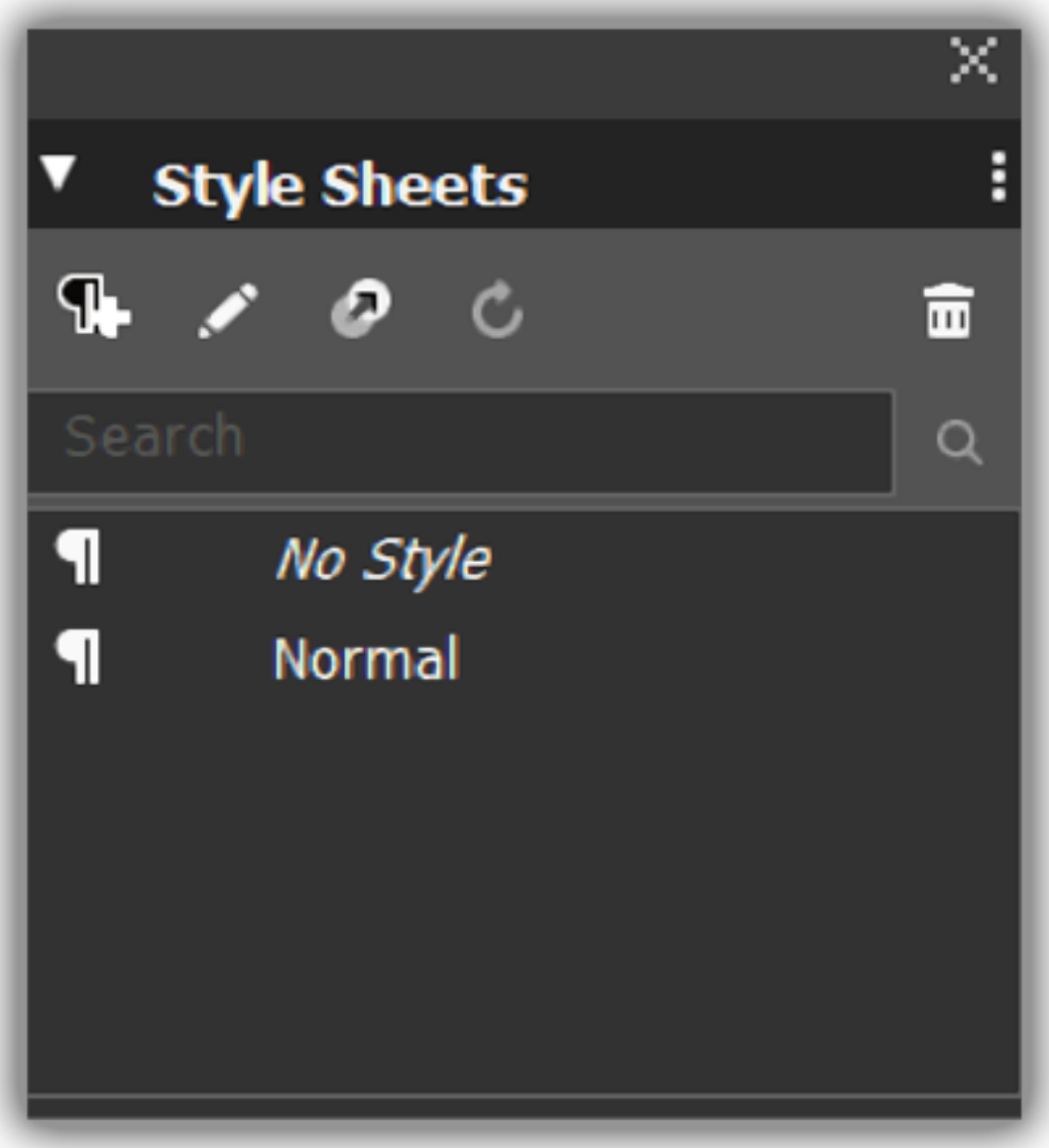
Metadata dialog

- Use the fields in this dialog box to enter information about your eBook:
  - In the **Keywords** field, use a comma to separate keywords.
  - Use the **Language** drop-down menu to specify the language of your output. This can be used by readers to search for the correct book.

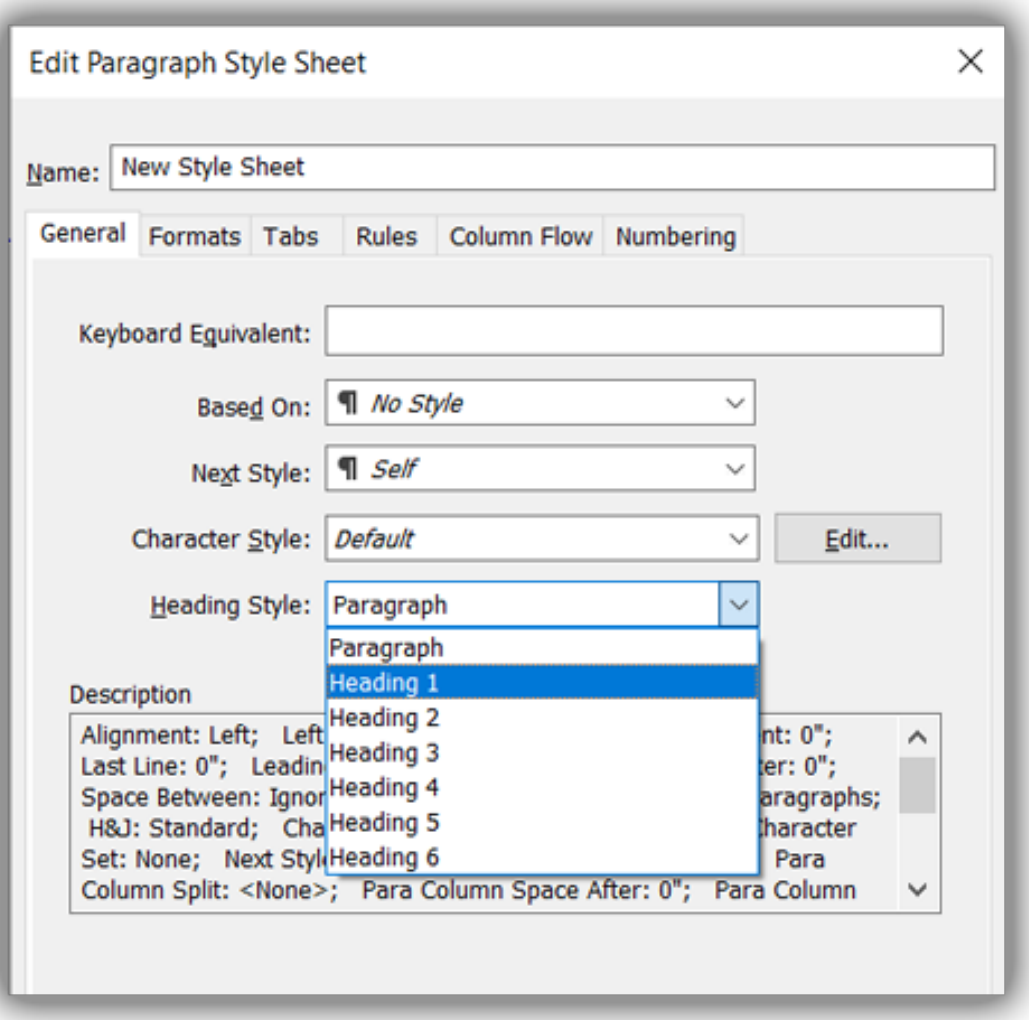
➡ If you don't specify any title meta data here, QuarkXPress will add the following default text to the exported output: “Created with QuarkXPress”. This will automatically be removed as soon as you add your own title.

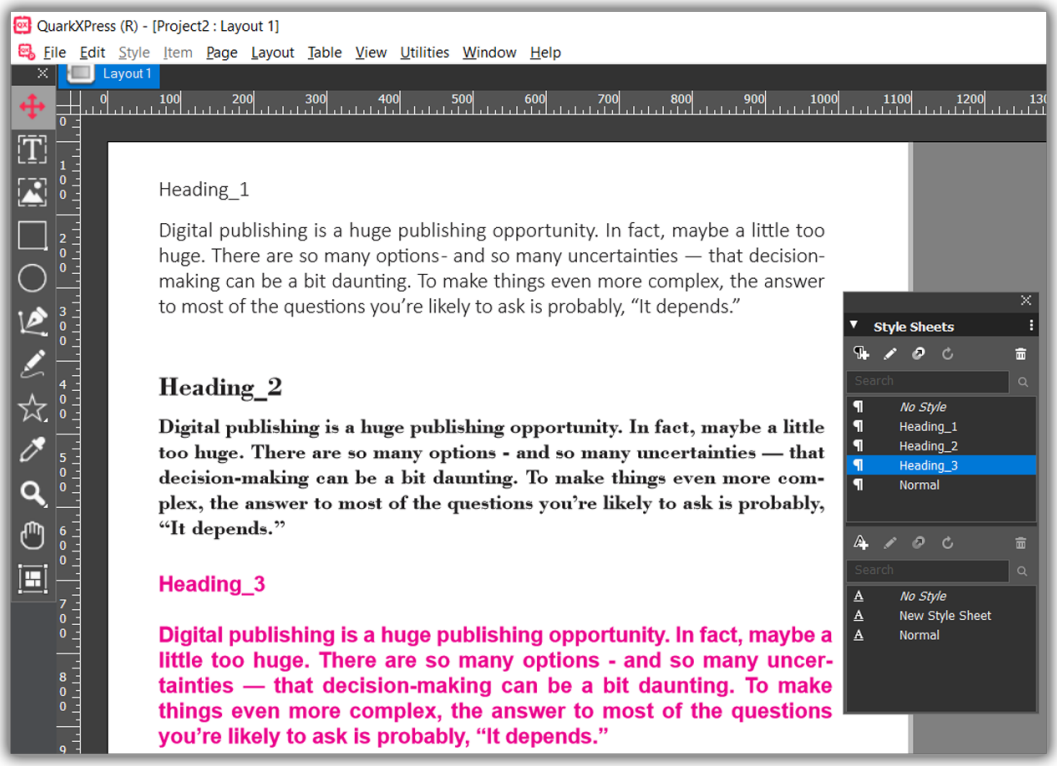
Support for Semantic Headers

QuarkXPress provides support for Semantic headers as well. The semantic tags allow the user to have SEO-friendly outputs by adding the same into headers and other elements. The use of different styles and effects such as Title, Header, Sub-Titles, etc. helps search engines to read and understand the content and its

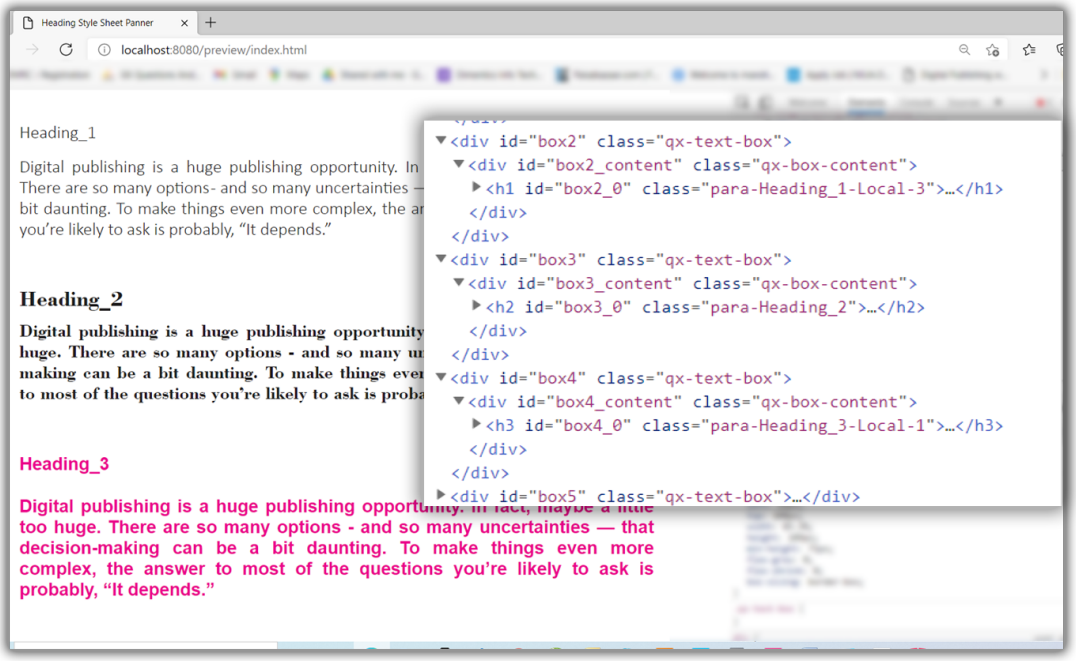


2 **Click on** **.**


Edit Paragraph Style Sheet
Name: New Style Sheet
General
Formats
Tabs
Rules
Column Flow
Numbering
Keyboard Equivalent:
Based On: ¶ No Style
Next Style: ¶ Self
Character Style: Default
Edit...
Heading Style: Paragraph
Paragraph
Heading 1
Heading 2
Heading 3
Heading 4
Heading 5
Heading 6
Description
Alignment: Left; Left
Last Line: 0"; Leadin
Space Between: Ignor
H&J: Standard; Cha
Set: None; Next Styl
Column Split: <None>; Para Column Space After: 0"; Para Column



Output preview.

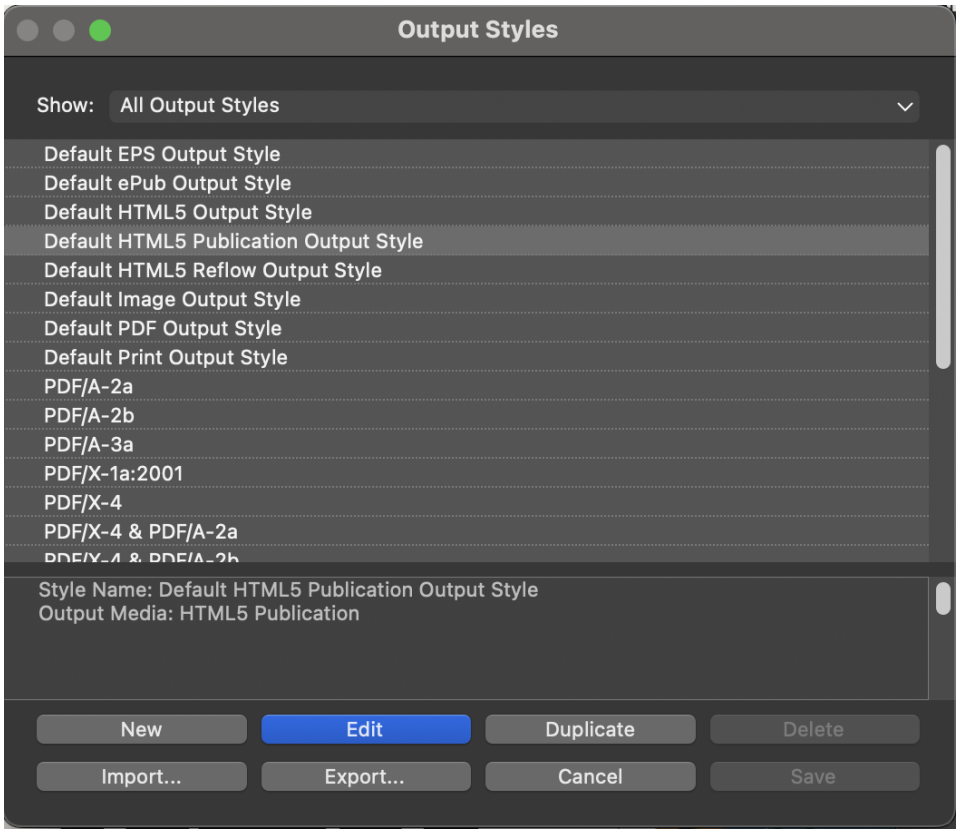


Working with Output Styles

Output styles let you capture settings for output in print, PDF, ePub, and HTML5 Publications . You can use output styles when using the File > Print, File > Export as > PDF, File > Export as > HTML5 Publication, File > Export as > EPS, File > Export as > ePub. QuarkXPress includes default settings for all output options, which can serve as a basis for you to customize according to need. Or, you can create output styles from scratch.

To create an output style:

1 Choose **Edit > Output Styles**. The **Output Styles** dialog displays.



Use the Output Styles dialog to create, import, export, edit, and remove output styles.

2 Choose an option from the **New** drop-down menu.

3 Enter a name for your style in the **Name** field.

4 Specify settings in the panes.

➡ For information about EPS and PDF publication options, see "*A Guide to QuarkXPress*". For more information about ePub options, see "*Exporting for ePub*".

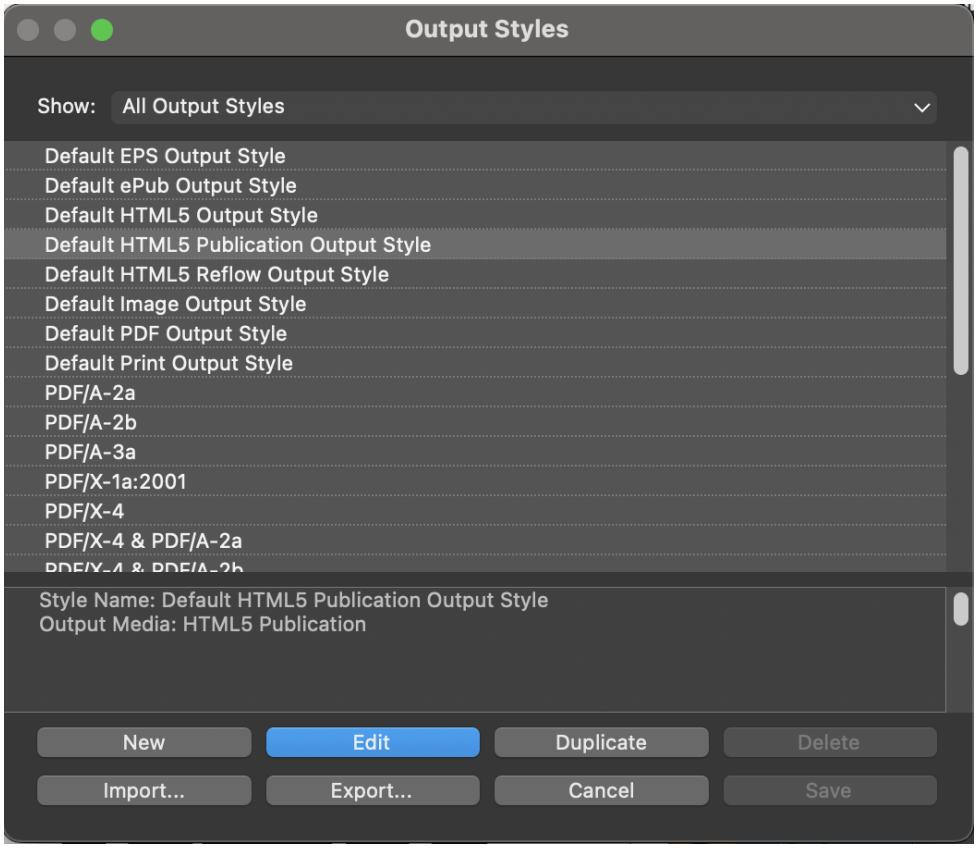
5 Click **OK**.

6 Click **Save**.

HTML5 Publication output styles

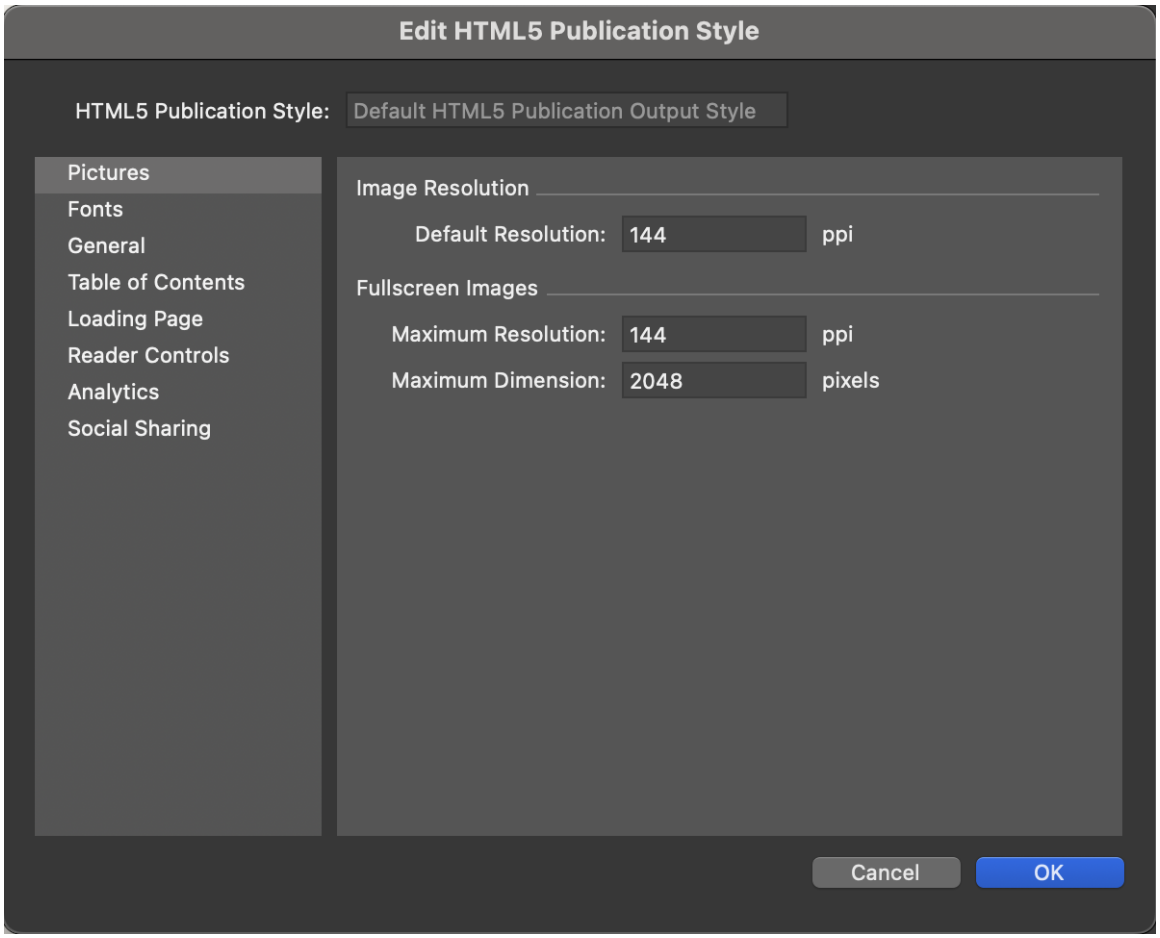
To specify the settings for HTML5 Publication output styles:

1 Choose **Edit > Output Styles**. The **Output Styles** dialog displays.



Output Styles dialog

- To edit an HTML Publication output style, choose the output style and click **Edit**. To add a new output style, select **HTML Publication** from the **New** drop-down menu. The **Edit HTML5 Publication Style** dialog displays.



Edit HTML5 Publication Style dialog

2 Enter a name for your style in the HTML5 Publication Style field.

- Use the **Pictures** pane to set default resolution for images exported for App. Choose maximum resolution and/or dimensions for converted/down sampled images (only applies to images used in HTML5 interactivity).
- Use the **Fonts** pane to enable/disable packaging referenced fonts in the App. If disabled, the fonts will NOT be collected and the output may not be WYSIWYG.

If enabled, users will get an alert about font licensing.

- Use the **General** pane to enable/disable converting QuarkXPress Layout Sections to Page Stacks. Enable/disable “Continuous Page Stacks” to create a single continuous page of all pages in a section. You can also set the Lock Orientation for Desktop and Mobile devices from this tab.
- Use the **Table of Contents** pane to set options for the table of contents in the HTML5 Publication file.
- Use the **Loading Page** pane to specify the appearance and behaviours of the loading page in a web reader.

- Use the **Reader Controls** pane to specify the appearance and behaviours of the layout in a web reader.
- Use the **Analytics** pane to enable/disable Google analytics and define a Google maps API key for geolocation.
- Use the **Social Sharing** pane to define the settings for social medium platforms.

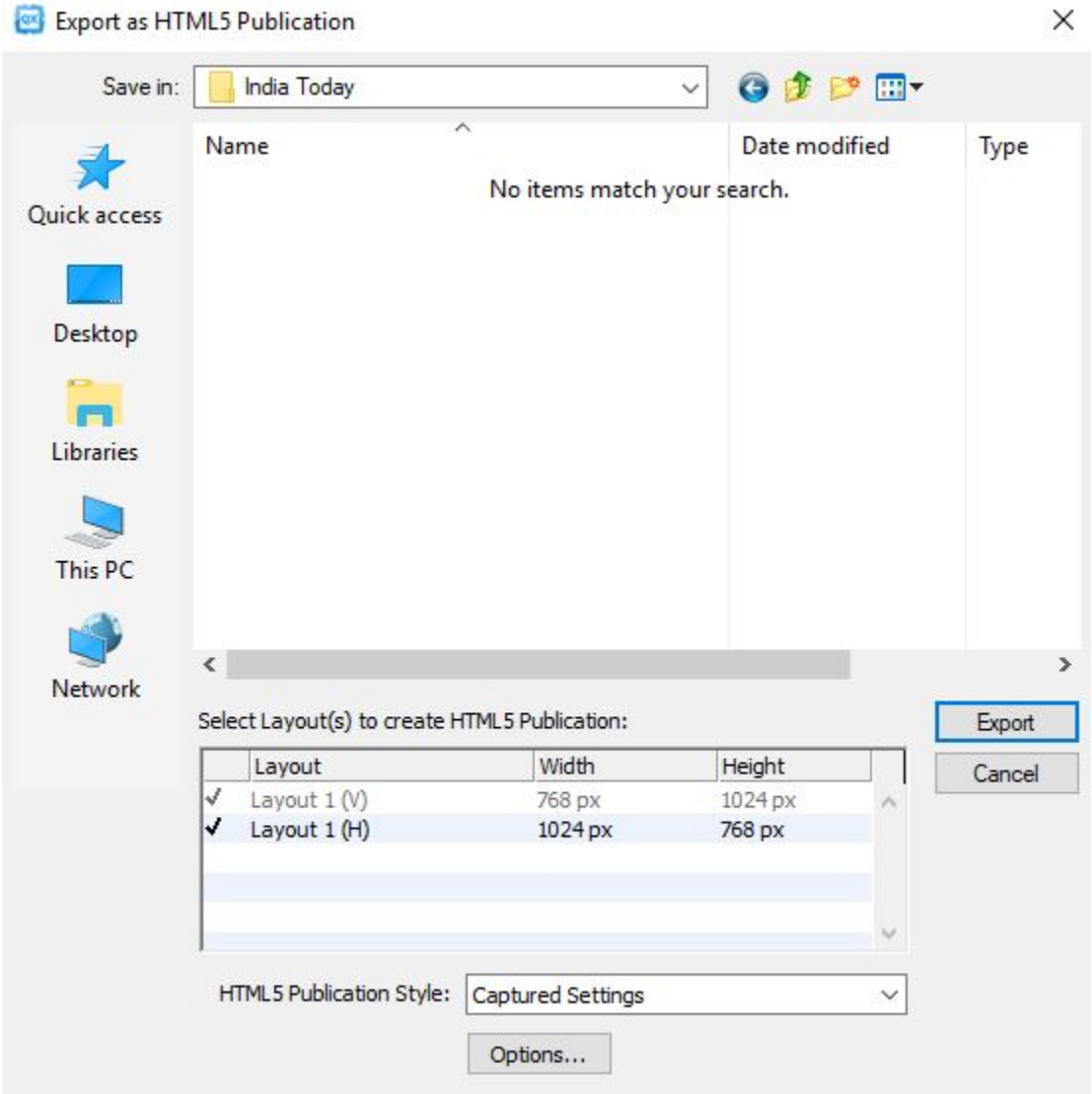
3 Click **OK**.

HTML5 Publications

Exporting as HTML5 Publication

To export a layout as HTML5 Publication:

1 Choose **File > Export as > HTML5 Publication**. The **Export as HTML5 Publication** dialog displays.

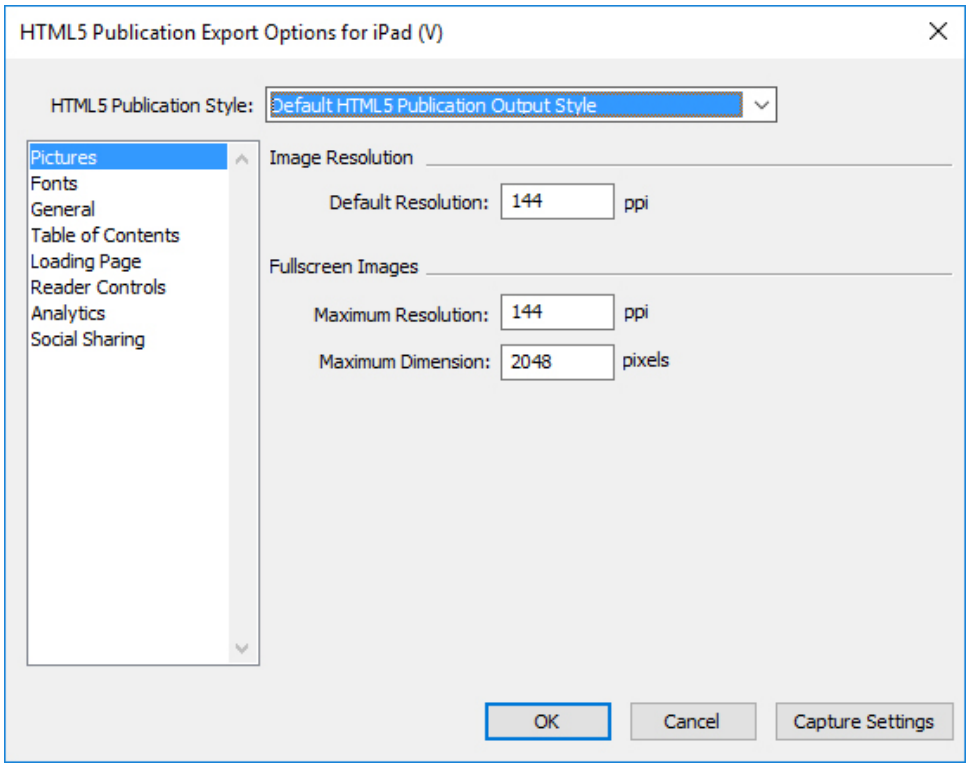


Export as HTML5 Publication dialog

2 Navigate to the folder you want to save the HTML5 Publication to.

3 Select the layout(s) you want to use to create the multi-device HTML5 publication.

4 Choose a predefined output style from the **HTML5 Publication Style** drop-down menu or click **Options** to define the export settings for the selected layout(s). If you click **Options**, the **HTML5 Publication Export Options** dialog displays.



HTML5 Publication Export Options dialog

➡ Output styles can be managed by selecting **Edit > Output Styles**. See "*A Guide to QuarkXPress*".

- Configure the output style using the following panes in the HTML5 Publication Export Options dialog.
  - Use the **Pictures** pane to specify the resolution of pictures in the HTML5 Publication file.
  - Use the **Fonts** pane to enable/disable packaging referenced fonts. If disabled, the fonts will NOT be collected and the output may not be WYSIWYG. If enabled, users will get an alert about font licensing.
  - Use the **General** pane to enable/disable converting QuarkXPress Layout Sections to Page Stacks in the App and to set the lock orientation. Enable/disable "Continuous Page Stacks" to create a single continuous page of all pages in a section.
  - Use the **Table of Contents** pane to select a named section from the list and all the pages in that section will be exported as TOC Pages. You may also choose to include pages in publication, otherwise these

pages will not be added to content pages. Specify a title for the TOC to be used in the TOC Banner.

- Use the **Loading Page** pane to specify the appearance and behaviors of the loading page in a web reader.
- Use the **Reader Controls** pane to specify the appearance and behaviors of the layout in a web reader.
- Use the **Analytics** pane to enable/disable Google analytics and define a Google maps API key for geolocation.
- Use the **Social Sharing** pane to define the settings for social medium platforms.

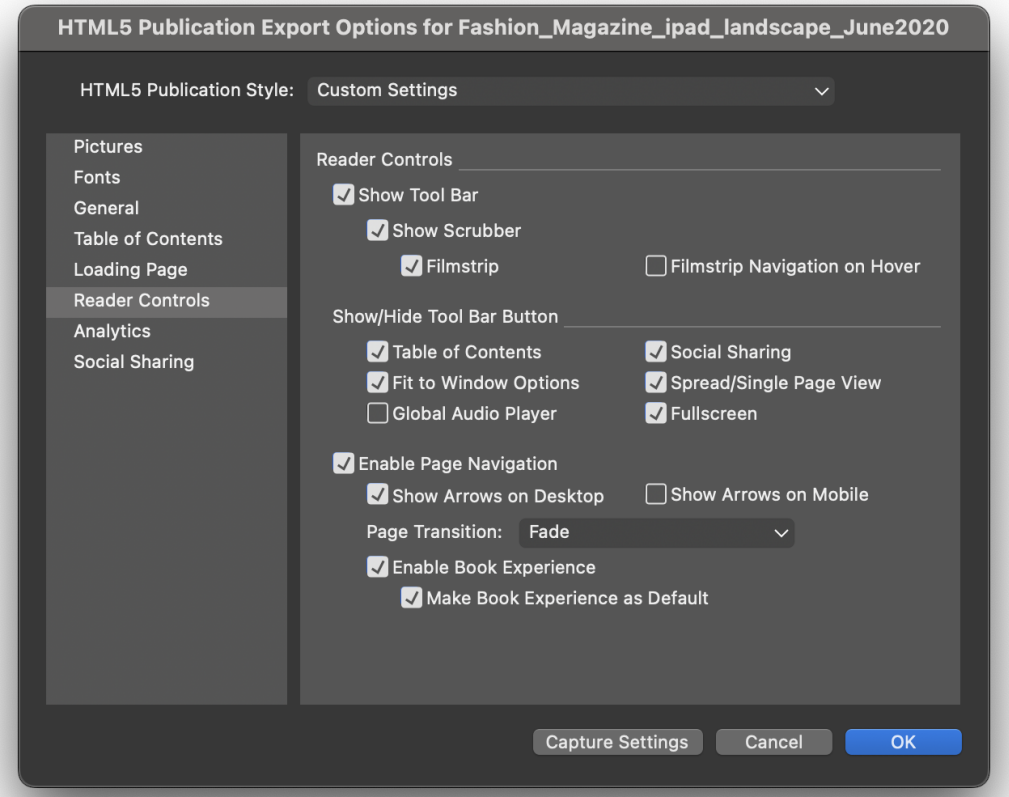
5 When you're finished configuring options, click **OK**. To capture the current settings without creating an HTML5 Publication file, click **Capture Settings**.

6 Click **Export**.

Reader Controls

The “Book Experience” feature has been introduced for HTML5 Publications. Users can enable the “Enable Book Experience” option in “Reader Controls” when exporting HTML5 Publications to achieve a realistic book flipping experience when viewing their publication in web browsers.

The “Enable Book Experience” selection is enabled by default to activate the book icon. However, in case user wants to experience the page flip view by default, the “Make Book Experience as Default” needs to be selected.



Reader Controls

Web app configuration

Configure your web app using the “reader-config.json”, the “ui-config.css” and the “manifest.json” files located at the root of your output.

Use these file to configure the following:

- Look and feel
- Features
- Analytics
- Social sharing

reader-config.json

General app configuration

Use the app section to configure some general application attributes:

- To specify the settings for the loading screen (the loading image, the font color and the background color), define these in the <loading> section, using the color, backgroundColor, and image parameters.
- You can choose the orientation of display pages in the respective devices using the mobile and desktop parameters in the <orientation> section.

Valid values for orientation > mobile and orientation > desktop are both, portrait, and landscape. For example, to display only landscape pages on desktop, lock the orientation of desktop to landscape. The default value is "both".

```
"app": {
  "title": "Quark App Studio Web App",
  "summary": "Pub description/summary goes here",
  "urlRoot": "./",
  "configFilename": "application-config.webreader.json",
 "orientation": {
      "mobile": "both",
      "desktop": "both"    },
  "privacy": {
      "enableCookieBanner": false,
      "cookieBannerMessage": "",
      "cookieAcceptButtonText": "",
       "cookieSource" : "",
       "cookieUrlButtonText" : ""    },
  "loading": {
       "image": "images/loading-logo.png",
       "backgroundColor": "#fff",
        "color": "#333333"    },
  "params": {
       "loglevel": -1    },
   "enableCookies": true,
  "enableExternal": true,
  "mobileOnly": false  },
"googlemaps": {
  "key": ""  },
```

- Use the key parameter under the googlemaps section to enter a Google Maps browser API key to use Google Maps (geolocation widget) in your webapp as keyless access to Maps is restricted by Google, (auto > googlemaps > key in the extensions). For more details see, "*http://developers.google.com/maps/documentation/javascript/get-api-key*".

Features configuration

Use the features section to enable/disable features in your Web App, and configure these features for your app.

➡ These parameters take values of true/false. If the value of any feature is “false", any other configurations you have for that feature will be ignored. (e.g If toc is “false", the TOC button will not be visible even if you have configured it in the toolbar section.

```
"features" : {
  "toc" : true,
  "banner" : true,
  "toolbar" : true,
  "pageFit" : true,
  "sharing" : true,
  "scrubber" : false,
  "analytics" : false,
  "fullScreen" : true
  "SpreadView" : true,
  "smartBanner" : true,
  "authorization" : true,
  "pageNavigation" : true,
  "variantSwitching" : true,
  "globalAudioPlayer" : true,
  "storeAccessibility" : true,
}
```

- Use the toolbar parameter to show/hide the toolbar in the webapp. If the toolbar display is true, the other toolbar button customizations will come into play.

```
"toolbar": {
  "visibility": "auto" | "initially" | "always",
  "items": [
    "storeButton",
    "tocButton",
    "scrubberBar",
    "pageFitButton",
    "spreadViewButton",
    "sharingButton",
    "globalAudioButton",
    "fullScreenButton"
  ]
},
```

  - Use the visibility parameter to configure visibility of the toolbar. The allowed values for the visibility parameter are: **auto**, **initially**, and **always**.
  - Use the items parameter to change the order of position of the buttons on the toolbar. The item will appear in the order they are listed in the array.

- Use the toc parameter to show/hide the TOC button in the toolbar. Turn it on if the publication uses section pages as table of contents. Further customize it using the toc section.
    - Display the TOC title as a banner using the banner parameter.
    - Visibility of the toc can be customised to show the toc when the web app is loaded using the showInitially parameter.

```
"toc": {
  "banner": true,
  "showInitially": false,
  "tocNonHtml": {
    "titleLines": 4,
    "showImage": false,
    "showHTMLTitle": false,
    "showDescription": false,
    "descriptionLines": 2
  }
},
```

- Use the scrubber parameter to show/hide the scrubber in the toolbar.
    - When the scrubber bar is enabled, use the filmStrip parameter under the scrubber section, to show/hide the filmstrip display on the page thumbnails in the issue.
    - Visibility of the filmstrip can be customised to show the filmstrip when the web app is loaded using the showInitially parameter under the filmstrip section.
    - Further enable/disable thumbnail navigation on the desktop using a mouse hover using the mouseHoverNavigation parameter under the filmstrip section.

```
"scrubber": {
  "filmStrip" : true
},
"filmStrip": {
  "showInitially" : true,
  "mouseHoverNavigation" : false
},
```

- Use the sharing parameter to show/hide the **Sharing** button in the toolbar.
- Use the globalAudioPlayer parameter to show/hide button to view the player for global audio.
- Use the pageNavigation parameter to stop the user from navigating to other pages, except via interactivities, the toc, or the scrubber bar. When pageNavigation is enabled, the user can configure the display of navigation arrows as mentioned in *Navigation configuration*.

- Use the fullScreen parameter to show/hide the button to enter fullscreen mode on the desktop.
- Use the pageFit to show/hide the button that can toggle between the page fitting options (ie **Fit Page to Window** (autofit mode) and **Fit Width**).
- Use the spreadView to show/hide the toggle button which can help switch between **Single Page** display or **Facing Pages** display for a publication with facing pages.

➡ If the publication has only single pages, then this button is not available by default (irrespective of the setting).

Social sharing configuration

Use the socialSharing and features sections to enable and configure the social media sharing options for your app.

- Use the sharing parameter of the features section to enable/disable the social sharing button from the toolbar.
- Use the socialSharing section to configure the social media sharing options for your app.
- Use the parameters of the options section (i.e. twitterShare, facebookShare, emailShare, linkedinShare) to enable/disable the available sharing options.

➡ For Facebook sharing, you can generate your own Facebook AppID and update it in the AppID property. For more information see "*http://developers.facebook.com/docs/apps/register*".

```
"features" : {
  ...
  "sharing" : true
  ...
}
"socialSharing": {
  "facebook": {
    "AppID": "125714554250404",
    "caption": "locale:facebook-sharing",
    "description": ""
  },
  "options": {
    "facebookShare" : true,
    "twitterShare" : true,
    "emailShare" : true,
    "linkedinShare" : true
  }
}
```

➡ The caption and description parameters for Facebook, under the Sharing section have been deprecated. (*https://developers.facebook.com/docs/sharing/reference/feed-dialog*)

Most content is shared to Facebook as a URL, so it's important to mark up your website with Open Graph tags to take control over how content appears on Facebook. For more information see "*https://developers.facebook.com/docs/sharing/webmasters*".

Configure the title of the post, images, and descriptions that are to be shown in a Facebook post through "index.html". By default this configuration is commented out, you can uncomment these lines and add content as per your requirements.

```
<meta property="og:title" content=""/>
<meta property="og:description" content=""/>
<meta property="og:image" content=""/>
```

- **title**: The title of your article without any branding such as your site name. Type the title in the 'content' field.
- **description**: A brief description of the content, usually between 2 and 4 sentences. This will be displayed below the title of the post on Facebook. Type the description in the 'content' field.
- **image**: The URL of the image that appears when someone shares the content to Facebook. Type the URL of the image in the 'content' field. Use images that are at least 1200 x 630 pixels for the best display on high resolution devices.

➡ At a minimum, you should use images that are 600 x 315 pixels to display link page posts with larger images. Images can be up to 8MB in size. See the following link for best practices: *//developers.facebook.com/docs/sharing/best-practices#images*

Analytics configuration

Use the analytics section to configure Google analytics tracking for your app.

- To specify a Google tracking ID for your app, use the trackingId parameter of the <GoogleAnalytics> section, to update with your own trackingID.

➡ For more information see "*http://www.google.com/analytics/*".

```
"analytics": {
  "GoogleAnalytics": {
    "trackingID": "UA-20677225-13",
    "enhancedLinkAttribution": false
    "anonymizeIp": false
  },
},
```

Navigation configuration

Use the pageNavigation section to configure the page navigation for your app.

- Use the showArrowsForMobile parameter to enable/disable the display of page navigation arrows on the mobile version of your webapp. The user can navigate pages using the arrows in addition to navigation using the scrubber bar, filmstrip, toc or interactivity.
- Use the showArrowsForDesktop parameter to enable/disable the display of page navigation arrows on the desktop version of your webapp. The user can navigate pages using the arrows in addition to navigation using the scrubber bar, filmstrip, toc or interactivity.
- Use the transition parameter to define page transtion effects (Fade, Slide and None).

```
"pageNavigation": {
  "showArrowsForMobile": false,
  "showArrowsForDesktop": true,
  "transition": slide
},
```

View configuration

Use the pageFitView and the spreadView sections to configure the view for your app.

pageFitView

- Use the defaultMode parameter to set the default fitting mode of the publication page in your webapp. Pages can be scaled up or down appropriately to fit the device size / browser size. Publication pages viewed in a webapp can be displayed as:
  - autofit - the pages will be scaled proportionally in both dimensions to display the complete page to fit in the device browser.
  - widthfit - size the pages to fill the device browser width and enable scrolling.
- Use the maxScaleForAutoFit parameter to specify the max limit for scaling a page in autofit mode. To limit scaling beyond 100% of the original dimensions, specify a value of 1. For 200% specify a value of 2. To have no limit restriction and to be able to scale pages as much as the browser size specify a value of 0.

```
"pageFitView": {
  "defaultMode": "autofit",
  "maxScaleForAutoFit": 4
},
```

spreadView

if facing pages (portrait or landscape) are available in the publication they will be displayed in landscape orientation in a device, or landscape like dimensions in the desktop browser. By default, the facing pages display is on, and a user can toggle the display to single page and vice versa.

- Use the mobile parameter under the showLandscapeFacing parameter to enable/disable the display of landscape facing pages on the device.

```
"spreadView": {
 "showLandscapeFacing": {
 "mobile": false,
 "desktop": true
 }
}
```

Privacy settings configuration

Use the app section to configure the privacy settings for your app:

```
"app": {
        ...
        "privacy": {
        "enableCookieBanner": false,
        "cookieBannerMessage": "",
        "cookieAcceptButtonText": "",
        "cookieSource" : "",
        "cookieUrlButtonText" : ""
        },
      },
```

- enableCookieBanner: When true, presents the cookie disclaimer at the bottom of the page. By default, the value is set as false.
- cookieBannerMessage: Text to be displayed in the cookie disclaimer. HTML tags can also be included in this text to create a hyperlink to the privacy policy page. (For example: "*This website uses cookies to improve your experience on our website. By continuing to browse this site, you are agreeing to our use of cookies. You can disable cookies in your browser settings at any time. More information in our Privacy policy.*".)
- cookieAcceptButtonText: The label of the button which dismisses the cookie disclaimer banner. (For Example: *"I Understand"*.)
- cookieSource: Path of the HTML (relative /URL) containing the Cookie Policy/Privacy Policy.
- cookieUrlButtonText: The label of the button to open the HTML page given in cookieSource. (For example: *"Learn More"*.)

ui-config.css

```
:root {
--app-background: #333;
--toolbarItem-background-color: #000;
--toolbarItem-background-color-on-hover: #00aeef;
--toolbar-svg-color: #a3a3a3;
--toolbar-svg-hover-color: #fff;
--scrubber-thumb-color: #a3a3a3;
--scrubber-thumb-hover-color: #fff;
--timeline-background-color: rgba(66, 65, 66, 0.88);
--timeline-thumbnail-border-color: #00aeef;
--toc-banner-color: #000;
}
```

- Use the app-background parameter to set the background color of the application.
- Use the toolbarItem-background-color parameter to set the background color of toolbar item.
- Use the toolbarItem-background-color-on-hover parameter to set the background color of the toolbar item on hover.
- Use the toolbar-svg-color parameter to set the svg color of the toolbar item.
- Use the toolbar-svg-hover-color parameter to set the svg color of the toolbar item on hover.
- Use the scrubber-thumb-color parameter to set the color of the dragging handle of the scrubber bar.
- Use the scrubber-thumb-hover-color parameter to set the color of the dragging handle of the scrubber bar on hover.
- Use the timeline-background-color parameter to set the background color of the filmstrip which shows thumbnails.
- Use the timeline-thumbnail-border-color parameter to set the border color of thumbnail of the current page in the filmstrip.
- Use the toc-banner-color parameter to set the background color of the toc banner.

manifest.json

➡ The URL of the web app must be 'https' for this feature to be available.

```
{
 "name": "App Studio",
 "short_name": "App Studio",
 "start_url": "index.html",
```

```
"display": "standalone",
"theme_color": "#3f51b5",
"background_color": "#000",
"icons": [{
   "src": "images/manifest/apple-touch-icon.png",
   "type": "image/png"
 },
 {
   "src": "images/manifest/apple-touch-icon-76x76.png",
   "type": "image/png",
   "sizes": "76x76"
 },...
```

- Use the name parameter to set the app name that appears on the home and splash screens.
- Use the display parameter to set the display behavior of the home screen. It has two values:
  - standalone: The home screen shortcut will open the app in full screen mode.
  - browser: The home screen shortcut will open the app in a browser in a new tab.
- Use the theme_color parameter to set the color of the status bar. The meta tag for theme_color is defined in the "index.html" file which overrides the theme_color defined in the manifest file.
- Use the background_color parameter to set the background color of the splash screen of the home screen app.
- Use the icons parameter to define the home screen app icons. Specify the icons of different sizes for different device sizes.

Cookie consent and privacy policy configuration

➡ Any text for cookie consent we have provided as an example in the configurations above is sample text only and should be reviewed and approved by your own legal team.

Your privacy policy should disclose the fact that your web app uses cookies, and supply them with a list of cookies that are used/stored.

We recommend that your privacy policy include something similar to the following:

What are cookies and how do they work?

Cookies are pieces of information stored on your hard drive containing information about you. These pieces of information allow the Service to

remember important information that will make your use of the Service more useful. Web App uses cookies for a variety of purposes. For instance, we use cookies to allow your automated login and to track your preferences. You can choose to turn off the cookies through your browser settings. If you turn off the cookies, you may still use the Service, but we will not be able to save your preferences.

What cookies does the Web App use?

The following is a list of the technical that are used/stored in the Web App:

Cookie Name	Cookie Purpose
ASconsent	Used to set the state of cookie consent banner, if user accepted once then next time banner will not be displayed.

For more information please see our entire privacy policy here....

➡ Again, any content that we suggest (especially descriptions) should be reviewed and approved by your legal team.

Web app capabilities

Autofit and Autoscale HTML5 Publication

Regardless of the layout size, users will be able to view the output on any device without truncation. The layout will be auto fit according to device size or desktop browser size, and scaled proportionally to completely fit to the device screen size. For example, content created for iPad can also be viewed on smaller devices like iPhones and vice versa.

Users can additionally switch to Fit to Window width and enable scrolling.

Zoom in/out page on mobile devices

Users will be able to zoom in/zoom out content on devices via pinch in and pinch out gestures. Users will also be able to perform content panning in zoom mode on mobile devices for an enhanced reading experience. When in zoom mode, page swipe will be restricted and panning action will take precedence. Users can perform reset zoom using a double tap.

Web app performance

Web app performance can be increased if the content uploaded on the server is compressed. Another way to improve performance is to utilize file compression.

For example, for the apache server, the ".htaccess" file in the root folder compresses the files and therefore enhances the performance. Users can contact the hosting service for this feature.

➡ This configuration is not the part of the standard build as you can have different web servers and the configuration would vary for every server.

Web app available for offline reading

The Web app will be available for offline reading once viewed. All text content in the issues will be available. Images and other assets on the current and next/previous page will also be available. Web app will be available in offline mode on browsers which support 'Service worker' as offline support is based on the cached mechanism available through service worker.

➡ The URL of the web app must be 'https' for this feature to be available.

Hosting an HTML5 package on a web server

To view your HTML5 publications in a web browser, (mobile devices, Windows or Mac) deploy a web server where you can host the exported package, exactly like you would host any website.

To deploy a web app:

1. Setup the web server.
2. Host the HTML5 publication package.
3. View the output.

To customize the look and feel, branding, and features that will be available in the hosted web app, see "*HTML5 Publication output styles*."

An example for deploying on macOS

Enable Apache web sharing on Mac OS X

1. Open the terminal, found at /Applications/Utilities/Terminal.
2. Run the command "sudo apachectl start".
3. Supply the root password.

Test to see if the web server is running

1. Open the browser and navigate to **http://localhost**.
2. You should see "It Works!".

Web share the digital output

1. Create a folder (i.e. 'Digital') and copy the digital output folder to /Library/WebServer/Documents/Digital.
2. Open the folder properties and unlock security.
3. Give **Read & Write** permissions to required users and **Read only** permissions to everyone, and apply to enclosed items.

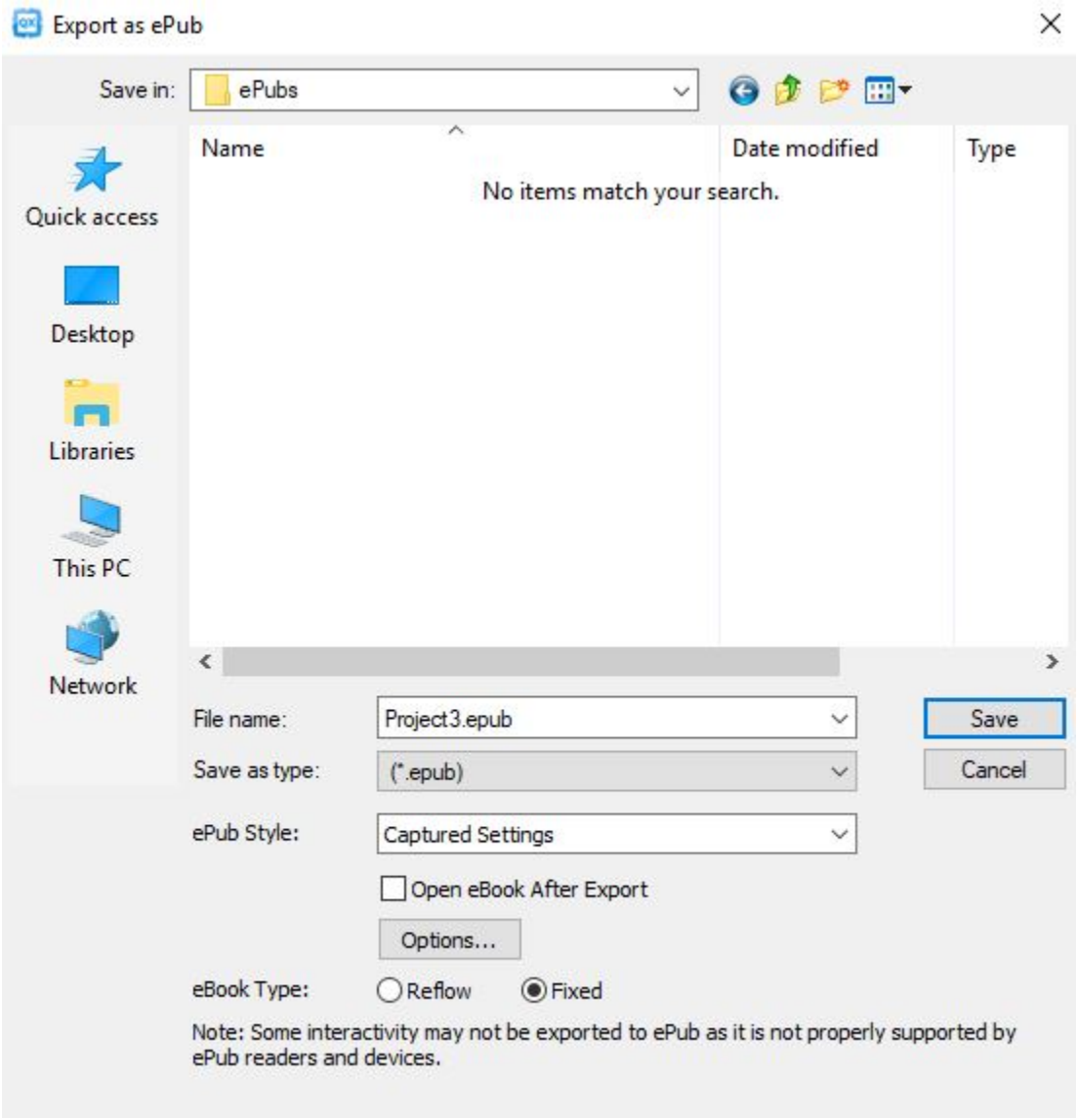
View the digital output

1. Open browser and navigate to http://localhost/Digital.
2. You will be redirected to the first page in the issue.

Exporting for ePub

To export a layout in ePub format:

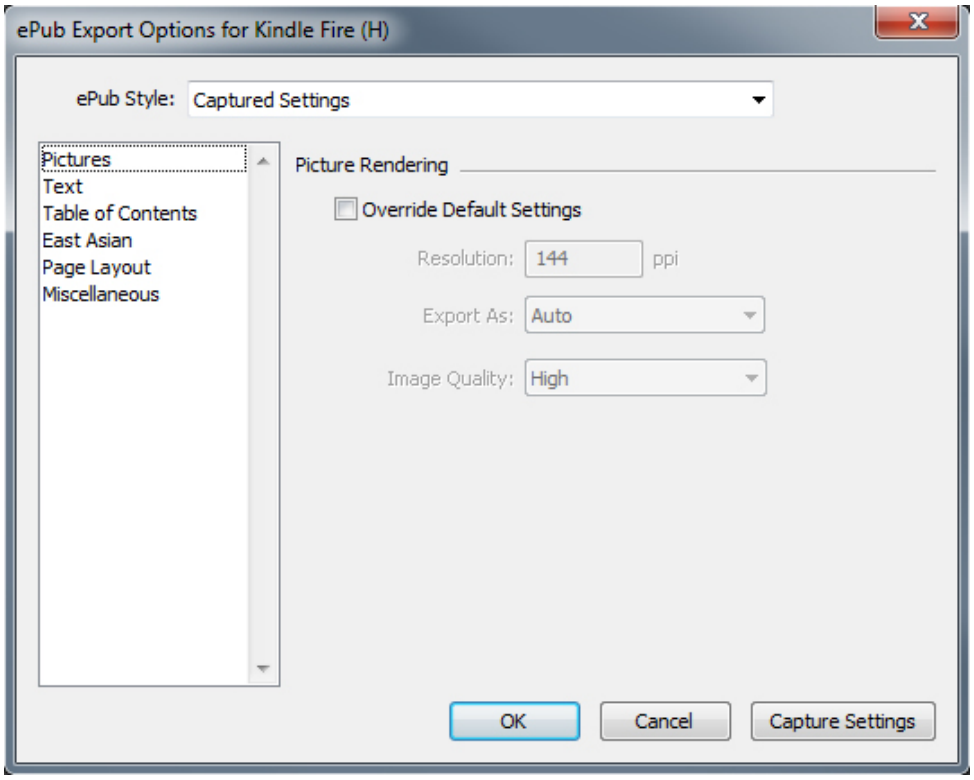
1 Choose **File > Export as > ePub**. The **Export as ePub** dialog displays.



Export as ePub dialog

2 Enter a name for the exported eBook. QuarkXPress will append the correct file extension.

3 Choose an output style from the **ePub Style** drop-down menu or click **Options**. If you click **Options**, the **ePub Export Options** dialog displays.



ePub Export Options dialog

- Use the **Pictures** pane to specify the resolution, format, and quality of pictures in the ePub file. To override any custom attributes applied to individual picture components, check **Override Box Settings**.
- Use the **Text** pane to specify how to export Drop Cap in Reflow eBooks: as **Native Text** or **Ignore**.
- Use the **Table of Contents** pane to set options for the table of contents in the ePub file. You can choose whether to use the Reflow article names defined in the **Articles** palette or you can use a predefined list as the Table of Contents. (For more information, see "*Creating a TOC for ePub*.") You can also specify the name of the TOC entry for the cover and the title displayed above the TOC in the HTML version of the table of contents.
- Use the **East Asian** pane to specify a default story direction and choose whether to export Rubi natively or in parentheses.
- Use the **Page Layout** pane to choose how Footnotes are exported in Reflow eBooks, and lets you specify orientation and spread options for Fixed Layout eBooks.
- Use the **Miscellaneous** pane to choose a target platform.

4 When you're finished configuring options, click **OK**. To capture the current settings without creating an ePub file, click **Capture Settings**.

➡ Output styles can be managed by selecting **Edit > Output Styles**. See "*A Guide to QuarkXPress*".

5 Choose **Reflow** or **Fixed** from the **eBook Type** section.

6 Click **Save**.

In reflowable ePub, bullets, numbering and outline numbers, including multi-level outline styles, are exported as text, so they appear correctly in all ePub readers.

➡ In QuarkXPress, users can export an ePub for Kindle-based publications with or without an optional cover page.

If a quark – in science – is the basis for all matter, then Quark – the company – is the basis for all content. The company revolutionized desktop publishing and today provides content design, automation and intelligence software for end-to-end content lifecycle management. Customers worldwide rely on Quark to modernize their content ecosystems so they can create complex print and digital layouts, automate omnichannel publishing of mission-critical documents, and analyze production and engagement insights for the greatest return on their content investments.

© 2026 Quark Software, LLC. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective owners. This content is informational and does not imply any guarantee or warranty.