



A Guide to QuarkXPress Server 9.2

Contents

About this Guide.....	5
What we're assuming about you.....	5
Where to go for help.....	5
Conventions.....	6
 Getting started with QuarkXPress Server.....	 7
QuarkXPress Server architecture.....	7
Understanding QuarkXPress Server.....	8
Server templates and static projects.....	8
The document pool and the Streaming Document Provider.....	9
Projects and layouts.....	9
Job Jackets and resources.....	9
Caching.....	10
QuarkXPress Server preferences.....	10
Quark License Administrator.....	11
Master-renderer environment.....	11
Changing logging levels in "log4j.xml".....	11
Understanding QuarkXPress Server XTensions software.....	12
The QuarkXPress Server XTensions API.....	12
Configuring QuarkXPress Server for launch.....	13
Launching QuarkXPress Server.....	14
Quitting QuarkXPress Server.....	14
 The QuarkXPress Server user interface.....	 15
Navigation pane.....	15
Administration menu.....	17
General Preferences dialog box.....	17
Renderer Preferences dialog box.....	20
Job Jackets dialog box.....	28
Check Out License dialog box.....	29
 Creating URL requests.....	 30
Understanding URL requests.....	30
Understanding QuarkXPress Server namespaces.....	31
Looking up a namespace.....	31

Understanding QuarkXPress Server parameters.....	32
Looking up a parameter.....	32
QuarkXPress Server Web Integration Guide overview.....	32
Render types.....	32
General parameters.....	32
Specialized parameters.....	33
Server control parameters.....	33
Document providers.....	33
 Modifying layouts with XML.....	 34
 Using SSL.....	 35
Secure Sockets Layer (SSL) support.....	35
Enabling SSL.....	35
Enabling HTTP and HTTPS.....	36
Verifying and using SSL.....	36
Keystores and SSL certificates.....	36
 QuarkXPress Server XTensions software.....	 38
CopyDeskArticle XTensions software.....	38
Rendering articles.....	38
Exporting articles.....	38
Adding articles to projects.....	39
Creating and deleting components.....	39
PDF Filter XTensions software.....	40
Modifier XTensions software.....	40
Using Modifier XTensions software.....	41
Creating XML for Modifier XTensions software.....	42
Layer XTensions software.....	42
InteractiveDesigner Server XTensions software.....	42
App Studio XTensions software.....	43
 Telegraph XTensions software.....	 44
Setting Telegraph preferences.....	44
Specifying a server for template upload.....	45
Using Telegraph XTensions software.....	47
Identifying QuarkXPress items and groups.....	47
Naming items and groups.....	47
Uploading templates.....	48
Uploading missing or modified pictures.....	50

QuarkXPress Server Manager.....	51
Understanding QuarkXPress Server Manager.....	51
Load balancing.....	51
Request timeout interval.....	52
Determining QuarkXPress Server instance availability.....	53
Logging.....	53
Caching.....	54
Web services.....	54
Working with QuarkXPress Server Manager.....	54
Starting QuarkXPress Server Manager.....	55
Request handler binding.....	56
Configuring QuarkXPress Server instances.....	56
Managing the cache.....	60
Managing logs.....	62
Using a proxy server.....	65
Generating automatic e-mail messages.....	65
Using other global settings.....	66
Saving a server configuration.....	68
Using custom error messages.....	68
Sending requests from a browser.....	70
Glossary.....	72
Legal notices.....	73

About this Guide

You do not need to read this guide from cover to cover. Instead, use this guide to quickly look up information, find out what you need to know, and proceed with your work. Use the detailed table of contents at the beginning of the guide to find information quickly.

What we're assuming about you

In writing this guide, we assume that you are familiar with your computer and know how to:

- Launch an application
- Open, save, and close files
- Use menus, dialog boxes, and palettes
- Work within a networked computing environment
- Use the mouse, keyboard commands, and modifier keys

Consult the documentation provided with your computer or other resources if you need more information in any of these areas.

Where to go for help


If you're new to QuarkXPress® Server, or if you want to explore one of its other longstanding features, consult the following resources:

- *A Guide to QuarkXPress Server*
- The QuarkXPress Server *Web Integration Guide*
- Third-party books
- General books about desktop publishing

If your issues are at the system level — saving files, moving files, activating fonts, for example — consult the documentation resources provided with your computer.

Conventions

Formatting conventions highlight information to help you quickly find what you need.

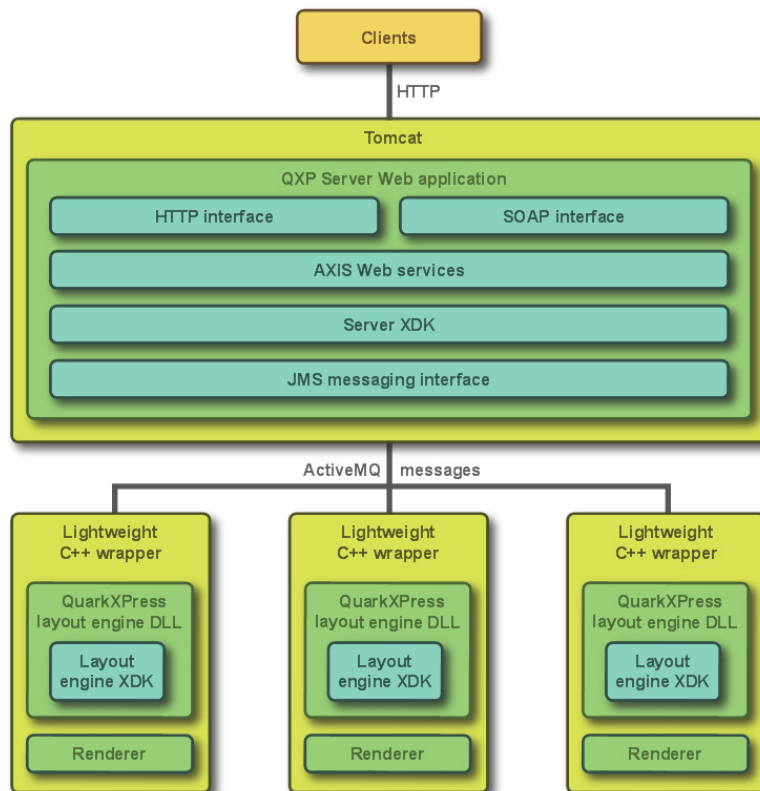
- **Bold type style:** The names of all dialog boxes, fields, and other controls are set in bold type. For example: "Click **OK**."
- **References:** In descriptions of features, parenthetical references guide you in accessing those features. For example: "The **Find/Change** dialog box (**Edit** menu) lets you find and replace text."
- **Arrows:** You will often see arrows (>), which map out the menu path to a feature. For example: "Choose **Edit > Style Sheets** to display the **Style Sheets** dialog box."
- **Icons:** Although many tools and buttons are referenced by name, which you can see by displaying ToolTips, in some cases icons are shown for easy identification. For example, "Click the  button on the **Measurements** palette to center text."
- **Cross-platform issues:** This application is quite consistent across operating systems. However, some labels, buttons, key combinations, and other aspects of the application must differ between Mac OS® and Windows® because of user interface conventions or other factors. In such cases, both the Mac OS and Windows versions are presented, separated by a slash, with the Mac OS version presented first. For example, if the Mac OS version of a button is labeled **Select**, and the Windows version is labeled **Browse**, you are directed to "Click **Select/Browse**." More complex cross-platform differences are mentioned in notes or parenthetical statements.

Getting started with QuarkXPress Server

QuarkXPress Server lets you render QuarkXPress projects in a variety of formats.

QuarkXPress Server architecture

The QuarkXPress Server architecture is shown in the following diagram.



QuarkXPress Server architecture diagram

Understanding QuarkXPress Server

QuarkXPress Server lets you output customized QuarkXPress layouts in a variety of formats — including JPEG, Portable Document Format (PDF), and PostScript® — from a centralized QuarkXPress Server application. To send a request to a QuarkXPress Server application, all you need to do is enter a URL into your Web browser's address field. For example, the following URL instructs the QuarkXPress Server application named "QXPServer" to return the file "MyProject.qxp" as a PDF file:

```
http://QXPServer:8080/pdf/MyProject.qxp
```

The QuarkXPress Server application receives these requests, renders (creates) the requested projects in the requested formats, and then returns the rendered file to the client application (in this case, the Web browser).

- ➔ The format of QuarkXPress Server URL requests is described in detail in "[Creating URL requests](#)" and in the QuarkXPress Server *Web Integration Guide*.
- ➔ You can also create custom applications that communicate with a QuarkXPress Server application using HTTP, Simple Object Access Protocol (SOAP), or another protocol. For more information, see the QuarkXPress Server *Web Integration Guide*.

You can think of QuarkXPress Server as a special version of QuarkXPress that runs on a server with the following main differences:

- Instead of accepting input from a keyboard and mouse, QuarkXPress Server accepts input in the form of URLs and other types of requests.
- Instead of sending a project to a printer, QuarkXPress Server renders (creates) output in a particular format and sends the rendered file to a client.

Server templates and static projects

QuarkXPress Server can open, render, and serve two types of projects:

- *Static projects* are QuarkXPress projects that can be rendered and served as-is in a variety of formats by the QuarkXPress Server application. For example, you might make a product manual available as a static project and allow customers to download that manual in a variety of formats.
 - *Server templates* are QuarkXPress projects that can be manipulated by the QuarkXPress Server application before being rendered and served. For example, you might make a sales piece available as a server template so that each person who downloads it receives a personalized copy.
- ➔ QuarkXPress Server can open QuarkXPress documents, projects, and templates created in any language edition of QuarkXPress 5.0 or later. QuarkXPress Server can save and export projects in QuarkXPress 8.x and QuarkXPress 9.x format.

The document pool and the Streaming Document Provider

QuarkXPress Server has two main ways to find or receive content for rendering.

- The document pool
- The Streaming Document Provider

QuarkXPress Server can read templates and static projects from a directory called the *document pool*. The document pool can be any directory that is available to QuarkXPress Server through a file system or an FTP server. You can use any of the following methods to place files in the document pool:

- Drag the files to the document pool directory.
- Use the **Add Files** command in the **Document Pool** screen of the QuarkXPress Server Web interface.
- Upload the files using FTP to the document pool directory.
- Use Telegraph XTensions® software to upload the files to the document pool from within QuarkXPress. (For more information about Telegraph XTensions software, see "[Telegraph XTensions software](#).")

For more information about the document pool, see "[The QuarkXPress Server user interface](#)."

QuarkXPress Server can also receive templates, projects, and other files as part of a multipart HTTP request. For more information, see "Using the Streaming Document Provider" in the *QuarkXPress Server Web Integration Guide*.

In addition, QuarkXPress Server can serve documents from a database, a content management system, or other sources. Collectively, the document pool and any other source of files to be served are referred to as *document providers*.

Projects and layouts

QuarkXPress projects can contain one or more layouts, and only one layout can be rendered at any given time. If you do not specify a layout when you send a rendering request, QuarkXPress Server renders the first layout in the project.

Job Jackets and resources

In QuarkXPress, *resources* are things such as style sheets, colors, H&Js, output styles, and item styles. Resources are stored in a *Job Jackets structure*, which can be either embedded in a project or stored in a separate Job Jackets file.

QuarkXPress Server uses a default Job Jackets file to make a default set of resources available to all projects handled by QuarkXPress Server, regardless of whether they are included in the projects and articles you render. You can update this file in two ways:

- Using the QuarkXPress Server Web interface. For more information, see "[Job Jackets dialog box](#)."

- Using request handlers. For more information, see [jobjacket](#) and [updateprefsfromjj](#) the *QuarkXPress Server Web Integration Guide*.

Once you've downloaded the default Job Jackets file, you can update its resources using QuarkXPress. For more information, see "Job Jackets" in *A Guide to QuarkXPress*.

The location of the default Job Jackets file is stored in the QuarkXPress Server preferences folder. For more information, see "[QuarkXPress Server preferences](#)."

Caching

To maximize efficiency, QuarkXPress Server uses cached versions of all rendered projects whenever possible. You can configure projects so that they are never cached, so that they are cached for a particular amount of time, or so that they are rendered every time they are served.

QuarkXPress Server preferences

When you launch QuarkXPress Server, the application creates preferences files that are functionally and structurally equivalent to the preferences files created by QuarkXPress. These preferences files reside in the QuarkXPress Server "Preferences" folder. QuarkXPress Server also creates a "QuarkXPress Server.prf" file in the "Preferences" folder. This file contains preference settings that are specific to QuarkXPress Server.

QuarkXPress Server uses these preferences the same way QuarkXPress uses them. If an XTensions module creates a project in QuarkXPress Server, that project draws its settings from the QuarkXPress Server preferences just as a new QuarkXPress project draws its settings from the QuarkXPress preferences.

Preferences files are stored in the following locations:

- Mac OS: `[User]/Library/Preferences/Quark/QuarkXPress Server Renderer 9.1/`
- Windows 2003 and Windows XP when running normally or under a domain user account: `C:\Documents and Settings\[user name]\Application Data\Quark\QuarkXPress Server Renderer 9.1`
- Windows 2003 and Windows XP when running as a service under a local user account: `C:\Documents and Settings\Default User\Application Data\Quark\QuarkXPress Server Renderer 9.1\`
- Windows Vista, Windows 2008, or Windows 2008 R2 64-bit when QuarkXPress Server is running normally or as a service under a domain user account: `C:\Users\[user name]\AppData\Roaming\Quark\QuarkXPress Server Renderer 9.1\`
- Windows 2008 or Windows Vista, when QuarkXPress Server is running as a service under a local user account: `C:\Windows\System32\config\systemprofile\AppData\Roaming\Quark\QuarkXPress Server Renderer 9.1\`

- Windows 2008 R2 64-bit, when QuarkXPress Server is running as a service under a local user account:

```
C:\Windows\SysWOW64\config\systemprofile\AppData\Roaming\Quark\QuarkXPress  
Server Renderer 9.1
```

For more information about preferences, see "[Administration menu](#)."

Quark License Administrator

To prevent unauthorized use, QuarkXPress Server requires the presence of a Quark® License Administrator (QLA) server to launch. QuarkXPress Server follows the configuration and control rules that are enforced by QLA. For more information about QLA, see the QLA documentation included with QuarkXPress Server.

Master-renderer environment

Requests for project renders are stored in a *connection queue*. The requests in the rendering queue can be processed by a single QuarkXPress Server application, or by a master QuarkXPress Server application and several renderers (additional instances of QuarkXPress Server). The master QuarkXPress Server application launches the available number of renderers and then passes the requests in the connection queue to those renderers as they become available. The number of renderers available for launch is determined by the number of licenses available from the QLA server.

The master QuarkXPress Server process and all of the renderers it launches share the following elements:

- The same application preferences (each renderer has its own preferences files, but QuarkXPress Server keeps them synchronized)
- The same document cache in memory
- The same memory cache
- The same server XTensions modules (a separate instance of each XTensions module runs with each renderer)
- The same server document pool (if defined in the QuarkXPress Server preferences or if a document provider is used in place of the document pool)

If a renderer unexpectedly quits, the master QuarkXPress Server restarts the renderer without requiring any action from you.

Changing logging levels in "log4j.xml"

You can change the logging levels for QuarkXPress Server. Options include **ERROR**, **INFO**, **WARN**, **DEBUG**, and **TRACE**.

- **ERROR** = includes messages that indicate disrupted and failed requests.
- **INFO** = includes messages that indicate the state of services.
- **WARN** = includes non-critical service error messages

GETTING STARTED WITH QUARKXPRESS SERVER

- **DEBUG** = includes messages that indicate server resource usage.
- **TRACE** = includes messages according to activity related to requests.

Refer to Java documentation for more information about logging levels.

To change logging levels:

- 1 Open the "conf" folder in your QuarkXPress Server folder.
- 2 Open "log4j.xml" in a text-editing application.
- 3 To define the logging level for QuarkXPress Server errors, scroll to `<logger name=com.quark.qxps`. The structure is as follows:

```
<logger name="com.quark.qxps">
  <level value="ERROR" />
</logger>
```
- 4 To define the logging level for QuarkXPress Server transactions, scroll to `<logger name=QXPSTransactionLogger`. The structure is as follows:

```
<logger name="com.quark.qxps" additivity="false">
  <level value="INFO" />
  <appender-ref ref="QxpsTransactionFileAppender" />
</logger>
```
- 5 To define the logging level for other activity, scroll to the `<root>`. The structure is as follows:

```
<root>
  <priority value="ERROR" />
  <appender-ref ref="QxpsServerAsyncAppender" />
</root>
```
- 6 Save and close "log4j.xml."

Understanding QuarkXPress Server XTensions software

QuarkXPress Server ships with a collection of XTensions software that adds capabilities to QuarkXPress Server. For example, PDF Export XTensions software lets QuarkXPress Server serve content in PDF format; Modifier XTensions software lets you retrieve, manipulate, and reconstruct XML representations of projects; and QuarkCopyDesk® Renderer XTensions software lets you create QuarkCopyDesk articles. Telegraph XTensions software works with QuarkXPress to allow designers to name boxes in template files so that those boxes can be addressed by URLs.

For more information about XTensions software included with QuarkXPress Server, see "[QuarkXPress Server XTensions software](#)" and "[Telegraph XTensions software](#)."

The QuarkXPress Server XTensions API

In addition to the XTensions modules included with QuarkXPress Server, developers can create custom XTensions software that add features. The complete server XTensions Application Programming Interface (API) documentation is available in the QuarkXPress Server XTensions Developer's Kit (XDK).

➔ As of version 8.0, the QuarkXPress Server XDK is Unicode-compliant.

The QuarkXPress Server XDK lets you create XTensions modules that provide the following abilities:

- The ability to register request handlers
- The ability to register project providers
- The ability to register new render formats
- The ability to add items to the list of response properties, cookies, and HTTP header items
- The ability to log messages in log files
- The ability to initiate a new transaction to be processed by the server
- The ability to completely control how projects are processed by the server

In addition, QuarkXPress Server XTensions software can register for the following basic callbacks:

- Pre-processing
- Content loading
- Layout modification
- Post-processing
- Removing slugs while running the QuarkXPress project renderer
- Analyzing the server after a transaction is complete
- Pre- and post-transaction callback

Configuring QuarkXPress Server for launch

To configure QuarkXPress Server prior to launch, open the file `[QuarkXPress Server application folder]/conf/ServerApp.properties` folder) and modify it as follows:

- To make the server run without loading any network interface, use the `-nonetwork` option with the `qxpservercore.serverRendererParameters`. In this mode, the only transactions a server can run are those passed to it by another process.
- To control whether renderers are monitored, set `qxpservercore.monitorrenderers.value` to true or false.
- To specify the query interval for monitoring renderers, set `qxpservercore.monitorrenderers.queryinterval.value` to a value in seconds.
- To specify the number of retries for monitoring renderers, set `qxpservercore.monitorrenderers.noofretries.value` to an integer. If a renderer process has been attempting to fulfill a request for the specified number of retries (with

the specified query interval in seconds between retries), the renderer monitor recycles that process.

- To control how many renderers the master process launches, specify a number for `qxpserver.subrenderers`. Note that the number of renderers you can launch depends on your license.

Launching QuarkXPress Server

On Windows, you can install QuarkXPress Server as an application or as a service (Quark recommends that you always run it as a service). On Mac OS, QuarkXPress Server always runs as an application.

If you install QuarkXPress Server on Windows as an application, you can launch it using the **Start** menu or by double-clicking the "ServerStartup.bat" in the QuarkXPress Server application folder.

- ➡ If you want to launch QuarkXPress Server as an application on Windows Server 2008, Windows Vista, and Windows 7, Quark recommends executing "ServerStartup.bat" as an administrator.

For information on launching QuarkXPress Server in a separate Tomcat installation, see "Deploying QuarkXPress Server externally" in the QuarkXPress Server *ReadMe*.

- ➡ QuarkXPress Server offers a browser-based user interface instead of a conventional user interface.

Quitting QuarkXPress Server

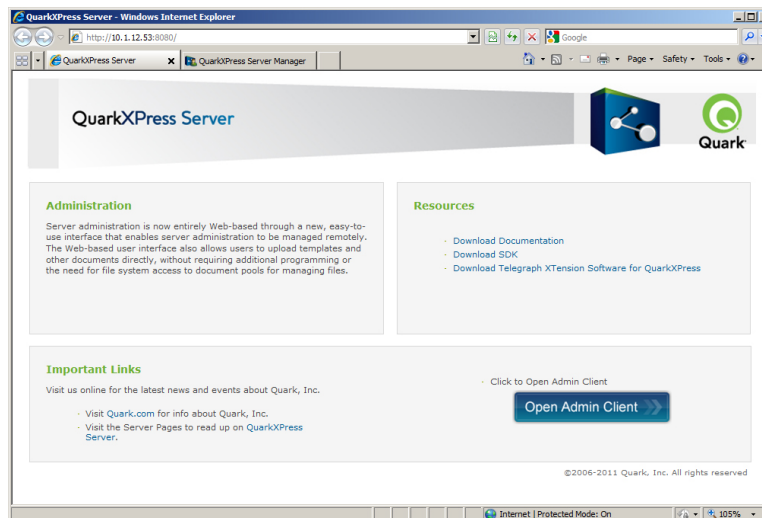
To quit QuarkXPress Server on Mac OS, press Control+C.

To quit QuarkXPress Server on Windows, press Control+C or double-click the "shutdown.bat" file in the QuarkXPress Server application folder.

The QuarkXPress Server user interface

QuarkXPress Server offers a browser-based user interface instead of a conventional user interface. This chapter describes that interface and explains how you can use it to configure and customize your QuarkXPress Server application and manage your server XTensions modules.

To view the QuarkXPress Server welcome page, launch a Web browser and enter the URL `http://[server]:[port]` (where `[server]` is the IP address or domain name of the server and `[port]` is the TCP/IP port on which the server is running). The welcome screen displays.



The QuarkXPress Server welcome page

To display the administrative client, click **Open Admin Client**. If the server has realm verification enabled, you will be asked to enter your user name and password. The administrative client displays.

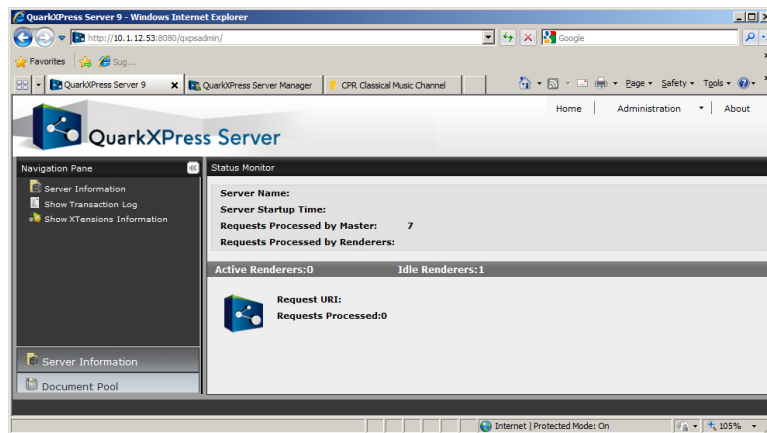
Navigation pane

The navigation pane on the left side of the has two areas. The **Server Information** area lets you view server information and the transaction log, and the **Document Pool** area

THE QUARKXPRESS SERVER USER INTERFACE

lets you view the contents of the document pool. You can collapse and expand this pane with the button at the right end of the **Navigation Pane** header.

If you click **Server Information**, the **Status Monitor** screen displays. The fields in the top area provide information about the server. The icons in the area below represent the renderers that are currently running and show which requests are being processed by each renderer in real time. This screen also shows how many active and idle renderers there are.

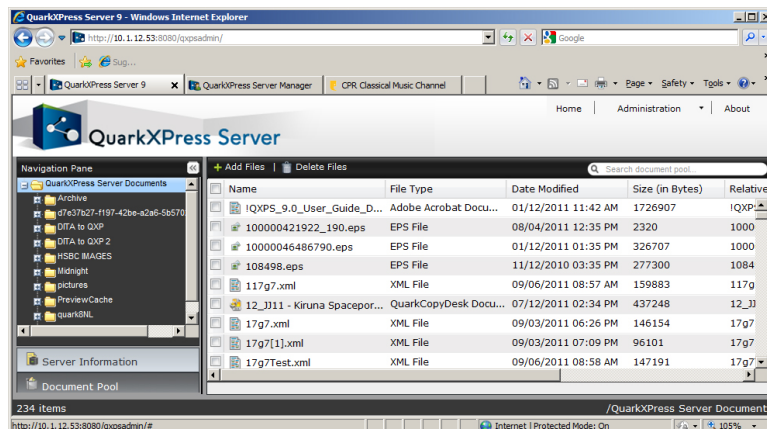


Status Monitor screen

If you click **Transaction Log**, the **Logs** screen displays the current transaction log. To download the active log, including recent crash reports, click **Download Logs** at the top of the **Logs** header.

If you click **Show XTensions Information**, the **XTensions Information** screen displays, allowing you to view the status of all installed XTensions modules.

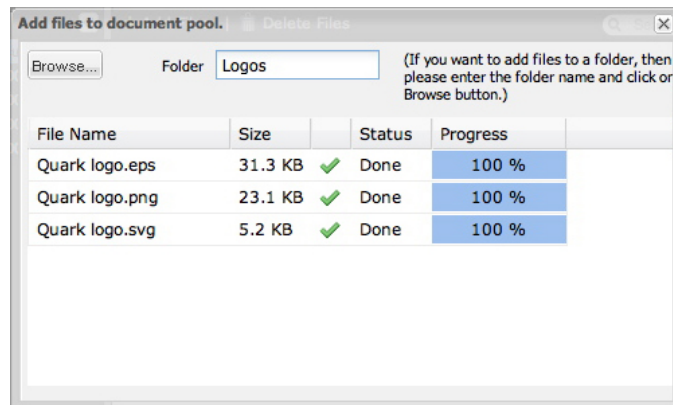
If you click **Document Pool**, the document pool screen displays.



Document Pool screen

You can Option+click/right-click the files in the document pool to access a variety of rendering options.

To add a file, click **Add Files** in the header. If you add multiple files at one time, you can create a folder to put them all into by entering the folder name in the **Folder** film.



Document Pool screen

To delete a file, check its box and then click **Delete Files**.

To search for a file or folder, use the search field in the header.

To preview a file, Control+click/right-click the file and choose **Preview**.

To get a file's relative path, double-click the file name.

Administration menu

The administration menu lets you manage QuarkXPress Server.

- ➔ You do not have to restart the server in GUI mode to set preferences.

General Preferences dialog box

The **General Preferences** dialog box (**Administration > Preferences > General**) lets you set preferences that are not related to rendering. It includes the tabs described in the following topics.

- ➔ You can also set general preferences using the `setprefs` request handler. For more information, see the *QuarkXPress Server Web Integration Guide*.

General Preferences — Server

The **Server** tab (**Administration > Preferences > General > Server**) includes the following controls.

Use the **Document Root Folder** field to specify the location of the document pool.

- ➔ Putting the document pool on a network connected drive is not recommended, because this negatively impacts the performance of QuarkXPress Server.

Use the **Allow Memory Caching** check box to control whether memory is cached.

Use the **Max Memory Cache Size** field to specify the maximum memory size allocated to the cache memory. Valid values are from 10MB to 1024MB.

Use the **Force Served Documents Closed** check box to control whether QuarkXPress Server closes projects from the document pool after it renders them, regardless of the Telegraph XTensions software setting. Uncheck this box to keep such projects open on the server.

Use the **Default Renderer Type** drop-down menu to specify the default rendering format for the server.

- **AVE:** Returns a .zip file containing an App Studio issue (.zave) file and its manifest.
- **ePUB:** Returns an ePUB file.
- **EPS Document:** Returns an Encapsulated PostScript (EPS) file.
- **JPEG:** Returns a JPEG file.
- **PDF:** Returns a PDF file.
- **PNG:** Returns a Portable Network Graphics (PNG) file.
- **PostScript:** Returns a PostScript file.
- **QCDDOC:** Returns a QuarkCopyDesk article.
- **QuarkXPress Document:** Returns a QuarkXPress project.
- **Raw Custom:** Returns a file in internal QuarkXPress format for use by server XTensions software developers.
- **RLE Raw Custom:** Returns a file in internal QuarkXPress format (compressed using Run Length Encoding) for use by server XTensions software developers.
- **SWF:** Returns a SWF (Flash) file.

Use the **Scale** field to specify the default scale percentage at which QuarkXPress Server should render projects.

Use the **Disable QXD Return** check box to specify whether QuarkXPress Server can return QuarkXPress projects to clients.

General Preferences — Log

The **Log** tab (**Administration > Preferences > General > Log**) includes the following controls.

Use the **Log Document Problems** check box to specify whether to include problem descriptions in transaction log files. The "Log" folder inside the QuarkXPress Server application folder contains three log files:

- **QuarkXPress Server Fatal Log.log:** This log lists all fatal errors.
- **QuarkXPress Server Log.log:** The log for the Java process. This log contains source code-level logging information that can be useful in troubleshooting.

- **QuarkXPress Server Transaction Log.log:** This log lists all transactions and all errors.

➔ Detailed application logging is enabled for these log files only if it is enabled in the "log4j.xml" file.

To log detailed transaction timing data, check **Log Timing Data**.

Logged problems include the following:

- **Missing Fonts:** If fonts are missing from a project that has been requested for rendering, a one-line description of each missing font is added to the error log. If QuarkXPress Server receives a request to render a project and does not have access to the fonts required by the project, it uses the fonts specified in the **Fonts** tab of the **Preferences** dialog box (**Administration > Preferences > General**). If these fonts are also unavailable, QuarkXPress Server substitutes Helvetica (Mac OS) or Arial (Windows). This behavior is the same as it is in QuarkXPress.
- **Missing Pictures**
- **Missing SXTs:** If a required server XTensions module is missing when a rendering request is received, a one-line description of each missing module is added to the error log. If the name of the missing module is not returnable, the XTensions module ID number is returned.
- **Text Encoding/Character Set Problems:** If text is sent to a text box in the template and the system does not have access to the correct font glyph, the issue is logged. The log data indicates the character set that the system attempted to convert. For example, the log entry might show that a request for Japanese characters was sent to an English project.

➔ Information about missing fonts and missing pictures is also recorded in the "QuarkXPressServerRenderer.log" file. This file also contains detailed timing information, including the transaction UID for each transaction. This log file can be found here:

- **Mac OS:** [drive]/Users/[user name]/Library/Logs/Quark
- **Windows:** [drive]:\Users\[user name]\AppData\Quark\Log or [drive]:\Documents and Settings\[user name]\Application Data\Quark\Log

➔ To download all logs to a non-server computer, click **Show Transaction Log** in the QuarkXPress Server Web interface, then click **Download Logs** on the home page.

General Preferences — Email

QuarkXPress Server can automatically notify someone by e-mail when the checked-out QLA license is about to expire. The **Email** tab (**Administration > Preferences > General > Email**) lets you specify where such e-mail messages should be sent.

Use the **Server** field to specify the domain name or IP address of the e-mail server that QuarkXPress Server should use to send messages (for example, mail.quark.com).

Use the **Port** field to specify the port number of the e-mail server that QuarkXPress Server should use to send messages. Valid values are from 0 to 255.

Use the **From** to specify the source e-mail address the QuarkXPress Server application should use to send messages (for example, `QXPSServer1@quark.com`).

Use the **To** to specify the e-mail address to which the QuarkXPress Server application should send messages (for example, `QXPServerAdmin@quark.com`).

General Preferences — Authentication

The **Authentication** tab (**Administration > Preferences > General > Authentication**) includes the following controls.

Check **Authenticate For Admin Requests** to enable the **Username** and **Password** fields. These fields let you control access to administrative parameters and features. You can enter up to 32 characters.

Renderer Preferences dialog box

Use the renderer **Preferences** dialog box (**Administration > Preferences > Renderer**) to set up default preferences for use when projects are created in QuarkXPress Server. It includes the tabs described in the following topics.

- ➔ Many of these preferences come into play only when you create a project in QuarkXPress Server, either using a `construct` request or through a request handled by a custom QuarkXPress Server XTensions module.
- ➔ You can also set renderer preferences using the `setrendererprefs` request handler. For more information, see the *QuarkXPress Server Web Integration Guide*.
- ➔ In versions of QuarkXPress Server prior to 9.0, these resources were accessed through the **Document Controls** submenu.

Preferences — Display

The **Display** pane (**Administration > Preferences > Renderer > Display**) includes the following controls.

Use the **Color TIFFs** drop-down list to specify the color depth of screen previews created for color TIFFs when they are imported.

Use the **Gray TIFFs** drop-down list to specify the resolution of screen previews created for grayscale TIFFs when they are imported.

Choose a profile that corresponds to your monitor from the **Monitor Profile** drop-down menu, or choose **Automatic**. Profiles can be placed in the "Profiles" folder in the QuarkXPress Server application folder.

Preferences — Input Settings

The **Input Settings** tab (**Administration > Preferences > Renderer > Input Settings**) includes the following controls.

Check **Smart Quotes** to force QuarkXPress to replace feet (") and inches (") marks automatically with the specified quotation marks.

Use the **Format** drop-down menu to specify the default characters to be used with the **Smart Quotes** feature and **Convert Quotes** option in the **Get Text** dialog box (**File > Get Text**).

To change the separators used for indicating sequential ranges for output, enter a value in the **Sequential Page Range Separator**. This value overrides the preferences set for a project.

To change the separators used for indicating nonsequential ranges for output, enter a value in the **Non Sequential Page Range Separator**. This value overrides the preferences set for a project.

Preferences — Font Fallback

The **Font Fallback** pane (**Administration > Preferences > Renderer > Font Fallback**) includes the following controls.

Check **Font Fallback** to activate the Font Fallback feature. When this feature is active, if the application encounters a character it cannot display in the current font, it attempts to find a font that can display the character.

If the application encounters a missing font when opening a project, it uses the preferences in this pane to determine which substitute fonts to use.

➡ If you add characters to an existing project and the font cannot support those characters, the application will search the system for a font that can display the characters.

Check **Search** to have the application search for a suitable font that is used in the active project. To restrict the search to a particular range, choose an option from the **Search Type** drop-down menu. To search the entire story where a missing font occurs, choose **Active Story**. To search a particular number of paragraphs in both directions, choose **Paragraph** and enter a number in the **Search Limit** field.

To indicate which fallback fonts should be used when no other font can be found (taking the **Search** settings into account), choose options from each of the drop-down menus in the **Font List** area.

To indicate which font should be used for the slug line when a layout is printed with registration marks turned on, choose an option from the **Slug Line Font** drop-down menu.

Preferences — Open and Save

The **Open and Save** pane (**Administration > Preferences > Renderer > Open and Save**) includes the following controls.

Choose an option from the **Encoding** drop-down menu to indicate how the applications should display characters in non-Unicode text.

Preferences — Fonts

The **Fonts** pane (**Administration > Preferences > Renderer > Fonts**) includes the following controls.

To specify default replacement fonts, check **Specify Default Replacement Font** and choose options from the **Roman** and **East Asian** drop-down menu.

To highlight characters that are in a Traditional Chinese encoding's UDA/VDA (User Defined Area/Vendor Defined Area) range so that these characters can be visually verified, check **Highlight character ranges defined by Traditional Chinese font vendors**.

Preferences — EPS

The **EPS** pane (**Administration > Preferences > Renderer > EPS**) includes the following controls.

To control whether the application should generate a preview of an EPS file or use the preview (if any) embedded in the file, choose an option from the **Preview** drop-down list. The option specified in this pane is used only when the EPS preview is being created. If you change the preference, you need to reimport the EPS file.

Preferences — PDF

Use the **PDF** pane of the **Preferences** dialog box (**Administration > Preferences > Renderer > PDF**) to set preferences for rendering in PDF format.

➔ The **PDF** pane displays only if PDF Filter XTensions software is loaded. For more information, see "[PDF Filter XTensions software](#)"

Use this pane to specify a PDF workflow:

- Click **DirectPDF** to generate PDF output in the browser. This is the default option.
- Click **PDFtoFolder** to generate and save PDF files to a folder. Click **Select/Browse** to specify a location for the folder in the **Watched Folder** field.
- Click **PS4D** (PostScript File for Later Distilling) to generate a PostScript file. Click **Select/Browse** to specify a location for the folder in the **Watched Folder** field.

Preferences — Project General Settings

The **Project General Settings** pane (**Administration > Preferences > Renderer > Project General Settings**) includes the following controls.

Check **Use OpenType Kerning** to activate the default kerning values for OpenType fonts. When OpenType kerning is active, it overrides any kerning specified through **Kerning Table Edit** (Utilities menu) for OpenType fonts.

To disable OpenType kerning for full-width characters, check **Do Not Kern Full Width Characters**.

Preferences — Print Layout Settings

The **Print Layout Settings** pane (**Administration > Preferences > Renderer > Print Layout Settings**) includes the following controls.

Use the **Master Page Items** drop-down menu to control what happens to master items when master pages are applied.

- Click **Keep Changes** if you intend modified master items on your layout pages to remain when a new master page is applied. The items that are kept are no longer master items.
- Click **Delete Changes** if you want modified master items on your layout pages to be deleted when a new master page is applied.

Use the **Framing** drop-down menu to specify whether frames are placed inside or outside text and picture boxes.

- When you click **Inside**, the distance between the text and the frame is determined by the box's **Text Inset** values (**Item > Modify**). When you place a frame inside a picture box, the frame overlaps the picture.
- When you click **Outside**, the frame is placed outside the box, increasing the box's width and height. The frame cannot extend beyond a constraining box or the pasteboard.

Use the **Auto Page Insertion** drop-down menu (Print layouts only) to determine whether pages are inserted automatically to contain text overflow from an automatic text box or a chain of text boxes (on a page associated with a master page that contains an automatic text box). The drop-down menu also enables you to determine where any pages will be inserted.

Preferences — Print Layout Measurements

The **Print Layout Measurements** pane (**Administration > Preferences > Renderer > Default Print Layout Measurements**) includes the following controls.

Use the **Horizontal** and **Vertical** drop-down menus to specify the measurement system for the rulers displayed along the top and left of the layout window. **Horizontal** corresponds to the top ruler; **Vertical** corresponds to the left ruler.

Use the **Points/Inch** field to override the default value of 72 points per inch. The application uses the value here as the basis for all point and pica measurements, as well as for all point-and-pica-to-inch conversions. The desktop publishing standard for points per inch is 72. However, the traditional typographic standard used on most metal typographic rulers is usually approximately 72.27 or 72.307 points per inch (range = 60 to 80 pt, measurement system = points, smallest increment = .001).

Use the **Ciceros/cm** field to specify a ciceros-to-centimeter conversion value different from the standard 2.1967 (range = 2 to 3 c, measurement system = ciceros, smallest increment = .001).

Preferences — Paragraph

The **Paragraph** pane (**Administration > Preferences > Renderer > Paragraph**) includes the following controls.

Use the **Auto Leading** feature to automatically set line spacing. Unlike paragraphs with absolute leading (identical line spacing above every line), paragraphs with auto leading may include lines with different leading when fonts and font sizes are mixed in the same paragraph.

Auto leading starts with a base amount of leading, which the application calculates by examining the ascent and descent values built into the fonts used in an auto-led line and the line above it; however, the user-specified text size plays the largest part in determining this base amount. Finally, a value specified by the user in the **Auto Leading** field is added to the base amount to arrive at the total amount of leading.

To specify percentage-based auto leading, enter a value from 0% to 100% in 1% increments. This value determines the amount of leading between two lines of text as follows: The largest font size in the line above is multiplied by the percentage value. This outcome is added to the base amount of auto leading between the two lines. Although the design of certain fonts complicates the process, here is a simplified example: 10-point text styled consistently in a "standard" font with **Auto Leading** set to 20% has 12 points of leading (10 pts + [20% of 10] = 12 pts).

Use the **Maintain Leading** check box to control the placement of a line of text that falls immediately below an obstruction in a column or box. If **Maintain Leading** is checked, the line's baseline is placed according to its applied leading value. If **Maintain Leading** is unchecked, the ascent of the line will abut the bottom of the obstruction or any applied runaround value.

In the **Lock to Grid Based On** area:

- Click **Ascent and Descent** to lock text to grid based on the ascenders and descenders of characters.
- Click **Font Size (Em Box)** to lock text to grid based on the size of the em boxes of the characters.

Preferences — Character

The **Character** pane (**Administration > Preferences > Renderer > Character**) includes the following controls.

Use the **Superscript** fields to control the placement and scale (size) of superscript characters. The **Superscript Offset** value determines how far below the baseline the application places a superscript character. The **Superscript Offset** value is measured as a percentage of font size. The default value is 33%. The **Superscript VScale** value determines the vertical size

of the character and is a percentage of font size. The **Superscript HScale** value determines width and is a percentage of the normal character width (as specified by the font designer). The default value for both scales is 60% (range = 0 to 100%, measurement system = percentage, smallest increment = .1).

Use the **Subscript** fields to control the placement and scale (size) of subscript characters. The **Subscript Offset** value determines how far above the baseline the application places a subscript character. The **Subscript Offset** value is measured as a percentage of font size. The default value is 33%. The **Subscript VScale** value determines the vertical size of the character and is a percentage of font size. The **Subscript HScale** value determines width and is a percentage of the normal character width (as specified by the font designer). The default value for both scales is 100% (range = 0 to 100%, measurement system = percentage, smallest increment = .1).

Use the **Small Caps** fields to control the scale of characters with the **Small Caps** type style applied to them. The **Small Caps VScale** value determines the vertical size of the character and is measured as a percentage of font size. The **Small Caps HScale** value determines width and is measured as a percentage of the normal character width (as specified by the font designer). The default value for both scales is 75% (range = 0 to 100%, measurement system = percentage, smallest increment = .1).

Use the **Superior** fields to control the scale of superior characters. The **Superior VScale** value determines the vertical size of the character and is measured as a percentage of font size. The **Superior HScale** value determines width and is measured as a percentage of the normal character width (as specified by the font designer). The default value for both scales is 60% (range = 0 to 100%, measurement system = percentage, smallest increment = .1).

Use the **Ligatures Break Above** field to use ligatures built into a font. A ligature is a typographic convention in which certain characters are combined into a single glyph. Most fonts contain ligatures for the characters "f" followed by "i" and "f" followed by "l". The **Ligatures Break Above** field enables you to specify the kerning or tracking value (measured in 1/200 em space increments) above which characters will not be combined into ligatures. For example, a headline with a large tracking value would probably not contain ligatures. The default value is 1 (range = 0 to 10, measurement system = .005 [1/200] em space, smallest increment = .001). To prevent the second two letters in "ffi" and "ffl" (as in office and waffle) from being combined into ligatures, check **Not "ffi" or "ffl"**. Three-character ligatures for these combinations, common in traditional typesetting systems, are not standardized in fonts designed for Mac OS, so some typographers prefer to keep all three letters separate rather than combine only two of them. Note that many PostScript fonts do not have "ffi" and "ffl" ligatures, but most OpenType fonts do. This option is unchecked by default.

Check **Auto Kern** to specify that the application should use kerning tables, which are built into most fonts, to control intercharacter spacing. The **Auto Kern Above** field enables you to specify the point size above which automatic kerning must be used. The **Auto Kern Above** feature also implements custom tracking information specified in the **Tracking Values** dialog box for a selected font (**Utilities > Edit Tracking**) in QuarkXPress. This option is checked by default, with a 4-point threshold (range = 0 to 72 pt, measurement system = various [", pt, cm, etc.], smallest increment = .001).

Check **Standard Em Space** to specify an em-space equivalent to the point size of the text (for example, 24pt text has a 24pt em space). If **Standard Em Space** is unchecked, the application uses the width of the two zeros in the current font as the em-space width. This option is checked by default. You can insert an em space in text by pressing Option+space/Ctrl+Shift+6.

Use the **Flex Space Width** field to change the 50% default width of a flexible space. To create a breaking flexible space, press Option+Shift+space/Ctrl+Shift+5; to create a nonbreaking flexible space, press Command+Option+Shift+space/Ctrl+Alt+Shift+5. The **Flex Space Width** value is expressed as a percentage of the normal en space for a given font and font size (range = 0 to 400%, measurement system = percentage, smallest increment = .1).

Use the **Accents for All Caps** check box to specify whether to include accent marks on accented characters with the All Caps type style applied. This option is checked by default.

Use the **Space between CJK & R** field to indicate how much space should be included by default between a Chinese, Japanese, or Korean character and an adjacent Roman character.

Preferences — Trapping

The **Trapping** pane (**Administration > Preferences > Renderer > Trapping**) includes the following controls.

Select a **Trapping Method**:

- Click **Absolute** to trap using the values in the **Auto Amount** and **Indeterminate** fields according to the object and background colors involved. If the object color is darker, the object is choked by the background using the **Auto Amount** value. If the object color is lighter, the object is spread into the background using the **Auto Amount** value.
- Click **Proportional** to trap using the value in the **Auto Amount** field multiplied by the difference between the luminance (lightness or brightness) of the object color and the background color.
- Click **Knockout All** to turn off trapping and print objects with a zero trap amount.

Check **Process Trapping** to trap each process separation plate individually when a page contains overlapping process colors.

Check **Ignore White** to specify that an object color in front of multiple background colors (including white) does not take white into account when trapping.

Enter a trapping value in the **Auto Amount** field or choose **Overprint**:

- Enter a value in the **Auto Amount** field to control the amount of trapping that QuarkXPress applies to object and background colors that have an **Auto Amount** specified in the **Trap Specifications** dialog box (**Edit > Colors > Edit Trap**), and to control the amount of trapping applied to items with an **Auto Amount (+) Trap Information** or **(-) specified** in the **Trap Information** palette (**Window > Trap Information**).

- Choose **Overprint** to cause overprinting for object and background colors with an **Auto Amount** specified in the **Trap Specifications** dialog box (**Edit > Colors > Edit Trap**), as well as for items with an **Auto Amount (+)** or **(-)** specified in the **Trap Information** palette (**Window > Trap Information**).

Enter a trapping value in the **Indeterminate** field or choose **Overprint**:

- Enter a value in the **Indeterminate** field to control the amount of trapping that QuarkXPress applies to object colors that are in front of indeterminate backgrounds (multiple colors with conflicting trapping relationships).
- Choose **Overprint** to cause an object color to overprint an indeterminate background.

Enter a **Knockout Limit** value. The knockout limit is the value (expressed as a percentage of darkness of the object color) that enables you to control the point at which an object color knocks out a background color.

Enter an **Overprint Limit** value. Overprint limit is a trapping setting that allows an object set to overprint to trap according to the **Auto Amount** value if the object's shade is less than a particular percentage.

Preferences — Color Manager

The **Color Manager** pane (**Administration > Preferences > Renderer > Color Manager**) includes the following controls.

To specify an engine for color transformation, choose an option from the **Color Engine** drop-down menu.

To achieve the darkest possible blacks in all output methods, check **Black Point Compensation**.

Use the **Source Setup** drop-down menu to specify the source color space of pictures and colors used in the application.

To enable the **Profile Information** command in the **Window** menu and the **Color Management** tab in the **Import Picture** dialog box, check **Enable Access to Picture Profiles**. This option allows you to view information about profiles.

To specify a default proof output setup, choose an option from the **Proof Output** drop-down menu.

To specify a rendering intent for soft proofing, choose an option from the **Rendering Intent** drop-down list. **Perceptual** scales all the colors in the source gamut so that they all fit within the destination gamut. **Relative Colorimetric** retains colors that are in both the source gamut and the destination gamut. The only source colors that are changed are those that are not within the destination gamut. **Saturation** considers the saturation of source colors and changes them to colors with the same relative saturation in the destination gamut. **Absolute Colorimetric** retains colors that are in both the source gamut and the destination gamut. Colors that are outside the destination gamut are adjusted in

relation to how they would look when printed on white paper. **Defined by Sources** uses the rendering intents defined in source setup for all colors and images.

To color manage vector content in imported EPS and PDF files, check **Color Manage Vector EPS/PDF**. Note that this preference applies only to EPS and PDF files imported after this box is checked.

To color manage vector content in EPS and PDF files that have already been imported in the active project, check **Include Existing Vector EPS/PDF in Layout**.

Preferences — Layers

The **Layers** pane (**Administration > Preferences > Renderer > Layers**) has the following controls.

To make new layers visible by default, check **Visible**.

To suppress the printout of new layers by default, check **Suppress Output**.

To make new layers locked by default, check **Locked**.

To maintain runaround on new layers so that text on visible layers flows around items on hidden layers, click **Keep Runaround**.

Preferences — Full Res Preview

Use the **Full Res Preview** pane (**Administration > Preferences > Renderer > Full Res Preview**) to control how the Full Res Preview XTensions module works.

To turn off Full Resolution Preview when opening a project, check **Disable Full Resolution Previews on Open**.

To specify a maximum cache folder size for full resolution previews, enter a value in the **Maximum Cache Folder Size** field.

Job Jackets dialog box

The **Job Jackets** dialog box lets you edit the Job Jackets file used by QuarkXPress Server. To edit the QuarkXPress Server Job Jackets file:

- 1 In the QuarkXPress Server Web interface, choose **Administration > Job Jackets**. The **Manage Job Jacket** dialog box displays.
- 2 Click the **Get Job Jacket for editing** button and save the Job Jackets file to the desktop.
- 3 Open the downloaded Job Jackets file in QuarkXPress and make any necessary changes to the **QXPSJobTicket** Job Ticket.
- 4 In the **Job Jacket** dialog box, click **Choose File** and select the modified Job Jackets file.
- 5 Click **Submit**. The QuarkXPress Server Job Jackets file is replaced with the modified version.

Check Out License dialog box

To check the QuarkXPress Server license out of Quark License Administrator, choose **Administration > Check Out License**. To specify the number of days for checkout, enter a value in days in the **Check out for** field. To be warned in advance of license expiration, check **Warn me** and use the corresponding fields.

Creating URL requests

You can use URL requests to make QuarkXPress Server render projects in a variety of formats, to use the features of server XTensions modules, and to control the server. The topics below provide an overview of how to construct server requests and use URL parameters.

This chapter also lists functions that let you control the server. For detailed information about constructing other types of URL requests, see the QuarkXPress Server *Web Integration Guide*.

Understanding URL requests

QuarkXPress Server URL requests should use the following format:

```
http://server:port/namespace/path/projectname?parameter=value
```

- **server**: Indicates the name or IP address of the QuarkXPress Server computer.
- **port**: Indicates the QuarkXPress Server application's port number. The default port number is 8080.
- **namespace**: Sets the render type (or indicates another server functionality to access). For more information, see "[Understanding QuarkXPress Server namespaces](#)".
- **path**: Indicates the path to the directory where the target project file is stored. The project to be rendered can either be located in the document pool (in which case paths are evaluated relative to the document pool directory), or can be streamed as part of a multipart HTTP Post request. When the project is streamed as part of the request, the project name will correspond to the name given to the HTTP request part which contains the project data.
- **projectname**: Identifies the project to be rendered.
- **parameter=value**: Optional parameters that provide more detailed control over how the target project should be rendered. Multiple parameter/value pairs, separated by the "&" character, can be included.

For example, the following URL asks the QuarkXPress Server application named "QXPServer" to return the file "MyProject.qxp" as a PDF file with hyperlinks and all fonts embedded:

```
http://QXPServer:8080/PDF/MyProject.qxp?includehyperlinks=1&embedallfonts=1
```

Some URL parameters require Boolean arguments. For such parameters, valid values include 1 or 0, true or false, y or n, and yes or no.

- ➔ You can also send requests to QuarkXPress Server using the HTTP GET and POST protocols and using XML with XSLT. For more information about these approaches, see the QuarkXPress Server *Web Integration Guide*.

Understanding QuarkXPress Server namespaces

QuarkXPress Server namespaces differentiate among types of requests that are otherwise identical. For example, consider the following three URLs:

```
http://QXPServer:8080/project1.qxp
```

```
http://QXPServer:8080/pdf/project1.qxp
```

```
http://QXPServer:8080/postscript/project1.qxp
```

These requests are identical, except each uses a different namespace (in *italic*). (The first request does not specify a namespace, but this simply means the project is to be rendered using the server's default render type.)

Namespaces can be used to determine the format in which a rendered project is returned, as indicated above, but they can also be used to direct a request to XTensions software that performs other functions. For example, if you use Modifier XTensions software's `xml` namespace, Modifier XTensions software can return an XML representation of the project.

Looking up a namespace

The QuarkXPress Server *Web Integration Guide* lists the namespaces for every QuarkXPress Server function. There is no single list of namespaces because some functions do not require a particular namespace or are available in multiple namespaces. To determine which namespace you want to use:

- 1 Open the QuarkXPress Server *Web Integration Guide*.
 - 2 Choose **Complete API List** from the **QuarkXPress Server Functions** menu at the top of the page.
 - 3 Click the link for the render type you want or the function you want to use. (The link might be a dynamic publishing process, a renderer, or a request handler.) The page corresponding to that render type or function displays.
 - 4 Locate the **Namespace** row. If the render type or function has an associated namespace, that namespace is listed here.
- ➔ Third-party XTensions can add their own namespaces. For information about a third-party namespace, see the documentation for the XTensions module that adds that namespace.

Understanding QuarkXPress Server parameters

Parameters let you control the details of how a request is executed. For example, you can use the `page` parameter to create a request that returns only the third page of a project:

```
http://QXPServer:8080/jpeg/project1.qxp?page=3
```

You can include multiple parameters in the same request; simply separate them with an ampersand (&). For example, here's a new version of the above URL that returns page three at a scale of 50%:

```
http://QXPServer:8080/jpeg/project1.qxp?page=3&scale=.5
```

Looking up a parameter

The QuarkXPress Server *Web Integration Guide* lists the parameters that are available for every QuarkXPress Server function. To determine which parameters you can use with a request:

- 1 Open the QuarkXPress Server *Web Integration Guide*.
- 2 Choose **Complete API List** from the **QuarkXPress Server Functions** menu at the top of the page.
- 3 Click the link for the render type you want or the function you want to use. (The link might be a dynamic publishing process, a renderer, or a request handler.) The page corresponding to that render type or function displays.
- 4 Locate the **Parameters** row. This row lists all available parameters, and includes a description and a list of valid values for each parameter.

QuarkXPress Server Web Integration Guide overview

The topics below provide an overview of the contents of the QuarkXPress Server *Web Integration Guide*.

Render types

QuarkXPress Server render types let you render content in a variety of formats, including AVE, ePUB, EPS, JPEG, PNG, PDF, PostScript, QuarkXPress, QuarkCopyDesk, and SWF (Flash). You can also return documents in two different "raw" formats. Developers can implement additional rendering formats through server XTensions software.

For more information, see the QuarkXPress Server *Web Integration Guide*.

General parameters

General parameters are parameters that are applicable to most render types. These parameters allow you to render an individual box, component, layout, page, or spread. You can also render a particular selection of layers, pages, and spreads. In addition, you can control whether the pasteboard is included and how the output should be scaled.

For more information, see the QuarkXPress Server *Web Integration Guide*.

Specialized parameters

Specialized parameters are parameters that serve special purposes. Some of these parameters work by asking the server to read the contents of XML files.

For more information, see the QuarkXPress Server *Web Integration Guide*.

Server control parameters

Server control parameters let you perform a variety of server administration tasks, including adding and deleting files from the document pool, retrieving information about files, and flushing caches.

For more information, see the QuarkXPress Server *Web Integration Guide*.

Document providers

Document providers allow QuarkXPress to render the following types of documents:

- **copydesk**: Allows QuarkXPress Server to render article files created by QuarkCopyDesk.
- **qcd doc**: Allows QuarkXPress Server to render and export QuarkCopyDesk articles.

The Streaming Document Provider provides an alternative delivery mechanism for projects, images, digital enrichments and modifier XML to be deployed to the server.

For more information, see the QuarkXPress Server *Web Integration Guide*.

Modifying layouts with XML

The `xml` namespace deconstructs a project according to the Modifier DTD. The `construct` namespace lets the server turn an XML representation of a QuarkXPress project back into an actual project. With these namespaces, you can deconstruct a project into an XML representation, change the XML in accordance with the Modifier DTD, and then have the server generate an updated version of the QuarkXPress project. You can even create new QuarkXPress projects from scratch using XML.

In addition, you can use the `construct` namespace to:

- Create a page based on master page
- Create a project from XML, using a Job Jackets™ file as the basis for the project
- Modify text font and style, including OpenType® styles
- Apply style sheets and local formatting to text
- Create and populate tables
- Import pictures into picture boxes and specify picture attributes

The DTD used for XML construction and deconstruction is completely Unicode®-compliant, making it ideal for use in international publishing. Furthermore, the use of this DTD ensures that the schema of XML output created by Constructor does not change when server preferences change. This DTD is provided in the QuarkXPress Server application folder and fully documented in the QuarkXPress Server *Web Integration Guide*.

➡ Deconstructor XTensions software and the `deconstruct` namespace are no longer supported.

Using SSL

You can configure QuarkXPress Server with different security options. In addition to your own network security specifications, you can specify Secure Sockets Layer (SSL) protocol for client applications.

Secure Sockets Layer (SSL) support

You can configure Tomcat (and therefore all QuarkXPress Server clients) to run in secure mode with Secure Sockets Layer (SSL) technology. This section explains the configuration process.

➡ It is also possible to run QuarkXPress Server without embedding Tomcat in the JVM. For more information, see the QuarkXPress Server *ReadMe* file.

To manage Web applications in the QuarkXPress Server environment, QuarkXPress Server embeds an instance of Apache Tomcat 6.18 in its JVM.

When you enable SSL, it applies to all QuarkXPress Server client applications.

Enabling SSL

The instructions below address two scenarios. The "server.xml" file you edit contains XML tags for both scenarios, which you need to enable or disable by "commenting" and "uncommenting" specific tags.

To enable SSL for secure HTTP for all QuarkXPress Server applications:

- 1 Open the "conf" folder in your QuarkXPress Server folder.
- 2 Open "server.xml" in a text-editing application.
- 3 Locate the following tag (preceded by the comment `<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->`) and comment it out.

```
<Connector port="8080" maxHttpHeaderSize="8192" maxThreads="150"
minSpareThreads="25" maxSpareThreads="75" enableLookups="false"
redirectPort="61399" acceptCount="100" connectionTimeout="20000"
disableUploadTimeout="true" URIEncoding="UTF-8"/>
```

- 4 Locate the following tag (preceded by the comment `<!-- Define a SSL HTTP/1.1 Connector on port 61399 -->`) and uncomment it.

```
<Connector port="61399" maxHttpHeaderSize="8192"MaxThreads="150"
minSpareThreads="25" maxSpareThreads="75"enableLookups="false"
disableUploadTimeout="true"acceptCount="100" scheme="https"
secure="true"clientAuth="false" sslProtocol="TLS" />
```

- 5 Replace 61399 with 61400 (or any port on which Tomcat will be listening for secure connections).
- 6 Save and close "server.xml."
- 7 Open the "ServerApp.properties" file (in the "conf" folder) and enter the port number from step 5 for `qxpswebserver.port`.

➔ This change means QuarkXPress Server client applications can use HTTPS. For example, the URL for a QuarkXPress Server user would be as follows: `https://[server name]:[port]/`.

Enabling HTTP and HTTPS

To enable HTTP and HTTPS:

- 1 Open the "conf" folder in your QuarkXPress Server folder.
- 2 Open "server.xml" in a text-editing application.
- 3 Uncomment the following tag:

```
<Connector port="61399" maxHttpHeaderSize="8192"MaxThreads="150"
minSpareThreads="25" maxSpareThreads="75"enableLookups="false"
disableUploadTimeout="true"acceptCount="100" scheme="https"
secure="true"clientAuth="false" sslProtocol="TLS" />
```

- 4 Save and close "server.xml."

➔ This feature allows QuarkXPress Server application users to access QuarkXPress Server with HTTPS or HTTP.

Verifying and using SSL

To verify and use SSL:

- 1 Start the QuarkXPress Server
- 2 Test QuarkXPress Server access by navigating to the QuarkXPress Server Web interface HTTPS. For example: `https://[server]:[port]/qxpsadmin`

Keystores and SSL certificates

A *certificate* is a file on a Web server that is used in encryption and confirmation between two endpoints to establish a secure connection. A *keystore* is essentially a database of digital certificates on the Web server.

You can obtain an SSL certificate from a trusted Certificate Authority (CA). Import the certificate into the keystore used by QuarkXPress Server's JVM.

For more information about the importance of keystores, use the following URL:

<http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html>.

QuarkXPress Server XTensions software

Just as XTensions software provides additional functionality to QuarkXPress, XTensions software enables QuarkXPress Server to do things it can't do by default. The XTensions modules included with QuarkXPress Server allow clients to render projects as PDF files, apply QuarkVista picture effects to pictures, dynamically update pictures in picture boxes and text in text boxes (as well as boxes themselves), import data on the fly, manipulate layers in projects, and more.

CopyDeskArticle XTensions software

CopyDeskArticle XTensions software allows QuarkXPress Server to do the following things:

- Render QuarkCopyDesk articles
- Export QuarkCopyDesk articles from a QuarkXPress project
- Add a QuarkCopyDesk article to an existing QuarkXPress project
- Create and delete components in a QuarkCopyDesk article within a QuarkXPress project

Rendering articles

To render QuarkCopyDesk articles, use the `copydesk` namespace, as follows:

```
http://[server]:[port]/[render type]/copydesk/[articlename]
```

For example, to render "Article.qcd" as a PDF file, you could use a URL like the following:

```
http://QXPServer.8080/pdf/copydesk/Article.qcd
```

Exporting articles

To export an article from a QuarkXPress project, use the `QCDDOC` namespace, as follows:

```
http://[server]:[port]/qcddoc/[project name]?article=[article name or ID]
```

For example, to export the article named "Article1" from the project named "Project1.qxp," you could use a URL like the following:

```
http://QXPServer.8080/qcddoc/Project1.qxp?article=Article1
```

- ➡ You cannot export an article from a QuarkXPress project unless the article has been created and named in QuarkXPress using CopyDeskArticleXT XTensions software.

To export an article in a particular format, use the `format` parameter.

- To export a standard article, use `format=fullfeatured`.
- To export an article in lightweight mode, use `format=lightweight`. This format can be useful in situations where file size is an issue. The lightweight file format includes only those items that a QuarkCopyDesk user is supposed to work on.

For example:

`http://QXPServer.8080/qcddoc/Project1.qxp?article=Article1?format=lightweight`

- ➡ You cannot export page pictures in lightweight mode.

Adding articles to projects

You can use Modifier XTensions software to create a QuarkCopyDesk article within an existing QuarkXPress project. For example, to add an article named "New Article" to the project named "Project1.qxp," you could use XML like the following:

```
<PROJECT PROJECTNAME="Project1.qxp" XMLVERSION="8.0">
  <LAYOUT POINTSPERINCH="72">
    <ID NAME="Layout 1" UID="1"/>
    <ARTICLE OPERATION="CREATE" DOCFORMAT="FULLFEATURED">
      <ID NAME="New Article" UID="4"/>
      <RGBCOLOR BLUE="98" GREEN="254" RED="0"/>
      <COMPONENT BOXNAME="Box1" BOXUID="1" COMPONENTCLASS="CT_TEXT"
        NAME="Component 1" UID="1"/>
    </ARTICLE>
  </LAYOUT>
</PROJECT>
```

For more information, see the QuarkXPress Server *Web Integration Guide*.

Creating and deleting components

You can use Modifier XTensions software to create and delete components in QuarkCopyDesk articles. For example, to add a component named "New Component" to the article named "Article 1" in the project named "Project1.qxp," you could use XML like the following:

```
<PROJECT JOBJACKET="Project1 Job Jacket" JOBTICKET="Default Job Ticket
1:Project1"
  PROJECTNAME="Project1.qxp" XMLVERSION="8.0">
  <LAYOUT POINTSPERINCH="72">
    <ID NAME="Layout 1" UID="1"/>
    <ARTICLE DOCFORMAT="FULLFEATURED">
      <ID NAME="Article 1" UID="1"/>
      <COMPONENT OPERATION="CREATE" BOXUID="9" COMPONENTCLASS="CT_TEXT"
        NAME="New Component"/>
    </ARTICLE>
  </LAYOUT>
</PROJECT>
```

To delete the component named "New Component" from the article named "Article1.qcd," you could use XML like the following:

```
<PROJECT JOBJACKET="Project1 Job Jacket" JOBTICKET="Default Job Ticket"
1:Project1"
  PROJECTNAME="Project1.qxp" XMLVERSION="8.0">
    <LAYOUT POINTSPERINCH="72">
      <ID NAME="Layout 1" UID="1"/>
      <ARTICLE DOCFORMAT="FULLFEATURED">
        <ID NAME="Article 1" UID="1"/>
        <COMPONENT OPERATION="DELETE" NAME="New Component"/>
      </ARTICLE>
    </LAYOUT>
  </PROJECT>
```

For more information, see the QuarkXPress Server *Web Integration Guide*.

PDF Filter XTensions software

PDF Filter XTensions software allows QuarkXPress Server to render a QuarkXPress project as a PDF file. To render QuarkXPress projects as PDF files when PDF is not the QuarkXPress Server default render type, use the **PDF** namespace, as follows:

```
http://[server]:[port]/pdf/[projectname]
```

For information about PDF preferences, see "[Preferences — PDF](#)." To take advantage of more detailed preferences, create a PDF output style and use that output style when rendering projects as PDF files.

For information about the parameters for exporting in this format, see the QuarkXPress Server *Web Integration Guide*.

Modifier XTensions software

Modifier XTensions software lets clients perform all of the following tasks using XML:

- Modify the properties of pictures in a QuarkXPress project
- Modify the text in text boxes within a QuarkXPress project
- Modify the properties of text boxes and picture boxes in a QuarkXPress project
- Create and delete picture boxes and text boxes in a QuarkXPress project
- Import text or text strings into text boxes within a QuarkXPress project
- Import pictures into picture boxes within a QuarkXPress project
- Save modified QuarkXPress projects in any supported format to any location on the network (and also in the QuarkXPress Server document pool)
- Create and delete pages
- Create and delete layers
- Move items within layers
- Create and delete tables

- Modify tables and their contents
- Create QuarkCopyDesk articles and components
- Create lines, anchored boxes, and Bézier boxes
- Create lists
- Group and ungroup boxes
- Divide layouts into sections

To use Modifier XTensions software, a client creates an XML file indicating the actions to be taken and sends that XML file to the QuarkXPress Server application, where Modifier XTensions software reads the XML and makes the requested changes. Clients can use a single XML file or string to manipulate multiple documents and boxes.

For more specific information about Modifier XTensions software, and for the Modifier DTD, see the QuarkXPress Server *Web Integration Guide*.

- ➔ Modifier XTensions software supports both GET and POST functionality.
- ➔ Modifier XTensions software supports XML containing code that uses Unicode UTF-8 and UTF-16 encodings. Use the `encoding` attribute of the XML declaration to specify an encoding, as you would with any other XML file.

Using Modifier XTensions software

To use Modifier XTensions software:

- 1 Create a QuarkXPress project. Note the IDs or names of any text and picture boxes you want to manipulate.
- 2 Upload the project to the QuarkXPress Server document pool.
- 3 Create XML that describes the changes you want, as described in "[Creating XML for Modifier XTensions software](#)."
- 4 Send the XML to the server in one of the following ways:

- Put the XML in a file on the server and then use a URL to point to the file, as follows:

```
http://server:port/namespace/path/projectname?modify=file:[absolute
path to XML file on server or relative path to XML file on server
relative to the document pool]
```

- Put the XML in the URL, as follows:

```
http://[server]:[port]/[namespace]/path/projectname?modify=[XML string]
```

- ➔ Clients can also send XML in the form of a POST request.

Creating XML for Modifier XTensions software

All XML used with Modifier XTensions software uses the Modifier DTD. This DTD is documented in full in the QuarkXPress Server *Web Integration Guide*.

In general, the structure for addressing items in a particular layout is as follows:

```
<PROJECT>
  <LAYOUT>
    <ID NAME="[name of layout]">
      <[Item being addressed]>
        <[Parameters of item]>
      </[Item being addressed]>
    </LAYOUT>
  </PROJECT>
```

For more information, see the QuarkXPress Server *Web Integration Guide*.

Layer XTensions software

You can use QuarkXPress Server Layer XTensions software to control the visibility of specific layers in a rendered QuarkXPress project. You can also add layers, delete layers, edit layer attributes, and control whether layers are rendered.

- ➔ When you use the `getdocinfo` namespace, QuarkXPress Server returns information about items on all of the layers in the QuarkXPress project, including layers that are not visible.

Clients can use the `layer` parameter to specify a layer (even a hidden layer) to be rendered. For example, the URL `http://QXPServer:8080/doc.qxp?layer=layer1` renders only the layer named "layer1" in the project named "doc.qxp."

Clients can specify more than one layer in a single URL. For example, the URL `http://QXPServer:8080/doc.qxp?layer=layer1,layer2` renders the layers named "layer1" and "layer2."

For information about layer preferences, see "[Preferences — Layers](#)." For information about the parameters for this feature, see the QuarkXPress Server *Web Integration Guide*.

- ➔ If **Suppress Output** is selected for a layer, QuarkXPress Server does not render that layer when producing PDF, EPS, or PostScript files.

InteractiveDesigner Server XTensions software

InteractiveDesigner XTensions software allows QuarkXPress Server to render both Print and Interactive layouts in QuarkXPress projects as SWF (Flash) files. If you render a Print layout as an SWF file, you can use the right and left arrow keys to navigate between pages in the SWF file.

To render an Interactive layout in a QuarkXPress project as a SWF file when SWF is not the QuarkXPress Server default render type, use the `SWF` namespace, as follows:

```
http://[server]:[port]/swf/[projectname]?layout="[layoutname]"
```

➡ For information about the parameters for exporting in this format, see the QuarkXPress Server *Web Integration Guide*.

App Studio XTensions software

The App Studio XTensions modules allow QuarkXPress Server to render Print and App Studio layouts in QuarkXPress projects as App Studio issue (AVE) files.

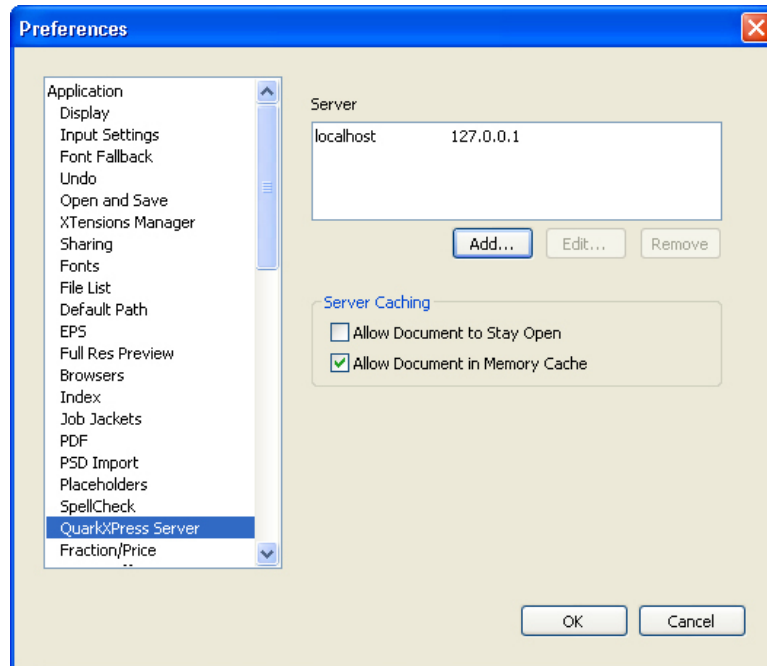
Telegraph XTensions software

Unlike the other XTensions software described in this guide, Telegraph XTensions software works with QuarkXPress, rather than with QuarkXPress Server. You can use Telegraph XTensions software to create QuarkXPress projects that can serve as templates in QuarkXPress Server. Using Telegraph XTensions software, you can assign unique names to individual items, define server caching parameters, and upload the template directly to a QuarkXPress Server computer.

➡ These topics explain how to use Telegraph XTensions software. It is assumed that you are already familiar with the functionality and user interface of QuarkXPress.

Setting Telegraph preferences

Telegraph XTensions software adds the **QuarkXPress Server** pane to the QuarkXPress **Preferences** dialog box (**QuarkXPress/Edit** menu). You can use this pane to configure settings for QuarkXPress Server templates, specify where to store your projects on the server, and control how projects are cached.



QuarkXPress Server pane of Preferences dialog box (QuarkXPress/Edit menu)

For information on adding a server, see "[Adding a server](#)."

To edit the properties of a server in the **Server** list, select the server name and then click **Edit**.

To remove a server from the **Server Setup** list, select the server and then click **Remove**.

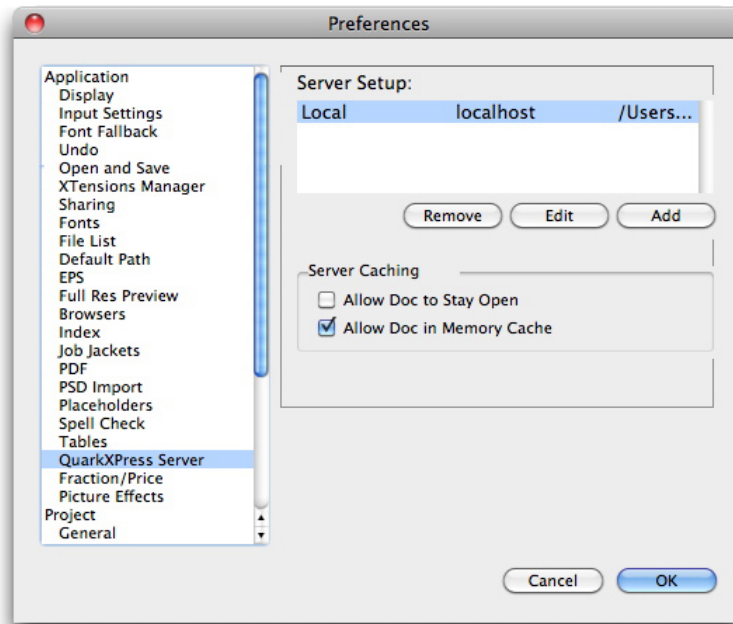
To allow projects checked in from this copy of QuarkXPress to remain open on the server after it has been served, check **Allow Document to Stay Open**.

To load projects checked in from this copy of QuarkXPress into the server memory cache, check **Allow Document in Memory Cache**.

Specifying a server for template upload

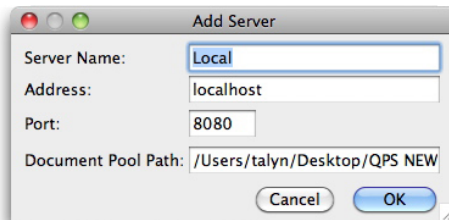
Before you can upload a template to a server with Telegraph XTensions software, you must add that server to the QuarkXPress preferences. To do so:

- 1 Choose **QuarkXPress/Edit > Preferences**. The **Preferences** dialog box displays.
- 2 Click **QuarkXPress Server** in the list on the left. The **QuarkXPress Server** pane displays.



QuarkXPress Server pane of **Preferences** dialog box

- 3 Click **Add**. The **Add Server** dialog box displays.



Add Server dialog box

- 4 Enter a human-readable name for the server in the **Server Name** field.
 - 5 Enter the server's name or IP address in the **Address** field.
 - 6 Enter the server's port number in the **Port** field. The default port number is 8080. Valid values are from 1 to 65535.
 - 7 In the **Document Pool Path** field, enter the path to the document pool directory on the server, or to a subdirectory within the document pool. If you leave this field blank, the path defaults to the document pool directory path specified in the **QuarkXPress Server Document Root** field (**QuarkXPress Server** > **Server Configuration** > **Server** tab).
- ➔ If you enter a folder path that does not exist, QuarkXPress Server can create the folders in the path when you upload the template to a QuarkXPress Server. To create folders when you upload, check **Generate Hierarchy On Document Upload** in the **Server** tab of the QuarkXPress Server **Server Configuration** dialog box (**QuarkXPress Server** > **Server Configuration**) before you upload the template to the server.

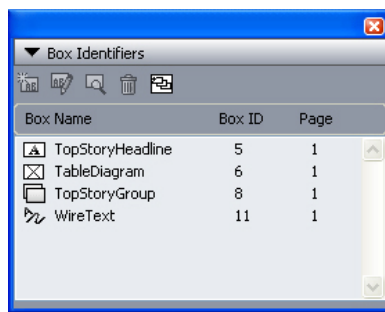
- 8 Click OK, then close the **Preferences** dialog box.

Using Telegraph XTensions software


Once you have configured preferences for Telegraph QuarkXTensions software, you can begin creating QuarkXPress Server templates. After you complete a template, Telegraph XTensions software can upload the file to a QuarkXPress Server computer.


Identifying QuarkXPress items and groups


The **Box Identifiers** palette lets you associate names with items and groups. To display this palette, choose Window > Box Identifiers.




Box Identifiers palette

To edit the name of an item or group, select its name in the **Box Identifiers** palette and then click **Edit Box Name** .

To scroll to the location of a named item or group, double-click the target item or group's name in the **Box Identifiers** palette. Alternatively, you can select the name and click **Go To** .

To delete an item's or group's name (without deleting the item or group itself), select the name in the **Box Identifiers** palette and then click **Delete** .


To make sure that all named boxes display in the **Box Identifiers** palette, click **Populate Named Boxes** .

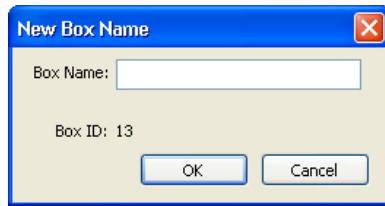
Each QuarkXPress item also has an identification number that you can use when you want to render individual project items in QuarkXPress Server. The number displays in the **Box ID** field in the upper right corner of the palette.

- ➔ In a chain of text boxes, all boxes use the same **Box Name**. However, each box has a unique **Box ID** number.

Naming items and groups

To specify a name for an item or group:

- 1 Using the Item tool, select the target item or group.
- 2 In the **Box Identifiers** palette, click **New Box Name** . The **New Box Name** dialog box displays.



New Box Name dialog box

- 3 Enter a name for the item or group in the **Box Name** field.
 - 4 Click **OK**. The name of the item or group displays on the **Box Identifiers** palette, next to the item's item ID and page number.
- ➔ In addition to items on layout pages, you can also name items on master pages. Items on layout pages that are based on items on master pages have a default name of "<Item name on master page><New box UID>"

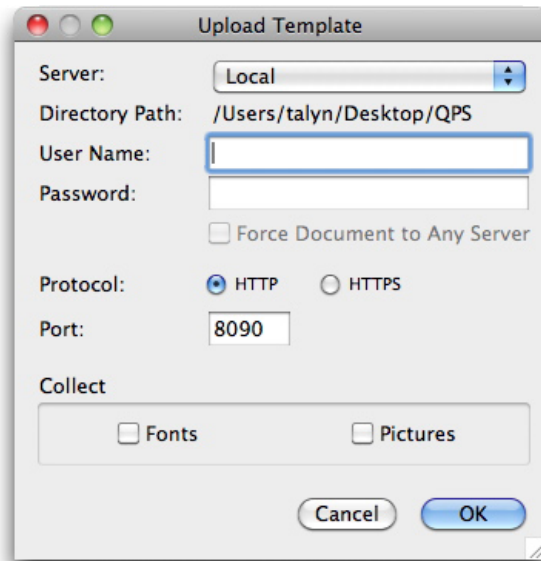
Uploading templates

Telegraph QuarkXTensions software can upload a project to the QuarkXPress Server computer in one simple step. You can also upload any required pictures and fonts, if you choose to collect the fonts and pictures during the upload.

- ➔ QuarkXPress Server automatically generates any file hierarchy necessary when you upload content to the document pool from Telegraph XTensions software.
- ➔ You must have Telegraph XTensions software 9.0 or later to upload templates to QuarkXPress Server 9.0 or later. (You can use previous versions of Telegraph with previous versions of QuarkXPress, but if you do so you must upload projects to the server using the document pool upload capability in the QuarkXPress Server Web interface.)
- ➔ This feature does not upload assets that are used in App Studio interactivity. You must upload such assets manually.

To upload the active project:

- 1 Choose **Utilities > Upload Template**. The **Upload Template** dialog box displays.
- ➔ If you have edited the project since you last saved it, QuarkXPress prompts you to save the project.



Upload Template dialog box

- 2 Choose a server from the **Server** drop-down menu. This drop-down menu includes the servers listed in the **QuarkXPress Server** pane of the **Preferences** dialog box (**QuarkXPress/Edit** menu).
 - 3 If you specified a directory path for the server, that path automatically displays in the **Directory Path** field. If you did not specify a directory path in the preferences, this field remains blank. This path defaults to the document pool directory specified in QuarkXPress Server.
 - 4 If you are uploading to QuarkXPress Server, and you have configured that server to require authentication, enter a valid user name and password in the **User Name** and **Password** fields. (If the QuarkXPress Server application does not require authentication, leave these fields empty.)
- ➔ To specify authentication information for a QuarkXPress Server application, choose **Administration > Preferences > General** in the QuarkXPress Server Web interface, check **Authenticate for Admin Requests** box and enter a user name and password.
- 5 To indicate which protocol to use for uploading, click **HTTP** or **HTTPS**.
 - 6 Enter the port for QuarkXPress Server in the **Port** field.
 - 7 To collect and upload fonts used by the project, check **Fonts**.
 - 8 To collect and upload pictures used by the project, check **Pictures**. This will upload high- or low-resolution pictures that are linked to or embedded in the project.
- ➔ If picture files are missing or have been modified since they were imported into the project, an alert displays. For more information about missing or modified picture files, see *"Uploading missing or modified pictures."*

9 Click **OK**.

➡ If you check **Fonts**, an alert reminds you of possible restrictions regarding copying font software. Click **OK** to continue uploading the project with the fonts, click **Do Not Collect Fonts** to upload the project without the fonts, or click **Cancel** to stop the upload.

The **Upload Status** window displays a progress bar that displays the status of the upload. When the upload is complete, a message notifies you whether the project uploaded successfully.

Uploading missing or modified pictures

If the picture files linked to the project are missing or have been modified since they were imported into the project, an alert displays at upload. Choose from among the following options:

- To continue the upload with low-resolution versions of the pictures, click **OK**.
- To stop the upload, click **Cancel**.
- To locate missing pictures or update modified pictures, click **List Pictures**.

If you click **List Pictures**, the **Missing/Modified Pictures** dialog box displays:

- To view a picture in the project, select the picture's name in the list and click **Show**.
- To locate a missing picture file, select it and click **Update**. The **Find** dialog box displays. Locate and choose the appropriate file, and then click **Open**.
- To update a modified picture file, click **Update**. Every instance of the modified picture in the project is updated.
- When **OK** displays in the **Status** column for each picture, click **Collect**. If the status of any picture is still **Missing** or **Modified** when you click **Collect**, that picture file will not be uploaded, but a low-resolution preview will remain in the project.
- To stop the upload and return to the project window, click **Cancel**.

QuarkXPress Server Manager

QuarkXPress Server Manager is a server application that efficiently routes rendering requests in an environment that uses one or more QuarkXPress Server applications.

QuarkXPress Server Manager uses load-balancing methods to determine which server in the QuarkXPress Server pool can best process a document request, and uses caching to improve speed and efficiency. QuarkXPress Server Manager also provides failsafe capability by reliably resubmitting failed requests, either to the same QuarkXPress Server instance or to a different one (depending on the error message returned by the server instance).

QuarkXPress Server does not require QuarkXPress Server Manager, but a QuarkXPress Server Manager server can coordinate multiple QuarkXPress Server applications so that they work together with maximum speed, reliability, and availability.

QuarkXPress Server Manager also provides a Web services interface that allows developers to use QuarkXPress Server features without having to use the HTTP interface.

To configure a QuarkXPress Server Manager server application, you must use the QuarkXPress Server Manager Web client. The topics below explain how the QuarkXPress Server Manager Web client works and provide examples for using it.

Understanding QuarkXPress Server Manager

Before you begin, take time to review the topics below so that you understand how this chapter is structured and how you can get the most out of it.

Load balancing

Load balancing ensures that each rendering request is sent to a server that is likely to be able to handle it quickly. QuarkXPress Server Manager lets you choose from three load-balancing settings:

- **Dynamic Load Balancer:** The QuarkXPress Server Manager server considers file size and throughput requirements for each request. For example, assume the following series of requests is sent to QuarkXPress Server Manager in an environment that uses two QuarkXPress Server instances:

Request	Size
1	8MB
2	1MB
3	2MB

- **Random Load Balancer:** Each rendering request is sent to a random server.
- **Round-robin Load Balancer:** Requests are sent to servers in a set order. For example, if you have three QuarkXPress Server instances and QuarkXPress Server Manager receives ten rendering requests, the requests are distributed as follows:

Request	QuarkXPress Server instance used
1	1
2	2
3	3
4	1
5	2
6	3
7	1

The first request is assigned to server #1, and the second request is assigned to server #2. When the third request arrives, QuarkXPress Server Manager checks the loads that the two servers are already handling and assigns the task to the server with the smallest load — in this case, server #2.

If a request fails because a server stops responding or because of a "File Not Found" error, QuarkXPress Server Manager does not resubmit that request to that server.

- ➔ The **Dynamic** setting is typically the most efficient setting for environments with more than one QuarkXPress Server instance.

For information about choosing a load-balancing setting, see "[Controlling load balancing](#)." Developers can implement their own load-balancing systems; for more information, see the QuarkXPress Server *Web Integration Guide*.

Request timeout interval

QuarkXPress Server Manager attempts to send each request to a QuarkXPress Server instance that can promptly handle that request. However, in some situations a QuarkXPress Server instance might be unable to process a request in a reasonable amount of time (for example, if the server is working on a large rendering job, or if the server computer has stopped functioning). If you specify a certain period of time as the request timeout interval,

QuarkXPress Server Manager will wait for the response until that period of time elapses, and then send the request to a different QuarkXPress Server instance. This ensures that a request does not get "lost" if its assigned QuarkXPress Server instance does not become available promptly.

For information about setting a request timeout interval, see "[Using other global settings](#)."

- ➡ If a client request fails despite being sent to multiple QuarkXPress Server instances, QuarkXPress Server sends the end user a customizable error message or exception so that the end user can appropriately handle the failure. A QuarkXPress Server Manager server can also automatically send e-mail to an administrator in the event of a problem; for more information, see "[Generating automatic e-mail messages](#)."

Determining QuarkXPress Server instance availability

QuarkXPress Server Manager uses two methods to determine the availability of a QuarkXPress Server instance: *ping* and *ping document*.

Ping

QuarkXPress Server Manager periodically sends a ping request to all of its QuarkXPress Server instances to determine whether they are available. Ping requests use the following format:

```
http://[Server]:[Port]/getprocessid
```

Ping document

QuarkXPress Server Manager periodically sends a render request to all of its QuarkXPress Server instances to determine whether they can render a document. Ping document requests use the following format:

```
http://[Server]:[Port]/[PingDocumentName.qxp]
```

You can specify the document to be used for this render request. To avoid long ping document rendering times, use a simple document.

You can specify the interval between pings and ping documents in the **Other Settings** pane in the **Global Settings** pane of the **QuarkXPress Server Manager** window (see "[Using other global settings](#)").

Logging

QuarkXPress Server Manager logs all interactions with QuarkXPress Server instances. QuarkXPress Server Manager log files contain the following information:

- Render requests
- QuarkXPress Server responses
- Information about events (such as alerts) that occur during the render-request process
- Details about requests that were sent to a different QuarkXPress Server instance after the first assigned QuarkXPress Server instance was unable to process the request

Within the logs, each QuarkXPress Server instance is identified by its IP address and port number.

You can export log files in XML (Extensible Markup Language) or comma-separated values (CSV) format. For more information, see "[Exporting log files](#)."

Caching

To increase speed and efficiency, QuarkXPress Server Manager caches information in memory. If the response to a render request, URL request, or file request is included in the QuarkXPress Server Manager memory cache, QuarkXPress Server Manager returns the response from the disk cache instead of sending the request to a QuarkXPress Server instance. For more information, see "[Managing the cache](#)"

- ➡ Requests that contain a binary parameter and multipart responses are not cached, regardless of whether global caching or command-specific caching is enabled.
- ➡ When the QuarkXPress Server Manager server application receives response data from a QuarkXPress Server instance, QuarkXPress Server Manager can return that response directly or write it as a file and return the file's URL. The second approach maximizes efficiency for SOAP-based clients, because SOAP transfers binary data very slowly. Cached response files have names that begin with "TMP_", and they are removed when they reach the age specified in the cache settings (see "[Managing the cache](#)"). QuarkXPress Server Manager uses the cache file this way regardless of whether caching is turned on or off; however, you can override this behavior by setting the `responseasurl` parameter to `false` for every request.

Web services

QuarkXPress Server Manager provides a Web services interface that makes it easy for developers to create applications that use QuarkXPress Server. This Web services interface provides the same functionality that is available through the QuarkXPress Server HTTP interface. For more information about the Web services interface, see the QuarkXPress Server *Web Integration Guide*.

Working with QuarkXPress Server Manager

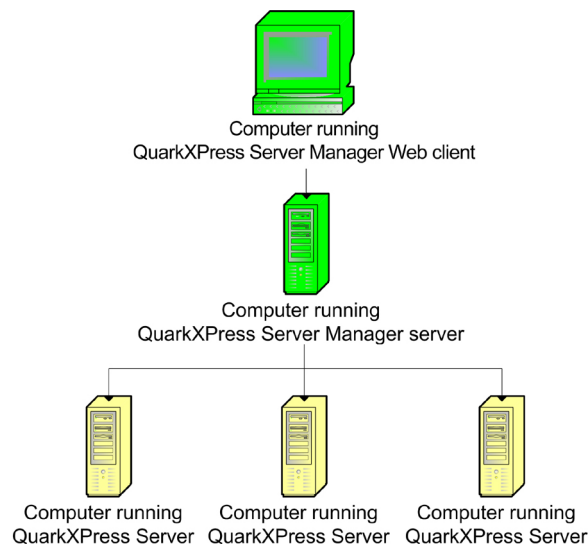
The general process for using QuarkXPress Server Manager is as follows:

- 1 Install QuarkXPress Server Manager software.
- 2 Launch one or more QuarkXPress Server instances on your network.
- 3 Launch the QuarkXPress Server Manager Console server application (see "[Starting the Manager server application](#)").
- 4 Launch the QuarkXPress Server Manager Web client (see "[Starting the Manager client application](#)").

- 5 Use the **Manage Servers** pane to add QuarkXPress Server instances, specify information about those servers (see "[Configuring QuarkXPress Server instances](#)"), and choose a load-balancing method (see "[Controlling load balancing](#)").
- 6 Configure proxy server settings, automatic e-mail settings, and various other settings in the **Global Settings** pane (see "[Using a proxy server](#)," "[Generating automatic e-mail messages](#)," and "[Using other global settings](#)").
- 7 As necessary, delete cache items and clear the QuarkXPress Server Manager server cache using the **Manage Cache** pane (see "[Managing the cache](#)").

Starting QuarkXPress Server Manager

Each QuarkXPress Server Manager server can handle multiple QuarkXPress Server instances.



QuarkXPress Server Manager diagram

Starting the Manager server application

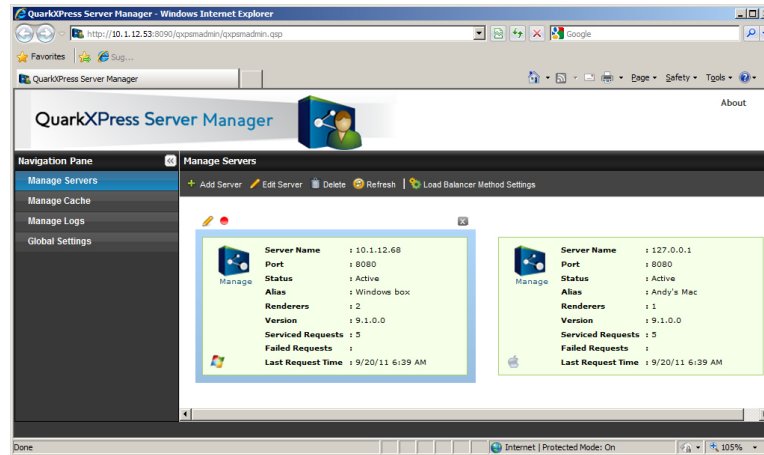
To launch the QuarkXPress Server Manager server application:

- *Mac OS:* Open the **QuarkXPress Server Manager/Server** folder inside the applications folder and double-click "QXPSMServerStart.command."
- *Windows (if you have not installed QuarkXPress Server Manager as a service):* Choose **Start > Programs > QuarkXPress Server Manager 9 > Start QuarkXPress Server Manager**. Alternatively, open the "Server" folder in the QuarkXPress Server Manager application folder and double-click the "QXPSMServerStart.bat" file as an administrator.

You can access API documentation in HTML format by navigating to [http://\[server\]:\[port\]](http://[server]:[port]), where **[server]** identifies the computer on which QuarkXPress Server Manager is running and **[port]** identifies the port it is running on.

Starting the Manager Web client

To launch the QuarkXPress Server Manager Web application, click **Open Admin Client** on the QuarkXPress Server Manager home page. The QuarkXPress Server Manager administration Web client displays.



The QuarkXPress Server Manager Web client

Request handler binding

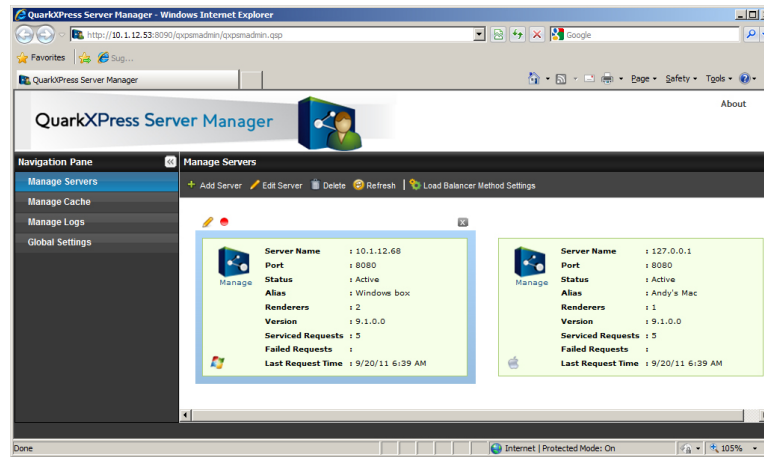
When you add a QuarkXPress Server instance to QuarkXPress Server Manager, you can choose to bind that server instance to particular rendering type or request type. When QuarkXPress Server Manager receives a matching request, it will send the request to only those server instances that are bound to that rendering type or request type. If multiple server instances are bound to a particular rendering type or request type, QuarkXPress Server Manager balances the load of such requests between the server instances.

An instance of QuarkXPress Server Manager that is not bound to any specific type of request is called a *generic server*. If a request is not bound to a particular server instance, QuarkXPress Server Manager sends that request to a generic server. If more than one generic server is available, QuarkXPress Server Manager balances the load of such requests between them.

You can bind a server to more than one rendering type or request type.

Configuring QuarkXPress Server instances

The **Manage Servers** pane lists the QuarkXPress Server instances the QuarkXPress Server Manager server is handling. You can use this pane to add QuarkXPress Server instances, edit the description of existing QuarkXPress Server instances, delete QuarkXPress Server instances, and choose a load balancing method.



Manage Servers pane

- ➔ To configure an individual QuarkXPress Server instance, click **Manage** under the icon for that instance. The QuarkXPress Server Web interface for that instance displays.

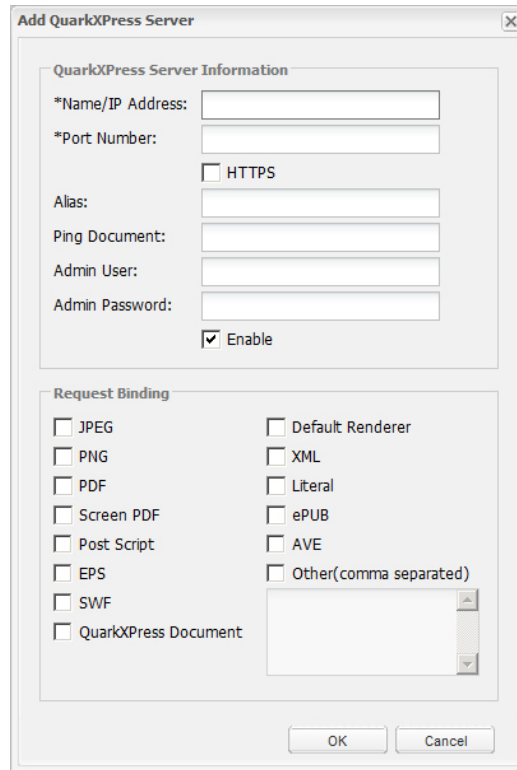
Adding and enabling a QuarkXPress Server instance

When you add and enable a QuarkXPress Server instance in the **Manage Servers** pane of the **QuarkXPress Server Manager** interface, the QuarkXPress Server Manager server begins routing rendering requests to that QuarkXPress Server instance.

- ➔ If you add and enable a QuarkXPress Server instance in this pane, be sure that clients are no longer sending rendering requests directly to that QuarkXPress Server instance; otherwise, the server will be handling both direct requests and routed requests, and the server might become overly busy. Note also that requests sent directly to such a QuarkXPress Server instance do not benefit from QuarkXPress Server Manager features such as load balancing, caching, and logging.

To add and enable a QuarkXPress Server instance:

- 1 Display the **Manage Servers** pane of the **QuarkXPress Server Manager** interface.
- 2 Click **Add Server**. The **Add QuarkXPress Server** dialog box displays.



The dialog box is titled "Add QuarkXPress Server". It contains two main sections: "QuarkXPress Server Information" and "Request Binding".

QuarkXPress Server Information:

- *Name/IP Address: [Text Field]
- *Port Number: [Text Field]
- ☐ HTTPS
- Alias: [Text Field]
- Ping Document: [Text Field]
- Admin User: [Text Field]
- Admin Password: [Text Field]
- ☒ Enable

Request Binding:

- ☐ JPEG
- ☐ PNG
- ☐ PDF
- ☐ Screen PDF
- ☐ Post Script
- ☐ EPS
- ☐ SWF
- ☐ QuarkXPress Document
- ☐ Default Renderer
- ☐ XML
- ☐ Literal
- ☐ ePUB
- ☐ AVE
- ☐ Other(comma separated) [List Box]

At the bottom are "OK" and "Cancel" buttons.


Add QuarkXPress Server dialog box

- 3 Enter the QuarkXPress Server instance's DNS name or IP address in the **Name/IP Address** field.
- 4 Enter the QuarkXPress Server instance's port number in the **Port Number** field.
- 5 If the QuarkXPress Server instance is running with the HTTPS protocol, check **HTTPS**.
- 6 To specify an alternate name for the server, enter a value in the **Alias** field. The **Alias** value displays in the **Manage Servers** pane of the **QuarkXPress Server Manager** interface.
- 7 If you choose to use a particular ping document for this server (see "[Ping document](#)"), make sure the project file is in the QuarkXPress Server instance's document pool and then enter the project's file name in the **Ping Document** field. The ping document is used only if the global **Ping Type** is set to **Ping Document** (see "[Using other global settings](#)"). Note that if you do not set a ping document here, and no global ping document is set (see "[Using other global settings](#)"), an error message might display to indicate that the server is registered but inactive.
- 8 Enter the QuarkXPress Server instance user name and password in the **Admin User** and **Admin Password** fields.
- 9 To specify that QuarkXPress Server Manager should begin sending rendering requests to this QuarkXPress Server instance, check **Enable**.
- 10 To restrict this server to one or more particular types of rendering, check the appropriate boxes in the **Request Binding** area. To add additional render types (for example, render types provided by QuarkXPress Server XTensions software), check **Other** and enter the

appropriate namespaces in the corresponding field as a comma-separated list. For more information, see "[Request handler binding](#)."

11 Click OK.

Editing a QuarkXPress Server instance



To edit the description of a QuarkXPress Server instance, display the **Manage Servers** pane of the **QuarkXPress Server Manager** interface, select the server in the list, and then click **Edit Server**. You can also display the **Edit QuarkXPress Server** dialog box by hovering the mouse cursor over the server and then clicking the **Edit Server** button  on the upper left.

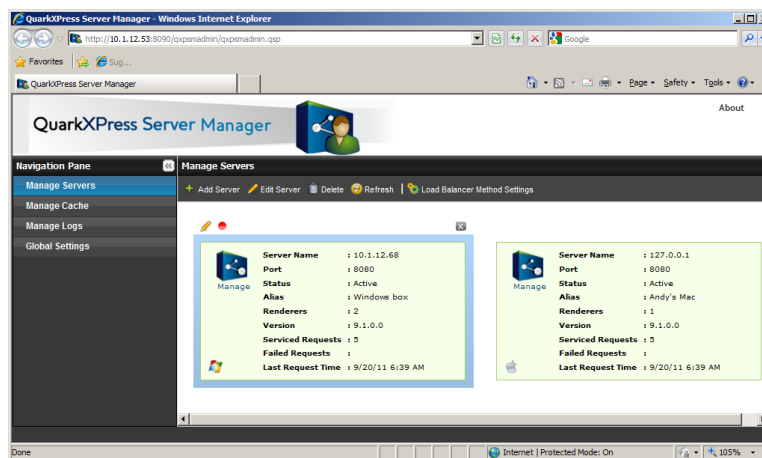
The options in the **Edit QuarkXPress Server** dialog box are the same as the options in the **Add QuarkXPress Server** dialog box (see "[Adding and enabling a QuarkXPress Server instance](#)").

Enabling and disabling routing to QuarkXPress Server instances

The **Status** field for each server in the **Manage Servers** pane of the **QuarkXPress Server Manager** interface shows the status of each QuarkXPress Server instance.

➔ The status fields are not updated automatically. To update the **Status** field for all servers, click **Refresh**.

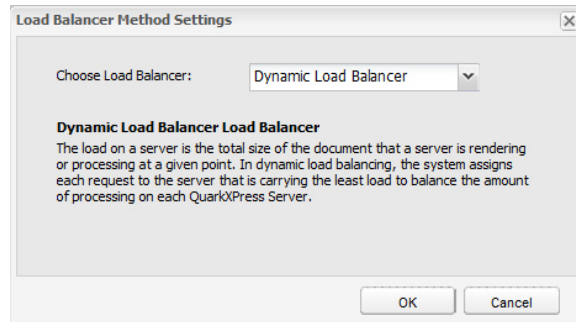
To enable or disable routing to a QuarkXPress Server instance, select the server, click **Edit Server** to display the **Edit QuarkXPress Server** dialog box, check or uncheck **Enable**, and then click **OK**. You can also enable or disable a QuarkXPress Server instance by hovering the mouse cursor over the server and then clicking the **Enable the Server** button  or **Disable the Server** button  on the upper left.



Manage Servers pane

Controlling load balancing

Load balancing ensures that each rendering request is sent to a QuarkXPress Server instance that is most likely to be able to handle it quickly. To define a load-balancing setting for the QuarkXPress Server Manager server, display the **Manage Servers** pane of the **QuarkXPress Server Manager** interface and click **Load Balancer Method Settings**. The **Load Balancer Method Settings** dialog box displays.




Load Balancer Method Settings dialog box

QuarkXPress Server Manager lets you choose from three load-balancing settings:

- **Dynamic Load Balancer:** Sends requests to servers based on file size
- **Random Load Balancer:** Sends each rendering request to a random server
- **Round-robin Load Balancer:** Sends requests to servers in a set order

For more information about these load-balancing settings, see "[Load balancing](#)"

Deleting a QuarkXPress Server instance

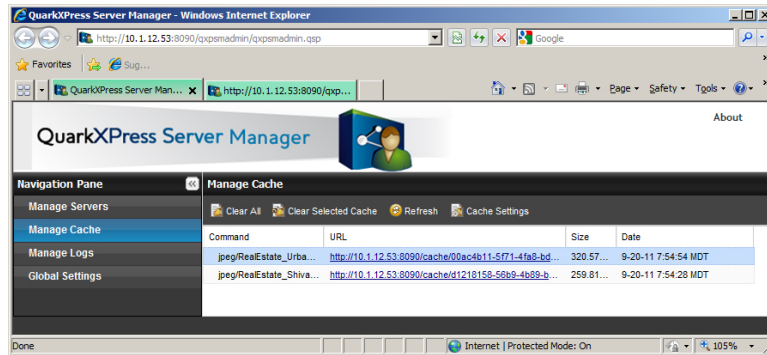
To delete a QuarkXPress Server instance from the list of available servers in the **QuarkXPress Server Manager** interface, display the **Manage Servers** pane, select the server name or IP address, and then click **Delete**. You can also delete a server by hovering the mouse cursor over the server and then clicking the **Delete the Server** button  on the upper left.

- ➔ Deleting a QuarkXPress Server instance from this dialog box does not shut down the QuarkXPress Server instance, but it does prevent the QuarkXPress Server Manager server from routing rendering requests to the QuarkXPress Server instance.

Managing the cache

Each QuarkXPress Server Manager server has an in-memory cache (in which it stores the keys to recently accessed items) and a disk-based cache (in which the items themselves are stored). If a request for a recently used item arrives, and a QuarkXPress Server Manager server has that request in its memory cache, the server can simply return the response from its disk cache instead of having to send the request to a QuarkXPress Server instance.

To manage the QuarkXPress Server Manager cache, display the **Manage Cache** pane of the **QuarkXPress Server Manager** interface.



Manage Cache pane

- ➔ Requests are stored in the cache only if the cache is turned on. For more information, see "[Configuring cache options](#)."
- ➔ The cache stores only the results of requests that do not deliberately bypass the cache.

Viewing a QuarkXPress Server Manager server cache

To view QuarkXPress Server Manager cache information, display the **Manage Cache** pane of the **QuarkXPress Server Manager** interface. For each file in the cache, this pane lists the command, URL, size, and time and date it was generated. To view a cache file, double-click the file name in the list.

Deleting files from the cache

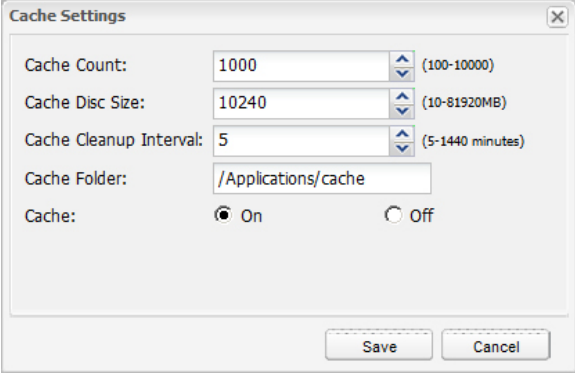
There is usually no need to manually delete files from a QuarkXPress Server Manager server's cache. When the cache reaches 95% of its capacity, QuarkXPress Server Manager automatically begins deleting the least recently used files in the cache to make room for new files. However, you can also manually clear files from the cache.

To manually delete cache files:

- 1 If you want to delete specific files, select those files in the list.
- 2 Click **Clear Selected Cache**. The **Clear Cache** alert displays.
- 3 Click **OK**.

Configuring cache options

To configure cache options, display the **Manage Cache** pane in the **QuarkXPress Server Manager** interface, then click **Cache Settings**. The **Cache Settings** dialog box displays.

A screenshot of the 'Cache Settings' dialog box. It contains five fields: 'Cache Count' with a value of 1000 and a range of (100-10000); 'Cache Disc Size' with a value of 10240 and a range of (10-81920MB); 'Cache Cleanup Interval' with a value of 5 and a range of (5-1440 minutes); 'Cache Folder' with a text field containing '/Applications/cache'; and 'Cache' with two radio buttons, 'On' (selected) and 'Off'. At the bottom are 'Save' and 'Cancel' buttons.

Cache Count:	1000	(100-10000)
Cache Disc Size:	10240	(10-81920MB)
Cache Cleanup Interval:	5	(5-1440 minutes)
Cache Folder:	/Applications/cache	
Cache:	<input checked="" type="radio"/> On <input type="radio"/> Off	

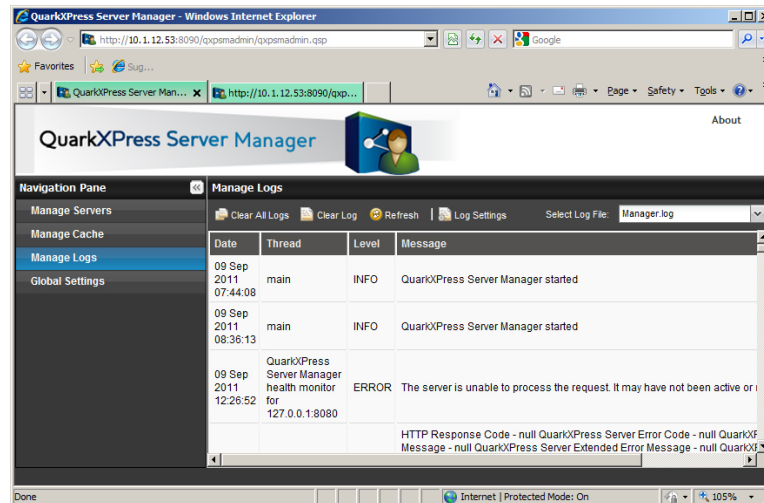
Manage Cache pane

- To set the maximum number of files allowed in the cache, enter a value in the **Cache Count** field. When the number of files in the cache reaches the number you set here, QuarkXPress Server Manager begins deleting the least recently used files to make room in the cache.
- To set the maximum disk cache size, enter a value in the **Cache Disk Size** field. When the disk cache reaches this size, QuarkXPress Server Manager begins deleting the least recently used files to make room in the cache.
- To specify an interval after which the cache should be periodically cleared, enter a value in the **Cache Cleanup Interval** field.
- To specify where cache files for the QuarkXPress Server Manager server should be stored, enter a path in the **Cache Folder** field.
- The **Cache** radio buttons let you control caching for the QuarkXPress Server Manager server. To turn caching on, click **On**. To turn caching off, click **Off**.

Managing logs

A QuarkXPress Server Manager server maintains logs of all of the requests it receives, the responses from the QuarkXPress Server instances, information about events (such as alerts) that occur during the render-request process, dates and times, and details about each request that was sent to a different QuarkXPress Server instance after its first assigned QuarkXPress Server instance was unable to process the request.

- ➔ To control what information is stored in the logs, use the **Manage Logs** pane of the **QuarkXPress Server Manager** interface. For more information, see "[Configuring logging options](#)."



Manage Logs pane

Viewing log file details

To view information about a specific log file, display the **Manage Logs** pane of the **QuarkXPress Server Manager** interface, then choose the log file name from the **Select Log File** drop-down menu.

Deleting log files

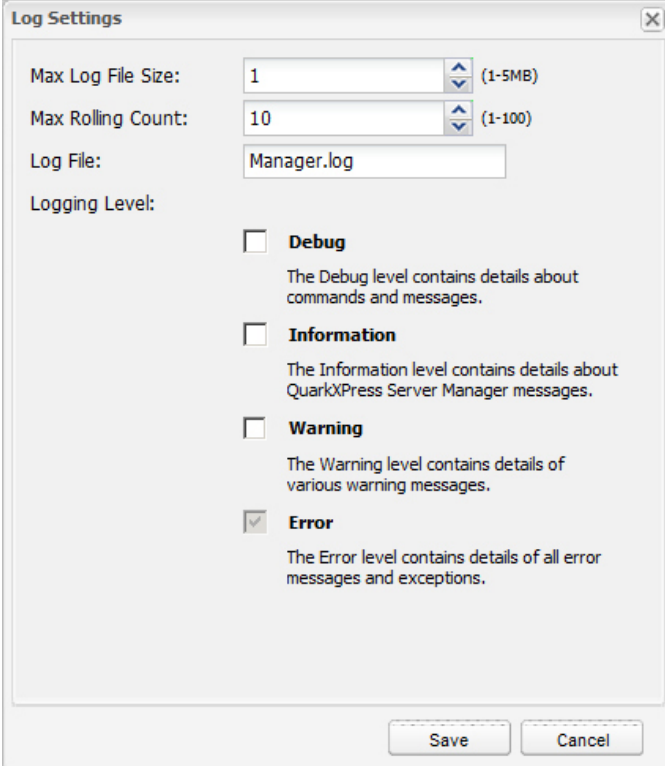
To delete a log file:

- 1 Display the **Manage Logs** pane of the **QuarkXPress Server Manager** interface.
- 2 Choose the target log file from the **Select Log File** drop-down menu.
- 3 Click **Clear Log**.

➔ To clear all log files, click **Clear All Logs**.

Configuring logging options

To configure logging options, click **Log Settings** pane in the **Manage Logs** pane of the **QuarkXPress Server Manager** interface. The **Log Settings** dialog box displays.

The image shows a 'Log Settings' dialog box with a title bar and a close button. It contains four input fields: 'Max Log File Size' with a value of 1 and a range of (1-5MB); 'Max Rolling Count' with a value of 10 and a range of (1-100); 'Log File' with the text 'Manager.log'; and 'Logging Level' with four radio button options: 'Debug', 'Information', 'Warning', and 'Error'. The 'Error' option is selected. Each option has a descriptive text below it. At the bottom right are 'Save' and 'Cancel' buttons.

Log Settings

Max Log File Size: 1 (1-5MB)

Max Rolling Count: 10 (1-100)

Log File: Manager.log

Logging Level:

- ☐ **Debug**
The Debug level contains details about commands and messages.
- ☐ **Information**
The Information level contains details about QuarkXPress Server Manager messages.
- ☐ **Warning**
The Warning level contains details of various warning messages.
- ☒ **Error**
The Error level contains details of all error messages and exceptions.

Save Cancel

Log Settings dialog box

To set the maximum log file size, enter a value in the **Max Log File Size** field. When a log file reaches this size, the current log file is closed and a new log file is created.

To specify the maximum number of log files to keep, enter a value in the **Max Rolling Count** field. When the number of log files reaches this limit, QuarkXPress Server Manager deletes the oldest log file each time a new log file is created.

To specify the root name of the log file for the QuarkXPress Server Manager server, enter that name in the **Log File** field. To place the log file in a particular directory, precede the file name with an absolute path. QuarkXPress Server Manager appends numbers to this name to create consecutively named log files.

To control how much information is stored in the log files, check a box in the **Logging Level** area:

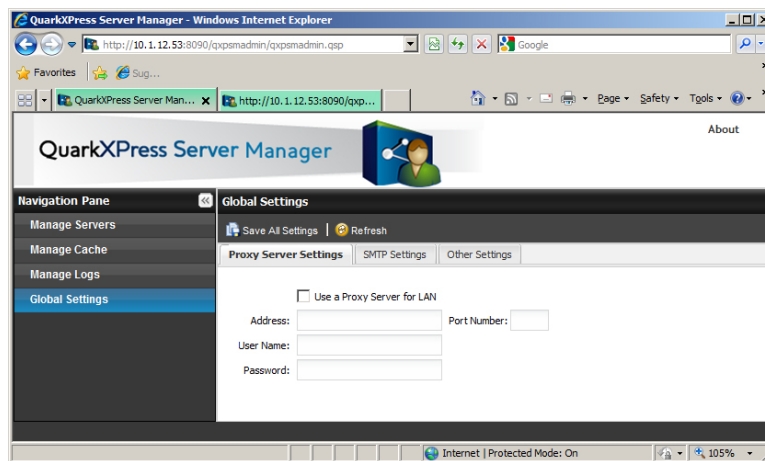
- **Debug:** Stores information such as the commands executed and the servers where those commands are executed. Also stores all of the information that is stored when **Information**, **Warning**, and **Error** are checked.
- **Information:** Stores informational messages such as startup messages and command-retry messages. Also stores all of the information that is stored when **Warning** and **Error** are checked.
- **Warning:** Stores warning messages. Also stores all of the information that is stored when **Error** is checked.
- **Error:** Stores error messages and stack traces for exceptions.

- ➔ The **Debug** and **Information** settings produce large logs that grow rapidly, so you might want to use these settings for troubleshooting only.

Using a proxy server

Some networks route network traffic through a proxy server for reasons of efficiency or security. To use a proxy server for all requests and responses between QuarkXPress Server Manager and QuarkXPress Server:

- 1 Display the **Proxy Server Settings** tab of the **Global Settings** pane of the **QuarkXPress Server Manager** interface.



Proxy Server Settings tab

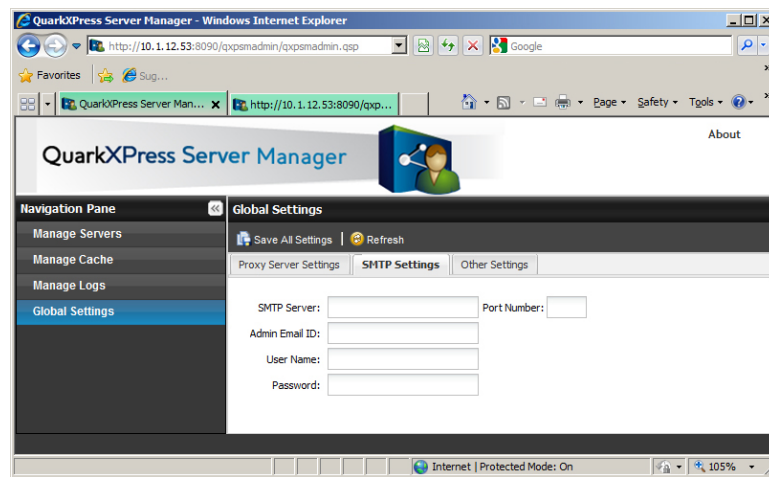
- 2 Check **Use a Proxy Server for LAN**.
- 3 Enter the proxy server's DNS name or IP address in the **Address** field.
- 4 Enter the proxy server's port number in the **Port Number** field.
- 5 Enter the proxy server's user name in the **User Name** field.
- 6 Enter the proxy server's password in the **Password** field.

Generating automatic e-mail messages

You can configure a QuarkXPress Server Manager server to automatically generate and send e-mail messages if particular events occur. To configure the QuarkXPress Server Manager server to automatically send e-mail messages:

- 1 Display the **SMTP Settings** tab of the **Global Settings** pane of the **QuarkXPress Server Manager** interface.
- 2 Enter a valid SMTP server name or IP address in the **SMTP Server** field and then enter the corresponding port number in the **Port Number** field.
- 3 Enter the e-mail address to which messages should be sent in the **Admin Email ID** field.

- 4 If this SMTP server requires validation, enter a valid user name in the **User Name** field and a valid password in the **Password** field.



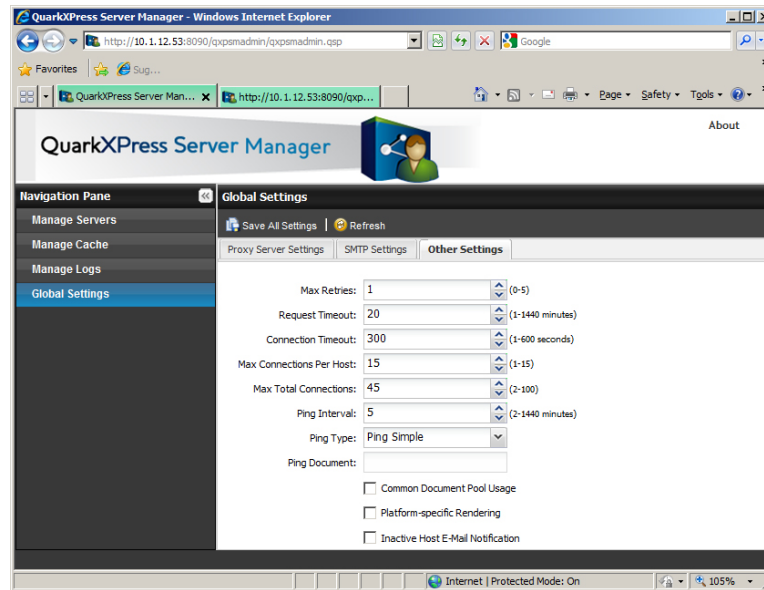
SMTP Settings pane

Two events can cause QuarkXPress Server Manager to generate an e-mail message:

- If **Inactive Host E-mail Notification** is checked (see "[Using other global settings](#)"), a message is sent when a QuarkXPress Server instance goes from the active state to the inactive state.
- If you have set up custom error messages (as described in "[Using custom error messages](#)"), certain QuarkXPress Server errors result in e-mail messages.

Using other global settings

To configure other global settings, display the **Other Settings** tab of the **Global Settings** pane of the **QuarkXPress Server Manager** user interface.



Other Settings tab of Global Settings page

- **Max Retries:** To specify the maximum number of times the QuarkXPress Server Manager server should submit a request to the QuarkXPress Server instances, enter a value in this field. When QuarkXPress Manager has unsuccessfully submitted a request this many times, the application returns an error message.
- **Request Timeout:** To specify the maximum number of minutes the QuarkXPress Server Manager server should wait for a response from a QuarkXPress Server instance, enter a value in this field. When this time has elapsed, the QuarkXPress Server Manager server retries the request (unless the **Max Retries** value has been reached).
- **Connection Timeout:** To specify the maximum number of seconds the QuarkXPress Server Manager server should spend attempting to establish a connection with a particular QuarkXPress Server instance, enter a value in this field.
- **Max Connections Per Host:** To specify the maximum number of connections the QuarkXPress Server Manager server should open for a particular QuarkXPress Server instance before it begins queuing requests, enter a value in this field.
- **Max Total Connections:** To specify the maximum number of connections the QuarkXPress Server Manager server should open for all hosts before it begins queuing requests, enter a value in this field.
- **Ping Interval:** To set the amount of time the QuarkXPress Server Manager server should wait between ping attempts (see "[Determining QuarkXPress Server instance availability](#)"), enter a value in this field.
- **Ping Type:** To indicate whether QuarkXPress Server Manager should use a simple ping or a ping document to test a QuarkXPress Server instance, choose an option from this drop-down menu.
- **Ping Document:** To indicate which QuarkXPress project the QuarkXPress Server Manager server should use for ping document requests, enter the project file name in this field.

Make sure a copy of the project file is in each QuarkXPress Server instance's document pool. Note that individual QuarkXPress Server instances can override this value by providing another document name. Note also that a ping document is used only if **Ping Type** is set to **Ping Document**.

- **Common Document Pool Usage:** Check this box if all managed QuarkXPress Server instances are using the same document pool. If this box is checked, then upload, delete, and saveas requests are sent to one of the available servers. If this box is unchecked, then upload, delete, and saveas requests are sent to all managed servers. Note that you must manually set each server to point at the common document pool.
- **Platform-specific Rendering:** A QuarkXPress Server Manager server can send requests that involve Mac OS project files to a Mac OS-based QuarkXPress Server instance, and send requests that involve Windows project files to a Windows-based QuarkXPress Server instance. Setting up the server in this manner can be desirable if the project files involved use fonts that are available on only one platform or the other. To enable platform-specific rendering for the QuarkXPress Server Manager server, check this box.
- **Inactive Host E-mail Notification:** To automatically generate an e-mail message when a QuarkXPress Server instance becomes inactive, check this box. The e-mail message is sent to the address specified in the **SMTP Settings** pane.

Saving a server configuration

Changes that you make to a server configuration in the QuarkXPress Server Manager client are not made to the QuarkXPress Server Manager server until you click **Save All Settings** in the **Global Settings** pane of the QuarkXPress Server Manager interface.

To discard any changes you have made since logging on to the QuarkXPress Server Manager server, click **Refresh**. The configuration of the QuarkXPress Server Manager server remains as it was.

- ➡ Disabling or enabling a QuarkXPress Server instance from the QuarkXPress Server Manager client is not considered a configuration change.

Using custom error messages

You can control which errors cause the QuarkXPress Server Manager server application to send e-mail messages to the address specified in the **SMTP Settings** pane (see "[Generating automatic e-mail messages](#)"). You can also define which messages are sent when such errors occur. The first step is to create a custom error code that corresponds to a QuarkXPress Server error code. After you set up this custom error code, you can specify whether that code generates an e-mail message and then create custom error messages.

Creating a custom error code

To create a custom error code:

- 1 On the computer where the QuarkXPress Server Manager server application is running, open the following file in a text-editing application:

```
[application folder]\server\conf\Manager_Server_ErrorCodeMapping.properties
```

- 2 Create a new line containing a QuarkXPress Server specific error code for which you want to generate automatic e-mail messages (with or without a custom text message). Follow the error code with an equals sign, a unique custom error code, and a return.
- 3 Save and close the file.

➡ This change will not take effect until you quit and restart the QuarkXPress Server Manager server application.

Flagging an error code to generate an e-mail message

To specify that a custom error code should cause an e-mail message to be generated:

- 1 Create a unique custom error code for the target QuarkXPress Server error. For more information, see "[Creating a custom error code](#)".
- 2 On the computer where the QuarkXPress Server Manager server application is running, open the following file in a text-editing application:

```
[application folder]\server\conf\ManagerErrorCodeMailOption.properties
```

- 3 Create a new line containing the unique custom error code you defined in step 2. Follow the custom error code with a tab, enter a 1 (to send the message) or a 0 (to suppress the message), and then press Return.
- 4 Save and close the file.

➡ This change will not take effect until you quit and restart the QuarkXPress Server Manager server application.

Creating custom error text

To define the text that should be sent in an e-mail message when an error occurs:

- 1 Create a unique custom error code for the target QuarkXPress Server error (as described in "[Creating a custom error code](#)").
- 2 On the computer where the QuarkXPress Server Manager server application is running, open the following file in a text-editing application:

```
[application folder]\server\conf\ManagerErrorCodeMessage.properties
```

- 3 Create a new line containing the unique custom error code you defined in step 2. Follow the custom error code with a tab and then enter the custom text to be returned for that error.
- 4 Save and close the file.

- ➡ This change will not take effect until you quit and restart the QuarkXPress Server Manager server application.

Sending requests from a browser

Like QuarkXPress Server, QuarkXPress Server Manager lets you send requests from a Web browser. This capability helps to ensure that you need to make only minimal changes when you update an application so that it sends requests to a QuarkXPress Server Manager server instead of a QuarkXPress Server instance.

Assume that a QuarkXPress Server instance expects requests in the following format:

```
http://[QXPServer]:[port]/[request]?[request_parameters]
```

If this is the case, a QuarkXPress Server Manager server will expect requests in the following format:

```
http://[QXPManagerServer]:[port]/qxpsm/request/[request]?[request_parameters]
```

In other words, a QuarkXPress Server Manager server accepts requests in a format that is similar to the request format used with a QuarkXPress Server instance. That means you can get the benefits of QuarkXPress Server Manager without having to completely rewrite your applications.

Additional parameters

In addition to request-specific parameters, QuarkXPress Server Manager accepts the following request parameters. These parameters can be submitted in the standard QuarkXPress Server GET format joined by an ampersand (&) with the other parameters in the body of the request.

- **qxpsm_bypassfileinfo**: When QuarkXPress Server Manager receives a request, it executes a **fileinfo** request on the document to get the document's size and last-modified date and time. These values are used for load balancing and for determining whether to serve the document from the cache. If the value of this parameter is set to true, the **fileinfo** request is not made, file size is considered to be zero for load-balancing purposes, and the document in the cache is considered to have changed.
- **qxpsm_context**: Set this value if you need to use it in a custom load balancer.
- **qxpsm_maxtries**: This parameter lets you specify a maximum number of retries for a specific request. If this parameter is absent or set to 0, the global maximum retries value is used. To disable maximum retries for a request, use the value -1.
- **qxpsm_password**: The value of this parameter, if supplied, is used as the "Admin Password" value when the request is forwarded to a QuarkXPress Server instance.
- **qxpsm_responseasurl**: By default, QuarkXPress Server Manager writes responses as temporary files in the cache folder and returns a URL to the client. This approach prevents the performance degradation that can result from sending binary data using SOAP. However, you might want QuarkXPress Server Manager to return the response directly if you are creating an application that processes that response (simple or multipart) with its own

logic. To make QuarkXPress Server Manager send a response to the browser rather than the URL of the temporary files in the cache, set this value to false. (Note that setting this value to false might result in decreased performance.)

- **qxpsm_responseredirect**: If you use servlet methodology to send a request to QuarkXPress Server Manager with **qxpsm_responseasurl=true**, QuarkXPress Server Manager returns a frameset page with one or more frames. If the response is not multipart, the frameset contains a single page with a URL pointing to a response file in the temporary cache. This can be problematic if, for example, you want to use the returned URL as an image link in an HTML page. In such situations, submit the request with both **qxpsm_responseredirect=true** and **qxpsm_responseasurl=true**. If you do this, QuarkXPress Server Manager returns the URL of the rendered file in the temporary cache instead of returning a frameset page. Note, however, that if the request results in a multipart response (such as the response returned by the **boxes** parameter), QuarkXPress Server Manager ignores the **qxpsm_responseredirect=true** parameter and returns the frameset page.
- **qxpsm_servername**: By default, the target QuarkXPress Server instance for each request is determined by the QuarkXPress Server Manager server's load-balancing system. To send a request to a specific QuarkXPress Server instance, set this parameter to the name or IP address of that QuarkXPress Server instance. Note that if you use this parameter with an IP address, you must also submit the port number using the **qxpsm_serverport** parameter.
- **qxpsm_serverport**: If you use the **qxpsm_servername** parameter with an IP address, supply the corresponding port number as this parameter's value.
- **qxpsm_timeout**: This parameter lets you specify a timeout (in milliseconds) for a specific request. If this parameter is absent or set to 0, the global timeout value is used. To disable timeout for a request, use the value -1.
- **qxpsm_usecache**: If you set this value to false, the request will be rendered regardless of whether it is cached and regardless of whether caching is enabled at the global level.
- **qxpsm_username**: The value of this parameter, if supplied, is used as the "Admin User" value when the request is forwarded to a QuarkXPress Server instance.
- **qxpsm_userpassword**: The value of this parameter, if supplied, is used as the "Admin User" password when the request is forwarded to a QuarkXPress Server instance. Default value is null, which means no password information. If this is not null, you must also provide **qxpsm_username**.

Glossary

Document pool: The document pool contains the projects that are available for rendering. By default, the document pool is a collection of discrete files or folders in a specific, identified folder located on the local server or on a connected network drive. When some type of external document provider (such as a content management system or database) is used, projects are not stored in the local document pool.

Document provider: The document provider is the source for projects that QuarkXPress Server renders. The most basic document provider is the local document pool. Other document providers can be enabled through the creation of Server XTensions software, which establish a virtual file system. Server XTensions software can register for control of a specified range of the QuarkXPress Server namespace. When a project is requested from this range, server XTensions software retrieves the file from the specified source and hands it to the server. Examples of document providers include content management systems such as a standard database, or a live data feed from an HTTP agent.

Layout: A layout is a sequence of same-sized pages in a QuarkXPress project. A project can contain one or more layouts. A layout is functionally equivalent to a QuarkXPress document in QuarkXPress 5 and earlier.

Project: A QuarkXPress project is a file created by QuarkXPress. A project can contain one or more layouts.

Rendering: Rendering is the process of generating a file in a particular format (such as JPEG, EPS, or PDF) from a QuarkXPress layout.

Rendering type: The rendering type is the format in which QuarkXPress Server can render QuarkXPress layouts. Some rendering types, such as JPEG and PNG, can be displayed in a Web browser, while others must be saved to the hard drive.

Server XTensions Software (SXT): Server XTensions software is XTensions software written specifically for QuarkXPress Server. For more information, see the QuarkXPress Server XTensions Developer's Kit.

Renderer: A renderer is a process launched by QuarkXPress Server to help process rendering requests. Renderers reside on the same server as QuarkXPress Server and share the same memory and preferences. When renderers are launched, QuarkXPress Server becomes a load-balancing "master server," passing incoming requests to renderers for faster response times.

Legal notices

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

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

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

Index

A

App Studio 43
application preferences 28
articles 38, 39
AVE 43

B

binding 56

C

caching 10, 60, 61
color management 27
components 39
connection queue 11
construct 34
copydesk 33, 38, 39
custom error messages 68, 69

D

deconstruct 34
default render type 17
document pool 9, 16, 17, 66
document providers 9, 33

E

e-mail notifications 65, 66, 69
EPS 22
error messages 68, 69

F

Flash 42
fonts 21, 22
frames 23
Full Res Preview 28

H

H&Js 24
hyperlinks 23
hyphenation 24

I

Interactive Designer 42

J

Job Jackets 28

K

kerning 22, 24

L

layers 28, 42
layouts 9
leading 24
ligatures 24
line spacing 24
load balancing 60
logging 16, 62, 63

M

master pages 23
master-renderer environment 11
measurement units 23
memory 17
missing fonts 21, 22
Modifier 40, 41, 42

N

namespaces 31
navigation pane 15
nonetwork option 13

P

- parameters 32, 33, 70
- passwords 70
- PDF 22, 40
- pictures 28
- ping 66
- platform-specific rendering 66
- preferences 10, 20, 28
- projects 9
- proxy servers 65

Q

- qcddoc 33
- QLA 11
- Quark License Administrator 11
- QuarkCopyDesk 38, 39

R

- render type, default 17
- render types 32
- renderer 11
- renderers 13
- request handler binding 56
- routing 59

S

- server configuration 57, 68

- server templates 8
- ServerpApp.properties file 13
- small caps 24
- static projects 8
- status monitor 16
- streaming documents 9
- subscript 24
- superior 24
- superscript 24
- SWF 42

T

- Telegraph 44
- templates 48
- timeouts 66, 70
- trapping 26

U

- Unicode 21
- URL requests 30, 32, 70
- user names 70

X

- XML 34, 40, 41, 42
- XTensions manager 16
- XTensions software 12, 38
- XTensions software API 12