

# Getting Started with QuarkXPress Server 2018 (v14.3)

2017  
**QUARKXPRESS  
SERVER**



# Contents

<b>Introduction.....</b>	<b>1</b>
<b>Related Documents .....</b>	<b>2</b>
<b>System Requirements.....</b>	<b>3</b>
System requirements: macOS .....	3
System requirements: Windows .....	4
<b>Installing QuarkXPress Server.....</b>	<b>5</b>
Installing QuarkXPress Server: macOS .....	5
Installing QuarkXPress Server: Windows.....	5
<b>Upgrading QuarkXPress Server .....</b>	<b>7</b>
<b>Deploying QuarkXPress Server Manager .....</b>	<b>8</b>
<b>Installing QuarkXPress Server Manager .....</b>	<b>8</b>
Installing QuarkXPress Server Manager: macOS .....	9
Installing QuarkXPress Server Manager: Windows .....	9
<b>Deploying QXPSM in a shared Spring context .....</b>	<b>9</b>
<b>Using Amazon S3 as a document pool in QuarkXPress Server ..</b>	<b>12</b>
<b>Create an S3 bucket in AWS .....</b>	<b>12</b>
<b>Create a policy .....</b>	<b>13</b>
<b>Create a role .....</b>	<b>15</b>
<b>Assign role to EC2 instance .....</b>	<b>15</b>
<b>Register Amazon S3 as a Docpool in QuarkXPress Server .....</b>	<b>16</b>
<b>Deploying in an external Web container .....</b>	<b>17</b>
<b>Deploying QXP Server in external Tomcat .....</b>	<b>17</b>
<b>Deploying QXPSM in external Tomcat.....</b>	<b>18</b>
<b>Deploying QXP Server in WebSphere .....</b>	<b>19</b>
Prerequisites.....	19
Installing the IBM-WebSphere developer version .....	19
Configure the Java 7 SDK in the IBM WebSphere .....	20
Start IBM WebSphere .....	20
QXPS WebApp deployment into IBM WebSphere UI .....	21

<b>Introduction to QLA Server.....</b>	<b>23</b>
<b>Uninstalling.....</b>	<b>24</b>
Uninstalling: macOS .....	24
Uninstalling: Windows.....	24
<b>Changes in this version .....</b>	<b>25</b>
New for QuarkXPress Server 2018 - September 2019 Update .....	25
<b>Changes in previous versions .....</b>	<b>26</b>
New for QuarkXPress Server 2018 - February 2019 Update .....	26
New for QuarkXPress Server 2018.....	26
New for QuarkXPress Server 2017 October Update .....	27
New for QuarkXPress Server 2017 March 2018 Update .....	27
New for QuarkXPress Server 2017.....	28
New for QuarkXPress Server 2016.....	29
New for QuarkXPress Server 2015 - April 2016 Update .....	29
New for QuarkXPress Server 2015 - February 2016 Update .....	30
New for QuarkXPress Server 2015 - October Update .....	30
New for QuarkXPress Server 2015.....	32
New for QuarkXPress Server10.5.2.....	32
New for QuarkXPress Server10.5.1.....	32
New for QuarkXPress Server10.5.....	32
New for QuarkXPress Server10.2.1.....	33
New for QuarkXPress Server10.1.2.....	34
New for QuarkXPress Server10.1.1.....	34
New for QuarkXPress Server10.1.....	34
<b>Known and resolved issues .....</b>	<b>35</b>
<b>Contacting Quark.....</b>	<b>36</b>
In the Americas .....	36
Outside the Americas .....	36
<b>Legal Notices .....</b>	<b>37</b>

# Introduction

QuarkXPress® Server is a high-fidelity content-composition and rendering server application. It is built on multiple technologies derived from QuarkXPress. Through a network connection and a browser, QuarkXPress Server can return rendered versions of QuarkXPress layouts and QuarkCopyDesk articles in a variety of formats. This document provides information about QuarkXPress Server software.

## Related Documents

For more information about using the QuarkXPress Server, you may need to reference the other QuarkXPress Server documentation/resources listed below.

Document/Item Name	Description	Where to find
QuarkXPress Server User Guide	A guide written for the end user of the QuarkXPress Server. Contains all of the QuarkXPress Server User Guide QXPS documentation, including documentation for QXPSM, and Modifier schema. This guide also contains all of the information which previously was contained in the Web Integration Guide.	<a href="http://www.quark.com/Support/Documentation/">http://www.quark.com/Support/Documentation/</a>
XDK	This is the C/C++ XTensions Developer Kit. It is comprised of XTension Headers, libraries and documentation.	Once installed, SDK is available at: <code>[installed directory]/Documents/SDK</code>
QXPSM Javadocs	This set of Javadocs provides information about the QuarkXPress Server Manager java interface. This documentation is available with the QuarkXPress Server Manager installer.	Once installed, Javadocs are available from the home page of QuarkXPress Server Manager: <a href="http://&lt;servernameorip&gt;:8090/">http://&lt;servernameorip&gt;:8090/</a>

# System Requirements

This section lists baseline minimum requirements for evaluation and demonstration. Like any enterprise system, performance is highly contextual based on data complexity, number of requests, and IT infrastructure. Your specific requirements for production shall require development as part of the overall solution planning process.

For more information, please contact <http://www.quark.com/support> or your Quark sales contact.

The following is a list of factors to consider when scaling up for production:

Number of requests per minute

- Content richness
  - Rendered page count
  - Number of raster images rendered
  - Number of vector images rendered
  - Number of images with transparency
  - Number of tables
  - Number of tables which span across multiple pages
  - Number of landscape pages
  - Number of footnotes in table cells
  - Number of cross-references
  - Number of image callouts or text callouts
- Output richness
  - PDF/print output styles
  - Digital output, e.g. number and size of interactive artifacts

## System requirements: macOS

### Software

- macOS 10.12 (Sierra™) or macOS 10.13.x (High Sierra™)
- Java J2SE™ Runtime Environment (JRE™) 8.0

## SYSTEM REQUIREMENTS

### Hardware

- 1 processor core and 4 sub-renderers with 4GB of RAM for each renderer\*
- 10GB of additional disk space per renderer for temporary files

➔ \*Like any enterprise system, performance is highly contextual based on data complexity, number of requests, and IT infrastructure. Contact your account manager for sizing your organization's publishing needs.

### System requirements: Windows

#### Software

- Microsoft® Windows® Server 2012 R2 64-bit and Windows® Server 2016 64-bit.
- .NET Framework 4.6.2 SP1 or later
- Java Development Kit (JDK) 1.8
- Java J2SE™ Runtime Environment (JRE™) 8.0
- Quark License Administrator (QLA) 4.6

➔ .NET Framework is not included in the installer. Download and install from [www.microsoft.com](http://www.microsoft.com) before starting QuarkXPress Server. You can install this component before or after installing QuarkXPress Server.

#### Hardware

- \*1 processor core and 4 sub-renderers with 4GB of RAM for each renderer
- 10GB of additional disk space per renderer for temporary files
- 5GB to 10GB hard disk space for installation
- CPU: 4 Core Intel Xeon, 64-bit
- TCP/IP network (Gigabit Ethernet recommended)

#### Power policy settings

QuarkXPress Server achieves better throughput under “High Performance” power policy setting as compared to a “Balanced” power policy setting.

- If QuarkXPress Server is installed on a bare metal machine, the power policy should be configured to “High Performance” on that machine.
- If QuarkXPress Server is installed on a virtual machine (Hyper V or ESXi), the power policy settings of the host machine should be configured to “High Performance” so that all virtual machines running on this host inherit this property.

➔ \*Like any enterprise system, performance is highly contextual based on data complexity, number of requests, and IT infrastructure. Contact your account manager for sizing your organization's publishing needs.

# Installing QuarkXPress Server

The topics below describe how to install QuarkXPress Server.

## Installing QuarkXPress Server: macOS

To install QuarkXPress Server:

1. Disable any virus protection software. If you are concerned about a virus infecting your computer, run a virus check on the installer file and then disable your virus detection software.
  2. Double-click the Installer icon and follow the instructions on the screen.
  3. When prompted, enter the validation code.
- ➔ Do not discard the validation code. You will need it if you have to reinstall.
4. In the **Quark License Server Details** screen, enter the IP address and port number of the main Quark® License Administrator (QLA) server in the **License Server Host Name/IP Address** and **License Server Port** fields. Optionally, enter the IP address and port number of a backup QLA server in the **Backup License Server Host Name/IP Address** and **Backup License Server Port** fields.
  5. In the **Port and Folder Settings** screen, enter a value in the **Port** field and specify the location of the QuarkXPress Server document pool.
  6. In the **Subrenderer Configuration** screen, enter the number of renderers you want to run. The number of available renderers is determined by the QLA license file.
  7. Complete the installation.
- ➔ After its initial launch, QuarkXPress Server creates a “Preferences” folder for each renderer. For example, if you have two renderers, QuarkXPress Server creates two folders named “Preferences sub1” and “Preferences sub2”. QuarkXPress Server creates preferences files at the following location:
- ```
[drive]/Users/[username]/Library/Preferences/Quark/QuarkXPress  
Server Renderer
```

## Installing QuarkXPress Server: Windows

To install QuarkXPress Server:

## INSTALLING QUARKXPRESS SERVER

1. Disable any virus protection software. If you are concerned about a virus infecting your computer, run a virus check on the installer file and then disable your virus detection software.
2. Double-click the Installer icon and follow the instructions on the screen.
3. In the **Software Verification** screen, enter the validation code.  
➔ Do not discard the validation code. You will need it if you have to reinstall.
4. When the **License Server Details** screen displays, enter the IP address and port number of the main Quark License Administrator (QLA) server in the **Host Name or IP Address** and **License Server Port** fields. Optionally, enter the IP address and port number of a backup QLA server in the **Backup Host Name or IP Address** and **Backup License Server Port** fields.
5. When the **Configuration Settings** screen displays:
  - Enter a port number for QuarkXPress Server.
  - To install QuarkXPress Server as an automatic service that launches when the computer starts up, check **Install QuarkXPress Server as a service**. You can choose **Local** or **Domain** from the **Choose System** drop-down menu. If you choose **Domain**, enter values in the **Domain**, **User**, and **Password** fields.
6. When the Destination screen displays:
  - The screen displays the drive and folder where QuarkXPress Server will be installed. To install QuarkXPress Server elsewhere, click **Browse** and navigate to the desired folder.
  - In the **Number of renderers to launch** field, enter the number of renderers you want to run. The number of available renderers is determined by the QLA license file.
7. Complete the installation.
8. When the installation completes, click **Finish**.  
➔ After its initial launch, QuarkXPress Server creates a “Preferences” folder for each renderer. For example, if you have two renderers, QuarkXPress Server creates two folders named “Preferences sub1” and “Preferences sub2”. The `getserverinfo` request returns the path of the QuarkXPress Server preferences under the element `PREFERENCESPATH`.  
➔ The URL for getting the QuarkXPress Server Preferences Path:  
`http://localhost:8080/getserverinfo`

# Upgrading QuarkXPress Server

1. Uninstall your current version of QuarkXPress Server as described in [“Uninstalling”](#).
2. Install the new version of the software as described in [“Installing QuarkXPress Server”](#).

# Deploying QuarkXPress Server Manager

There are three ways to deploy QuarkXPress Server Manager:

- As a standalone server. To deploy QuarkXPress Server Manager in this manner, install the software as described in “[Installing QuarkXPress Server Manager](#),” then double-click the `server/QXPMServerStart.command` (Mac OS X) or `server/QXPMServerStart.bat` (Windows) file in the QuarkXPress Server Manager application folder.
- As an independent Web application in an existing Tomcat Web container. For more information, see “[Deploying QXPSM in external Tomcat](#).”
- As a Web application that shares the same Spring context. For more information, see “[Deploying QXPSM in a shared Spring context](#).”

## Installing QuarkXPress Server Manager

QuarkXPress® Server Manager coordinates rendering requests in a multiple QuarkXPress Server instance environment, using load-balancing, fail-safe, and caching capabilities to determine which server in the pool can best process each document request. If the first QuarkXPress Server instance in an array is unable to process a render request, QuarkXPress Server Manager sends the request to another instance. QuarkXPress Server Manager also provides an object-oriented programming environment for working with QuarkXPress Server, and it exposes a set of Web services and Web objects that can be used in any SOAP-compatible environments such as .NET, JavaScript™, Java™, and Objective-C.

The QuarkXPress Server Manager installation includes three primary components:

- QuarkXPress Server Manager Server (the server application that provides services for load-balancing, fault-tolerance, and the SOAP server that allows third-party communication with QuarkXPress Server instances through Web services).
- The QuarkXPress Server Manager Web-based administrative client.
- The QuarkXPress Server SDK, which includes Web services API documentation; Web service stubs and samples for .NET, Java, and Objective-C; and an Extensibility Tool for generating custom stubs. This component is helpful for developing client-side solutions using the QuarkXPress Server Manager Web services and servlet interface.

## Installing QuarkXPress Server Manager: macOS

- ➔ You must uninstall the previous version of QuarkXPress Server Manager before installing this version. See “[Uninstalling](#)” for more information.

To install QuarkXPress Server Manager:

1. Disable any virus protection software. If you are concerned about a virus infecting your computer, run a virus check on the installer file and then disable your virus detection software.
2. Double-click the Installer icon and follow the instructions on the screen.
3. In the **QuarkXPress Server Manager Server Port Settings** screen, enter a port number for the QuarkXPress Server Manager server’s Tomcat server and specify the location of the QuarkXPress Server Manager cache folder. (The cache folder will store temporary copies of content generated by the QuarkXPress Server instances that are managed by this QuarkXPress Server Manager server.)
4. Complete the installation.

## Installing QuarkXPress Server Manager: Windows

- ➔ You must uninstall the previous version of QuarkXPress Server Manager before installing this version. See “[Uninstalling](#)” for more information.

To install QuarkXPress Server Manager:

1. Disable any virus protection software. If you are concerned about a virus infecting your computer, run a virus check on the installer file and then disable your virus detection software.
2. Double-click the Installer icon and follow the instructions on the screen.
3. When the **Destination Folder** screen displays:
  - The screen displays the drive and folder where QuarkXPress Server will be installed. To install QuarkXPress Server elsewhere, click **Browse** and navigate to the desired folder.
  - Enter a port number for QuarkXPress Server Manager.
  - Specify a cache folder location. (The cache folder will store temporary copies of content generated by the QuarkXPress Server instances that are managed by this QuarkXPress Server Manager server.)
  - To install QuarkXPress Server Manager Server as a service, check **Install QuarkXPress Server Manager Server as a Service**.
4. Complete the installation.

## Deploying QXPSM in a shared Spring context

In this type of deployment, QuarkXPress Server Manager can be embedded in a Spring application and can share the same Spring container instance and Spring context. This deployment model also allows the consuming of QuarkXPress Server Manager services (`RequestService` and `AdminService`) as POJOs (plain old Java objects).

## DEPLOYING QUARKXPRESS SERVER MANAGER

- ➔ This type of deployment model is possible only when the application in which QuarkXPress Server Manager is to be embedded is a Spring application and is deployed with an embedded or external Tomcat Web container.

To install QuarkXPress Server Manager in a shared Spring context:

1. Copy the `qxpsm` and `qxpsmadmin` Web application to the `webapps` folder for your existing application, so that QuarkXPress Server Manager can be recognized as a Web application in the existing Web container.
  - Set `cacheFolderContextFile` to the path of the “cache.xml” file relative to the current working directory.
  - Set `serverXmlFilePath` to the path of the “Server.xml” file relative to the current working directory.
2. Copy all QXPSM-dependent jar files from `[QXPSM_Home]/Server/dependencies` and `[QXPSM_Home]/Server/lib` to a common location, then add all of these jar files to the application classpath so that they are available to other Web applications that want to use the QuarkXPress Server Manager API locally.
3. Import the QuarkXPress Server Manager Spring context file, “ManagerContainerConfig.xml,” into the existing application Spring context.
4. Create a Java class named `CustomContextLoaderListener` that extends `org.springframework.web.context.ContextLoaderListener`. Override the method `loadParentContext()` as shown here:

```
public class CustomContextLoaderListener extends
org.springframework.web.context.ContextLoaderListener
{
    private static final Logger logger =
Logger.getLogger(GlobalContextLoaderListener.class);
    public void contextInitialized(ServletContextEvent
servletContextEvent)
    {
        super.contextInitialized(servletContextEvent);
        String contextName =
servletContextEvent.getServletContext().getServletContextName(
);
    }
    protected ContextLoader createContextLoader()
    {
        return new CustomContextLoader();
    }
    class CustomContextLoader extends ContextLoader
    {
        protected ApplicationContext
loadParentContext(ServletContext servletContext)
        {
            //=====
            =
                //Here, returns the spring context to be used as
parent context
            //=====
            =
                return
com.xyz.myapplication.MyApplicationContext.getContext();
        }
        protected WebApplicationContext
```

```

createWebApplicationContext(ServletContext servletContext,
    ApplicationContext parent)
    {
        return
        super.createWebApplicationContext(servletContext, parent);
    }
}

```

5. Create a file named “applicationContext.xml” in `webapps/qxpsm/WEB-INF` and fill it with the following content:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">
<beans default-lazy-init="false" default-autowire="no"
default-dependency-check="none">
</beans>

```

6. Open the QuarkXPress Server Manager “web.xml” file and update `contextConfigLocation` and `listener-class` as follows:

```

<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/applicationContext.xml</param-value>
</context-param>
<listener>
<listener-class>CustomContextLoaderListener</listener-class>
</listener>

```

7. If the existing application is already using Spring’s `PropertyPlaceholderConfigurer`, open “ManagerContainerConfig.xml” and specify the parent in the `placeholderConfig` bean as follows:

```

<bean id=" placeholderConfig " parent="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceh
olderConfigurer">

```

# Using Amazon S3 as a document pool in QuarkXPress Server

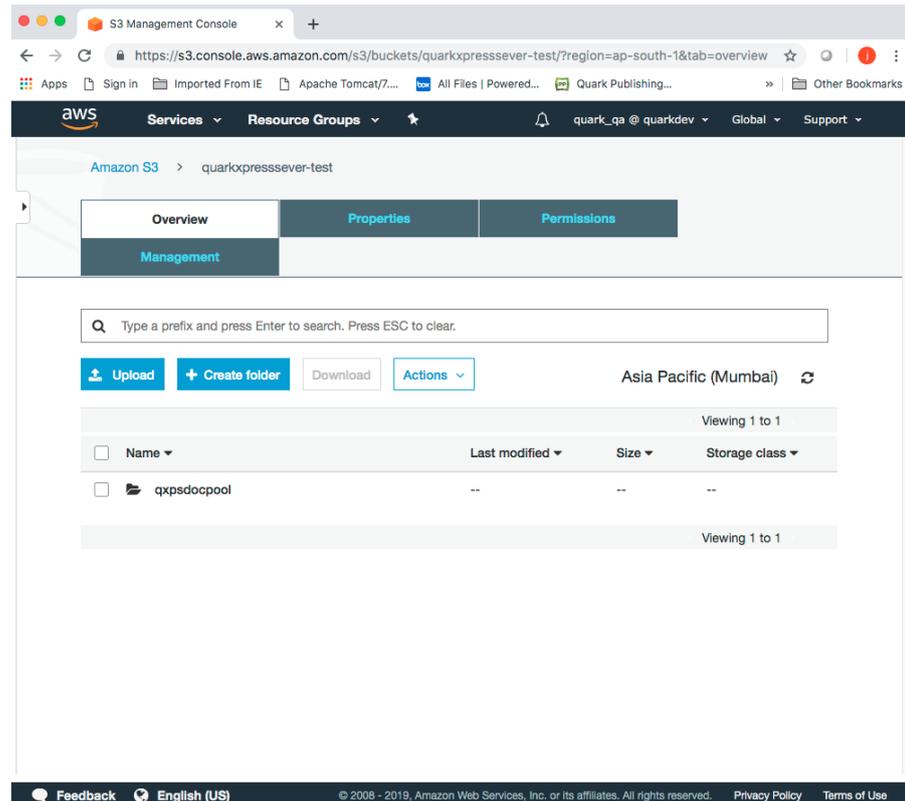
This section describes the steps necessary to use Amazon S3 as a docpool in QuarkXPress Server.

- ➔ We recommend installing/using QuarkXPress Server on an AWS EC2 Instance to use S3 as document pool for QuarkXPress Server.

## Create an S3 bucket in AWS

To create an S3 bucket in AWS:

1. Log on to the AWS Console using a valid account and user.
  2. On the **AWS Management Console**, under **Storage**, click **S3**.
  3. Click **Create bucket**.
  4. On the **Name and Region** pane, specify the bucket name and an appropriate region, and click **Next**.
  5. On the **Configure Options** pane, **DO NOT CHECK Keep all versions of an object in the same bucket**, and click **Next**.
- ➔ This is mandatory as QuarkXPress Server uses S3 as temporary storage and does not require versioning.
6. On the **Set permissions** pane, block all public access to this bucket, and click **Next**.
  7. On the **Review** pane, review your selections and click **Create bucket**.
  8. Select the newly created bucket and click **Create Folder**.
  9. Create a “qxpsdocpool” folder inside the bucket which will be used as document folder.



➔ Optionally an appropriate server-side encryption can be chosen for data at rest.

### Create a policy

The following are the set of S3 bucket permissions required by QuarkXPress Server:

- s3:PutObject
- s3:GetObject
- s3:ListBucket
- s3>DeleteObject

Create a policy and assign the S3 bucket permissions to the EC2 instance where QuarkXPress Server is running.

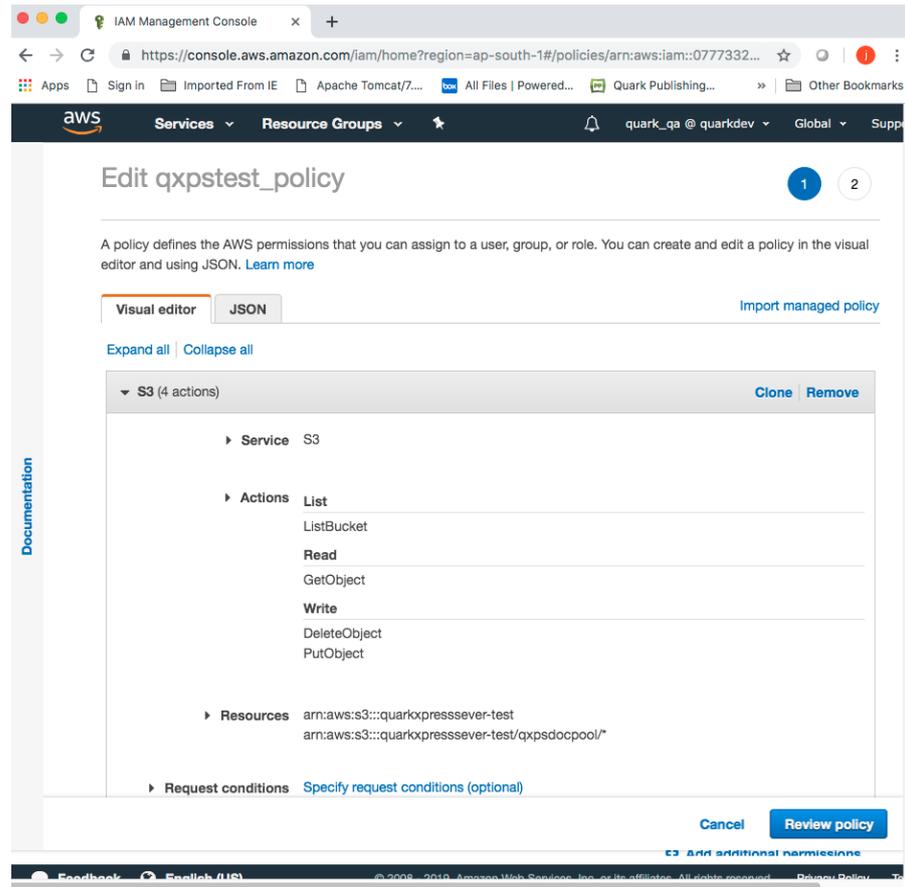
To create a policy and assign the permissions:

1. On the **AWS Management Console**, under **Security, Identity, & Compliance**, click **IAM**
2. Select **Policies** from the **Dashboard** menu.
3. Click **Create policy**.
4. Select the **JSON** tab and create the policy using the following json:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Sid": "VisualEditor0",
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:ListBucket",
    "s3:DeleteObject"
  ],
  "Resource": [
    "arn:aws:s3:::quarkxpressever-test",
    "arn:aws:s3:::quarkxpressever-test/qxpsdocpool/*"
  ]
}
```

5. On the **Review policy** pane, provide an appropriate Name and Description and click **Create policy**.
6. You can also create the policy using the Visual Editor also as shown below:



## Create a role

- ➔ If QuarkXPress Server running an EC2 instance has a role already assigned to it, this step can be skipped by simply attaching the newly created policy to that role.

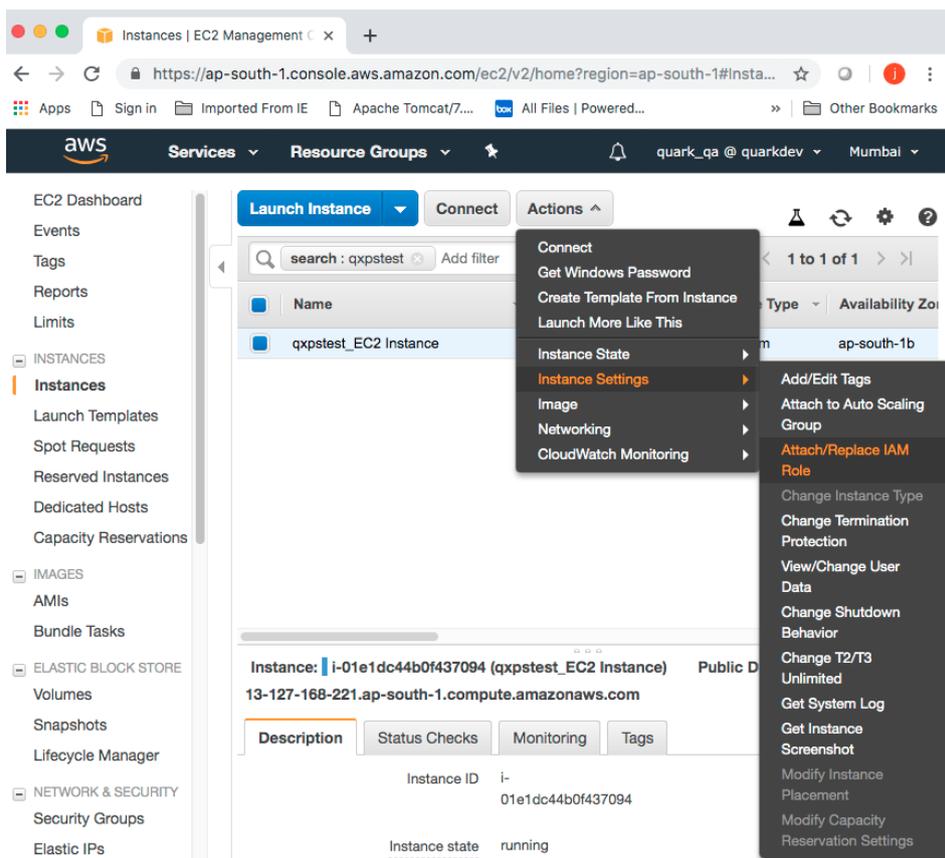
Create a role and assign your newly created policy to this role:

1. Select **Roles** from the **Dashboard** menu.
2. Click **Create role**.
3. Click **AWS service** and select **EC2** as the service that will use this role.
4. Click **Next: Permissions**.
5. Under **Attach permissions policies**, select the policy you have created in the previous step.
6. Click **Next: Tags**.
7. Click **Next: Review**.
8. On the **Review** pane, provide a **Role name**, review your selections and click **Create role**.

## Assign role to EC2 instance

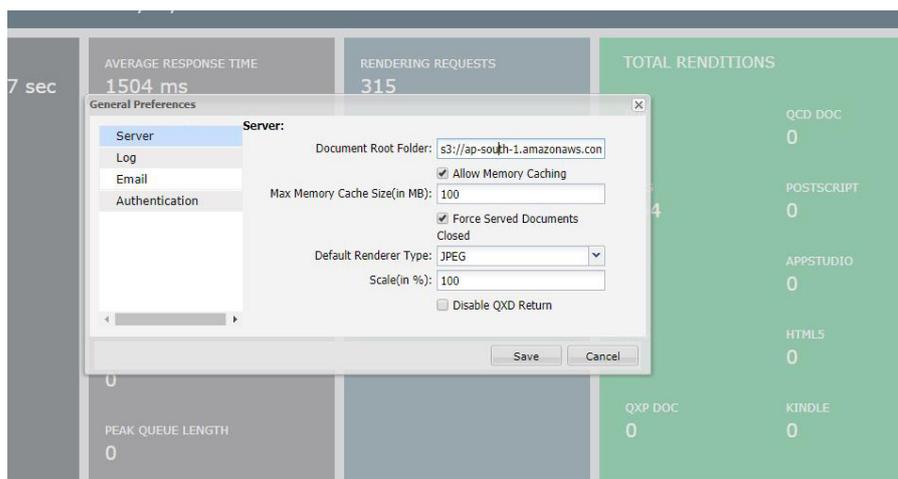
If you created a role using the instructions in the previous section, attach that role to the QuarkXPress Server's Platform EC2 instance:

## USING AMAZON S3 AS A DOCUMENT POOL IN QUARKXPRESS SERVER



### Register Amazon S3 as a Docpool in QuarkXPress Server

1. Open QuarkXPress Server's web-based Admin UI.
2. Go to **Administration > Preferences > General**.



3. In the **Document Root Folder**, specify the docpool URL:

`s3://ap-south-1.amazonaws.com/QXPS-Bucket/qxpsdocpool/`

- ➔ Alternately, you can set the document pool folder by using the following request:
- `http://localhost:8080/setprefs?DocumentRootFolder=s3://ap-south-1.amazonaws.com/QXPS-Bucket/qxpsdocpool/`

# Deploying in an external Web container

For development, and simple test deployments, QuarkXPress Server and QuarkXPress Server Manager are provided pre-installed in an instance of Apache Tomcat. However, for production deployment alongside other Web applications, QuarkXPress Server and QuarkXPress Server Manager can be deployed in an existing Tomcat or WebSphere.

## Deploying QXP Server in external Tomcat

To install QuarkXPress Server as an independent Web application in an existing Tomcat Web container:

1. Verify that Tomcat 7.0.77 is installed.
2. Set up the following environment variables:
  - `JAVA_HOME` = Java installation folder. For example: `E:\Program Files\Java\jdk1.8.0_121`
  - `CATALINA_HOME` = Tomcat installation folder. For example: `E:\apache-tomcat-7.0.77`
3. Copy contents of the `[QXPS_BUILD]/webapps` folder to the `[TOMCAT_HOME]/webapps` folder.
4. Copy the `[QXPS_BUILD]/conf` folder to `[TOMCAT_HOME]/bin`.
5. (*macOS only*) Open the file `[Tomcat_Home]/bin/conf/ServerApp.properties` and make the following changes:
  - Uncomment the Mac OS X settings section:
 

```
#qxpserver.serverRendererPath=./webapps/ROOT/QuarkXPressServer
Renderer/QuarkXPress Server.app/Contents/MacOS
#qxpserver.serverRendererExecutableName=QuarkXPress Server
#qxpserver.DocumentRootFolder=/QuarkXPress Server
Documents
```
  - Comment out the Windows settings section:
 

```
qxpserver.serverRendererExecutableName=QuarkXPress Server
Renderer.exe
qxpserver.DocumentRootFolder=C:\\QuarkXPress Server
Documents
```
6. In the `[TOMCAT_HOME]/bin/conf/qla.properties` file, set the host name, port number, and serial number of your instance of QLA Server.

## DEPLOYING IN AN EXTERNAL WEB CONTAINER

- Enter the IP address or hostname of the QLA Server in the `QlaServer.machinename=` field.
  - Enter the port number of the QLA Server in the `QlaServer.port=` field. If you have a backup QLA server, enter the IP address (or hostname) and port number in the `Backup.QlaServer.machinename=` and `Backup.QlaServer.port=` fields.
7. (Windows only) Execute `vcredist_x64.exe` provided with the build before starting QXPS.
  8. Copy the `Documents` folder to `[TOMCAT_HOME]`.
  9. Open `[TOMCAT_HOME]/conf/server.xml` and add the following inside the `<Host name="localhost" appBase="webapps".....>` tag:

```
<Context path="/qxps/documents"
docBase="../../Documents/Documentation" >
<Manager pathname=""/>
</Context>
```

Uncomment the Mac OS X settings section.

### Deploying QXPSM in external Tomcat

To install QuarkXPress Server Manager as an independent Web application in an existing Tomcat Web container:

1. Verify that Tomcat 7.0.77 is installed.
2. Locate the Tomcat installation folder (referred to here as `[Tomcat_Home]`).
3. Locate the QuarkXPress Server Manager external Tomcat installation folder (referred to here as `[QXPSM_Tomcat]`).
4. Drag the contents of `[QXPSM_Tomcat]/conf` to `[Tomcat_Home]/conf`.
5. Drag the contents of `[QXPSM_Tomcat]/webapps` to `[Tomcat_Home]/webapps`.
6. Drag `[QXPSM_Tomcat]/qxpsm` to `[Tomcat_Home]`.
7. Open the file `[Tomcat_Home]/conf/ManagerContainerConfig.xml` and make the following changes in the `ContainerAdapter` bean:
  - Set `cacheFolderContextFile` to the path of the “cache.xml” file relative to the current working directory.
  - Set `serverXmlFilePath` to the path of the “Server.xml” file relative to the current working directory.
8. Open the file `[Tomcat_Home]/conf/Catalina.properties` and specify the path of the shared class loader. For example:

```
shared.loader=${catalina.home}/qxpsm/lib/*.jar,${catalina.home}/conf
```
9. Open the file `[Tomcat_Home]/conf/Catalina/localhost/cache.xml` and set the `docbase` attribute to the path of the appropriate cache folder.
10. Double-click the “startup.sh” shell file (macOS) or the “startup.bat” file (Windows). These files can be found in the `TOMCAT_HOME/bin` folder.

## Deploying QXP Server in WebSphere

To install QuarkXPress Server as an independent Web application in an IBM WebSphere Web container:

### Prerequisites

1. Download the IBM Installation Manager 1.6.2 software and extract its contents to your local folder (for example, `IBM_MANAGER_DIR`).
2. Download the IBM WebSphere 9.0.0.3 developer version software. This is in 3 parts in an archived (.zip) format.
3. Extract all 3 parts of IBM WebSphere 9.0.0.3 to your local folder (for example, `IBM_WEBSPHHERE_DIR`).
4. Download the *SDK for Java 1.8* for IBM WebSphere 9.0.0.3, This is in 3 parts in an archived (.zip) format.
5. Extract all 3 parts of *SDK for Java 1.8* to your local folder (for example, `SDK-FOR-Java-1.8_DIR`).

### Installing the IBM-WebSphere developer version

1. Go to the `$(IBM_MANAGER_DIR)` folder and run “install.exe” with administrator privilege.
2. Launch the IBM Installation Manager 1.6.2 with administrator privilege.
3. Go to **File->Preferences**. The **Preferences** dialog displays. Select the **Repositories** option on the left side.
4. Click the **Add Repository** button from the right side. Browse to and select the “respository.config” file from the `$(IBM_WEBSPHHERE_DIR)/was.repo.9003.developers.ilan_part1` folder.
5. Click **Apply** and then click **OK**.
6. Go to the IBM Installation Manager, select the **Install** option.  
  
IBM WebSphere 9.0.0.3 will be installed in your Program Files location.  
`C:\Program Files (x86)\IBM\WebSphere.`
7. During the installation, you will be asked to enter the configuration file for Part 2 and Part 3 of IBM WebSphere. Select the files from their respective folders:
  - `$(IBM_WEBSPHHERE_DIR)/was.repo.9003.developers.ilan_part2`
  - `$(IBM_WEBSPHHERE_DIR)/was.repo.9003.developers.ilan_part3`
8. Go to **Start->WebSphere Customization Toolbox**. Select the **Create** option on the right side.  
  
The Profile Management Tool dialog displays.
9. Select `Application Server` as the WebSphere Application Server and click **Next**.
10. Select **Typical file creation** and click **Next**.

11. Select **Disable Administrative Security** and click **Next**.
12. Select **Create a profile**.

### Configure the Java 7 SDK in the IBM WebSphere

1. Open the cmd window. Navigate to the `$(WS_INSTALLED_DIR)/AppServer/bin` folder.
2. Type the command `managesdk.bat -listAvailable`.
3. Launch the IBM installation manager.
4. Go to **File->Preferences**. The **Preferences** dialog displays. Select the **Repositories** option on the left side.
5. Click the **Add Repository** button from the right side. Browse to and select the “repository.config” file from the `$(SDK-FOR-Java-1.8_DIR)/was.repo.9003.java7_part1` folder.
6. Click **Apply** and then click **OK**.
7. Go to the IBM Installation Manager, select the **Install** option.  
  
IBM WebSphere 9.0.0.3 will be installed in your Program Files location.  
`C:\Program Files (x86)\IBM\WebSphere`.
8. During the installation, you will be asked to enter the configuration file for Part 2 and Part 3 of IBM WebSphere. Select the files from their respective folders:
  - `$(SDK-FOR-Java-1.8_DIR)/was.repo.9003.java7_part2`
  - `$(SDK-FOR-Java-1.8_DIR)/was.repo.9003.java7_part3`
9. The Java7 SDK for IBM WebSphere will be installed successfully. Check it at the `$(WS_INSTALLED_DIR)/AppServer/java_1.8_64` folder
10. Open the cmd window. Navigate to the `$(WS_INSTALLED_DIR)/AppServer/bin` folder.
11. Type the command `managesdk.bat -listAvailable`.
12. Type the command `managesdk.bat -enableProfile -profileName AppSrv01-sdkname 1.8_64`. This binds IBM WebSphere into Java7 runtime.

### Start IBM WebSphere

1. Enter `First Step.bat` from the start menu or from `$(WS_INSTALLED_DIR)/AppServer/profiles/AppSrv01/firststeps/firststeps.bat`.  
  
The **WebSphere Application server** dialog displays.
2. Start the Server.

## QXPS WebApp deployment into IBM WebSphere UI

- Copy the [QXPS\_BUILD]/conf folder to WebSphere's current directory (for example, \$(WS\_INSTALLED\_DIR)/AppServer/profiles/AppSrv01). Open "ServerApp.properties" and update the following:
  - `qxpswebserver.embeddedwebcontainer= false`
  - `qxps.webapp.context.name= /qxps`
- Open the "QXPSLoadComponents.properties" file located at \$(WS\_INSTALLED\_DIR)/AppServer/profiles/AppSrv01/conf and remove the following entries:
 

```
<import
resource="classpath:com/quark/azure/auth/AzureAuthContext.xml"
/>
<import
resource="classpath:com/quark/qxps/azure/config/AzureConfigura
tion.xml"/>
<import
resource="classpath:com/quark/qxps/azure/config/AzureConfigura
tion.xml"/>
```
- Open the "PropertiesFilePlaceholders.properties" files located at \$(WS\_INSTALLED\_DIR)/AppServer/profiles/AppSrv01/conf and remove the following entries:
  - `<value>file:./conf/AzureAuthenticate.properties</value>`
- In the [TOMCAT\_HOME]/bin/conf/Qla.properties file, set the host name, port number, and serial number of your instance of QLA Server.
  - Set `QlaServer.machinename` to the IP address or hostname of the QLA server.
  - Set `QlaServer.port` to the port number of the QLA server.
  - If you have a backup QLA server, set `Backup.QlaServer.machinename` and `Backup.QlaServer.port` to its IP address or hostname and port number.
- Extract the "EAR (qxps-2018.ear)" file. Open the "qxps-servlet.xml" file located at \$(QXPS-EAR\_DIR)\qxps-2018\source\qxps\WEB-INF\ and remove the following entries:
 

```
<value>file:./conf/AzureAuthenticate.properties</value>
<property name="authManager"
ref="azureAuthenticationManager"/>
<prop key="/secure/azureauth">azureAuthReplyController</prop>
<property name="authResultCBHandler"
ref="qxpsAzureAuthResultCBHandler"/>
<property name="authManager"
ref="azureAuthenticationManager"/>
```
- Create the WAR bundle by using the "CreateQXPS\_EAR.bat" batch file located at the extracted folder: \$(QXPS-EAR\_DIR)
- Open <http://localhost:9080/ibm/console/> in a browser and login.
- Select the **Application** tab from the right and select **New Application**.
- Select **New Enterprise Application**. When asked for the path to the new application, choose the file and click **Next**.

## DEPLOYING IN AN EXTERNAL WEB CONTAINER

10. When asked “How do you want to install the application?”, select Fast Path and click Next.
11. Change the application name if required and click Next.
12. Select both the modules and click Next.
13. Select both the web modules and click Next.
14. Select the qxpsadmin module and click Next and then Finish.
15. Save the changes to the master configuration.
16. Select the **Server** tab from the right and select **WebSphere Application Servers> server1 > Web server plug-in properties**.
17. Set the maximum number of connections that can be handled by the application server to 150.
18. Save the changes.
19. Select the **Application** tab from the right and select **Application Types -> WebSphere Enterprise applications**. Select **qxps** and **Start**.

# Introduction to QLA Server

QLA Server is the software component that issues licenses to the Quark application clients. After you install Quark License Administrator Server, download your license file from the Quark Website. The server can be any computer on the network that meets the minimum system requirements. The server monitors license usage over the network and grants licenses to the client computers.

QLA clients are the computers on which Quark applications are installed — for example, QuarkXPress, Quark Publishing System® (QPS®), or QuarkXPress Server software. When a user attempts to launch a Quark application on a client computer, the client sends a request to the QLA Server. The server, which is always listening for requests, determines whether there is an available license. If there is, then the server grants a license and the client computer can launch the Quark application. If no licenses are available, the server does not grant a license. When this occurs, the Quark application does not launch.

In the QLA Control Panel application, the **Configuration Options** tab lets you specify how QLA monitors licenses. This tab includes the **Server Configuration** area and the **Extended Check-out Configuration** area.

In the **Server Configuration** area:

- The **Heartbeat Interval (Seconds)** value lets you specify the number of seconds between attempts by the client to reach the server. If a Quark application fails to successfully connect to the server within this time, then that application automatically attempts to reconnect.
- The **Heartbeat Retries** field lets you specify the number of times a client can attempt to make a connection before reporting a connection failure.

For example: If the **Heartbeat Interval (Seconds)** value is 3600 (one hour) and the **Heartbeat Retries** value is 10, the system waits 10 hours before failing. If the client is able to successfully connect to the server on or before the tenth attempt, outstanding licenses are restored. If not, the application checks in all allocated licenses.

# Uninstalling

If the uninstaller detects a running instance of the application, an alert displays. Close the application before you begin to uninstall.

## Uninstalling: macOS

To uninstall QuarkXPress Server, open the `Applications` folder on your hard drive (or the folder in which QuarkXPress Server was installed) and drag the `QuarkXPress Server` folder to the trash.

To uninstall QuarkXPress Server Manager, open the `Applications` folder on your hard drive (or the folder in which QuarkXPress Server Manager was installed) and drag the `QuarkXPress Server Manager` folder to the trash.

## Uninstalling: Windows

To uninstall QuarkXPress Server or QuarkXPress Server Manager:

1. Choose **Start > All Programs > QuarkXPress Server 2018 > Uninstall** or **Start > All Programs > QuarkXPress Server Manager > Uninstall**.
2. Click **Yes**.

The uninstaller gathers the information necessary for uninstalling and then an alert displays to notify you that the contents of the application folder will be deleted.

3. Click **Yes** to continue or **No** to cancel.

# Changes in this version

This section will cover the various changes introduced in the current version of QuarkXPress Server.

➔ For details about all of these changes, see *A Guide to QuarkXPress Server*.

## New for QuarkXPress Server 2018 - September 2019 Update

Changes in version 14.3 included the following:

- INLINETABLES: Added the ability to specify multiple inset in a table cell.
- Introduced the ability to specify measurement units in case of deconstruct xml. Provided UI in QXPS renderer preferences dialog.

# Changes in previous versions

This section will cover the various changes introduced in previous versions of QuarkXPress Server.

➔ For details about all of these changes, see *A Guide to QuarkXPress Server*.

## New for QuarkXPress Server 2018 - February 2019 Update

Changes in version 14.2.1 included the following:

- QuarkXPress Server is compatible with QuarkXPress 2018 - January 2019 Update (14.2.1).
- Added support for Indic languages.
- Updated XTensions for Smart Content Toolkit

## New for QuarkXPress Server 2018

Changes in version 14.0.2 included the following:

- QuarkXPress Server is compatible with QuarkXPress 2018 (14.x).
- Added accessibility features in PDF output:
  - PDFs published from Platform now include tagging and alternate text for images to enhance accessibility.
  - Added a new URL parameter, `createtaggedpdf` (true|false), to configure the tagged PDF setting.
- Added support for box borders:
  - Added a new `BORDER` element with over 40 attributes to support the ability to apply different borders to each side of a box.
  - The `FRAME` element has been retained for backward compatibility and still works in modifier-based flows.
  - The deconstruct XML now represents borders and related attributes under the new `BORDER` element.
- Added support for the ability to span footnotes/endnotes over columns:
  - Specify using the `Footnote Across Columns` attribute in the footnote/endnote separator style.

- This feature is also supported in the automation flow.
- Modifier Schema Changes:
  - Added support for heading styles in paragraph stylesheets.
  - Added support for heterogeneous borders and frames.
  - Changed `EBOOKMETADATA` to `LAYOUTMETADATA`. This lets you apply metadata to the layout for PDS and eBook export.
  - Added `PICTURE@ALTTEXT`. This lets you set alternate text for each image to enhance accessibility.

### New for QuarkXPress Server 2017 October Update

Changes in version 13.1 included the following:

- QuarkXPress Server is compatible with QuarkXPress 13.1 and QuarkCopyDesk 13.1.
- Configuration changes:
  - New option to configure the path for the QuarkXPress Server “PublishingRescueFolder” to store publishing data to accelerate troubleshooting of publishing failures:
- Two ways to configure the path
  - Edit the “PublishingUtilContext.xml” file  
(`[installpath]/Server/publishing`) to set the path as a value for the `rescueFolderPath` property (requires server restart)
  - Use JConsole (no server restart required)

### New for QuarkXPress Server 2017 March 2018 Update

Changes in version 13.2.2 included the following:

- Compatible with QuarkXPress and QuarkCopyDesk 13.2.1
- **Modifier Schema Changes**
  - Added the ability to represent master spreads and its items in XML rendition.

## CHANGES IN PREVIOUS VERSIONS

- Added the ability to create or modify items on master spreads using modifier XML.
- Added the ability to represent and modify gradients.
- Added the ability to represent blend modes in XML and apply blend modes using modifier XML.
- Added the ability to represent page and spread guides in modifier deconstruct XML.
- Added the ability to represent lock type in deconstruct XML and lock items on the layout using Modifier XML.
- Added the ability to represent text stroke in deconstruct XML and apply text stroke using Modifier XML.
- Added the ability to apply a hyperlink in a table cell.

### New for QuarkXPress Server 2017

Changes in version 13.0 included the following:

- QuarkXPress Server is compatible with QuarkXPress 2017 including support for the following QuarkXPress 2017 features:
  - Export as Responsive HTML5 Publication (Multi-device Output)
  - Paragraph and text shading
  - Column Flow - Column Split and Span
- Added support for Item Styles
- Rendering metrics and monitoring - telemetry records metrics for executed rendering jobs:
  - Average & Peak Response Time
  - Busy Period
  - Utilization %
  - Total Transactions Processed by Renderers
  - Rendering Throughput
  - Arrival Rate of rendering requests
  - Average & Peak Queue Time
  - Average, Current, and Peak Queue Length
  - Failed Requests and Error %
  - Average Time per Page
  - Total Pages Rendered
  - Total Images Processed

- Total Renditions to PDF, HTML5, QXPDOC, etc.
- Request timeout and ability to collect data for failed requests in a rescue folder to simplify troubleshooting
- Transaction-level font management

### New for QuarkXPress Server 2016

Changes in version 2016 Update included the following:

- QuarkXPress Server is compatible with QuarkXPress 2016.
- Introduced new attributes (`MINWIDTH`, `MAXWIDTH`, `SPAN`) under the `INLINETABLE` modifier. These new attributes provide greater control on table layout.
- Added support for index generation.
- Added support for equation publishing with Wiris Math Equations.
- Added support for table auto-fit rules.
- Added support for Azure Cloud Service configuration.
- Added support for using the DITA Open Toolkit as a publishing engine for your DITA content.
- Added the ability to apply a hyperlink on `INLINEBOX`.
- Added the ability to process nested `<PARAGRAPH>` elements.
- Added the ability to specify `ELLIPSIS/CLIP/CUSTOM STRING` for text overflow.
- Added new elements to Modifier Schema to support the cross-references feature introduced in QuarkXPress 2016.
- Introduced a new attribute (`TEXTOVERFLOW`) under the `STORY` modifier. Possible values are `ELLIPSIS` | `CLIP` | `CUSTOM STRING`. This attribute lets you control the representation of overflow text. The option `ELLIPSIS` renders an ellipsis “...” to represent over flow text. The option `CLIP` clips the text. The option `CUSTOM STRING` allows you to supply custom text. (e.g Continue Reading... or Read more...) to represent over flow text. These are particularly useful when you have a finite sized box designated to hold title text. If the title text is too lengthy to fit into the designated box on the Master Page, using the `TEXTOVERFLOW` attribute allows you to specify the behavior you want to see in the rendered output.

### New for QuarkXPress Server 2015 - April 2016 Update

Changes in version 2015 - April 2016 Update included the following:

- Added the ability to capture outline numbering in the Table of Contents.
- Added the ability to generate Table of Contents in a `BOX` element under a `STATICCONTENT` element.

## CHANGES IN PREVIOUS VERSIONS

### API Changes

- Migration from Ver-11.2.1 to QuarkXPress & QuarkXPress Server 2015 (11.2.2)
- Migration from Ver-11.1 to QuarkXPress & QuarkXPress Server 2015 (11.2.1)
- ➔ No migration anticipated unless you want to use new attributes. However, testing is recommended if you are using callouts ([CALLOUTANCHOR](#)).
- Migration from Ver-11.0. to QuarkXPress & QuarkXPress Server 2015 (11.1)
- ➔ No migration anticipated unless you want to use new attributes. However, testing is recommended if you are using callouts ([CALLOUTANCHOR](#)).

### New for QuarkXPress Server 2015 - February 2016 Update

Changes in version 2015 - February 2016 Update included the following:

- Added support for the new Regions feature by enhancing the [INLINETABLE](#), [INLINEBOX](#), and [CALLOUTANCHORS](#) behaviors in the following ways:
  - Added the ability to break tables nested inside a Box type region across pages.
  - Added the ability to place callouts in a Box type region.
  - Added the ability to place callouts in cells of tables nested in Box type regions.
- Added the ability to insert foot notes and endnotes in [INLINEBOX](#) and [CALLOUT](#) Boxes.
- Added the ability to insert foot notes and endnotes in Table Cells.
- Introduced the ability to get a JPEG rendition of all pages of a QuarkXPress Project as a multipart response or as a downloadable archive (.zip). The downloadable (.zip) has helped to improve Quark Publishing Platform's Live Image Preview.

### API Changes

- Migration from Ver-11.1 to QuarkXPress & QuarkXPress Server 2015 (11.2.1)
- ➔ No migration anticipated unless you want to use new attributes. However, testing is recommended if you are using callouts ([CALLOUTANCHOR](#)).
- Migration from Ver-11.0. to QuarkXPress & QuarkXPress Server 2015 (11.1)
- ➔ No migration anticipated unless you want to use new attributes. However, testing is recommended if you are using callouts ([CALLOUTANCHOR](#)).

### New for QuarkXPress Server 2015 - October Update

Changes in version 2015 - October Update included the following:

- Added the ability to control whether or not inline table rows will break across pages using the new `BREAKROWACROSSPAGES` attribute of the `INLINETABLE` element.
- Added the ability to control whether the callout anchor in the flow stream hinged to a callout must remain on the same page using the new `KEEPONSAMEPAGE` attribute of the `CALLOUTANCHOR` element. This additional flexibility supports cases where multiple anchored callouts need to be positioned near relevant text in the flow.
- Enhanced the ability to create a Table of Contents, so that you can create a Table of Contents even in a callout box.

### API Changes

- Migration from Ver-11.0 to QuarkXPress & QuarkXPress Server 2015 (11.1 )
  - ➔ No migration anticipated unless you want to use new attributes. However, testing is recommended if you are using callouts (`CALLOUTANCHOR`).
- Migration from Ver-10.5.x to QuarkXPress & QuarkXPress Server 2015 (11.0.1 or 11.1)
  1. The `FootNoteSXT` that was bundled in 10.5.x SDK is discontinued. The elements `FOOTNOTESTYLE` and `FOOTNOTE` used in 10.5.x deployments, need to migrate to the new footnote structure in QuarkXPress and QuarkXPress Server 2015. This would also mean elimination of the `FOOTNOTESTYLE` and `FOOTNOTE` mark up in your current XSLT's. The publishing templates are also expected to be saved in QuarkXPress 2015 (V-11.0.1) in order to use footnotes.
  2. The custom XT's have to be recompiled using latest XDK.
  3. One of the capabilities offered by `PageCountInsertionXT` is implicitly achievable using the new Content Variables feature of QuarkXPress 2015.
- Migration from Ver-10.2.2 to Ver-10.5/10.5.1
  - ➔ No new markup in ModifierXML schema. No migration effort is expected.
- Migration from Ver-10.1.x to Ver-10.2.2 OR Migration from Ver-10.1.x to Ver-10.2
  1. For flow automation use cases, Modifier schema does have some new markup however the existing ModifierXML's (via XSLT) will work in QuarkXPress Server Ver-10.2 as is. No migration effort is anticipated.
  2. For custom QuarkXPress Server XTension (Mac) using Ver-10.1.x XDK, it has to undergo migration to Ver10.2 in the form of compiling with Ver-10.2 XDK. The x64 target in XCode Project settings of the XT have to be changed appropriately (pointing to x64 XT\_LayoutEngine.lib) and compiled such that all 64-bit specific errors and warnings are eliminated.
- Migration from Ver-9.5.x to Ver-10.2
  1. `INLINETABLE@ORIENTATION` is obsolete. Instead, for landscape pages with landscape tables, use `PAGESEQUENCE@ORIENTATION`. This is the recommended approach for landscape oriented pages.

## CHANGES IN PREVIOUS VERSIONS

2. For flow automation use cases, Modifier schema does have some new markup. ModifierXML's (via XSLT) would seamlessly work in QuarkXPress Server Ver-10.x, as long as you are not using new markup in your 9.5.x transforms.
  3. For custom QuarkXPress Server XTension (Win) or (Mac) using Ver-9.x XDK, it has to undergo migration to 64-bit. The x64 target in Visual Studio Project/ Xcode settings of the XT have to be changed appropriately (pointing to x64 XT\_LayoutEngine.lib) and compiled such that all 64-bit specific errors and warnings are eliminated.
- Migration from Ver-8.x to Ver-10.2
- ➔ If you are a QuarkXPress Server Ver-8.x customer, a lot has been changed between Ver-8.x & Ver-9.x. Refer to the ReadMe documents.

### New for QuarkXPress Server 2015

Changes in version 2015 included the following:

- Added support for Footnotes in flow automation with the introduction of the new `REFNOTE` element, which allows you to insert a reference note of type footnote or endnote.
- Performance optimizations in the flow automation use cases.
- Enhanced the `INLINETABLE` element to allow the ability to compute the column widths of tables automatically, depending on the length of the text in a cell or image in a cell.

### New for QuarkXPress Server10.5.2

This version of QuarkXPress Server contains bug fixes.

### New for QuarkXPress Server10.5.1

This version of QuarkXPress Server contains performance improvement with memory

utilization and bug fixes.

### New for QuarkXPress Server10.5

Changes in version 10.5 included the following:

- Performance optimizations when parsing large modifier XML's (i.e. 200,000 or more lines).
- Introduced the `INLINETABLE@BREAKABLE` (true | false) attribute. This is an optional attribute and does not impact any existing transforms written for Modifier. If the `BREAKABLE` attribute is true (the default), inline tables will break across pages, if the `BREAKABLE` attribute is false, the inline table will NOT break across pages.

- Added the ability to display the Rendering Job processing timing details in the transaction log.
- Introduced a new QuarkXPress XTension, *PaginationRuleUIXT*. It enables template designers to define the sequence of the application of master pages during the process of dynamic pagination in flow document automation use cases.

## New for QuarkXPress Server 10.2.1

Changes in version 10.2.1 included the following:

- Added support for automatic callout stacking. This allows for tiling of multiple callouts by providing additional parameters in the modifier.
- Added support for nested anchoring, allowing you to anchor an item within another anchored item. This allows for nesting of `INLINEBOX` and `INLINETABLE`.
- The `INLINEBOX` element has been enhanced to support the following:
  - Column balancing, allows you to automatically cause text to flow and bottom align across multiple columns automatically.
  - Support for over-matter and straddle heads/straddle tables with multicolumn text in flow.
  - The ability to layout a mixture of single and multi-column sections on a page.
- QuarkXPress Server is now a 64-bit application on Mac OS X.
- Added the ability to apply interactivity on an `INLINEITEM` element.

## API changes

- Migrating from version 10.1.x to version 10.2.1
  - In flow automation use cases, the Modifier schema will have new markups, however your existing ModifierXML's (via XSLT) will work in QuarkXPress Server 10.2.1 as is. No migration effort is anticipated.
  - Custom QuarkXPress Server XTensions (*Mac OS X*) using the 10.1.x version of the XDK have to be recompiled with the 10.2 version of the XDK in order to migrate them for use in 10.2.1. (The x64 target in the XCode Project settings of the XTension have to be changed (to point to the x64 `XT_LayoutEngine.lib`) and compiled such that all 64-bit specific errors and warnings are eliminated.)
- Migrating from version 9.5.x to version 10.2.1
  - In flow automation use cases, the Modifier schema will have new markups, however your existing ModifierXML's (via XSLT) will work in QuarkXPress Server version 10.x, as long as your changes are anticipated in the 9.5.x transforms you have been using.

## CHANGES IN PREVIOUS VERSIONS

- Custom QuarkXPress Server XTensions (*Windows or Mac OS X*) using the 9.x version of the XDK have to undergo migration to a 64-bit. x64 target. The Visual Studio Project/ Xcode settings of the XTension have to be changed to point to the x64 XT\_LayoutEngine.lib) and compiled such that all 64-bit specific errors and warnings are eliminated.

### New for QuarkXPress Server 10.1.2

This version of QuarkXPress Server contains bug fixes.

### New for QuarkXPress Server 10.1.1

Changes in version 10.1.1 included the following:

- If any `ENTRY` element of `INLINETABLE` that corresponds to a text/picture cell, has several paragraphs of text or a picture of large height, such that the row can potentially grow beyond the height of the page, then an automatic row addition takes place and any remaining text is automatically placed in the new row. This behavior is applicable across pages.

### New for QuarkXPress Server 10.1

Changes in version 10.1.1 included the following:

- QuarkXPress Server is now a 64-bit application on Windows. On Mac OS X, it is still a 32-bit application.
- Added support in deconstruct and modifier for additional App Studio interactivities:
  - 360° image
  - Animation
- App Studio output now supports unbounded pages for vertical and horizontal layout.

# Known and resolved issues

For lists of known issues and resolved issues in this version of the software, visit the Quark Website at <http://www.quark.com>. Product documentation, including lists of known and resolved issues, is available under **Support > Product Documentation**.

# Contacting Quark

The support portal allows you to log support tickets, track tickets, receive status notifications, chat with a technical support representative, search the Knowledge Base, and access product documentation.

With direct access to documentation across all Quark software - from QuarkXPress and App Studio to Quark Enterprise Solutions - you can find answers to your questions at your convenience. Our support team is also available to help, either through our support portal or via phone for our maintenance contract customers.

If you are a Quark customer and have a current maintenance or support contract your account has already been created for you using your registered email address. If you do not have a support contract, you can purchase a single support incident to get your problem resolved. If you have purchased or registered a supported product, you are eligible for free support for the first 90 days.

## In the Americas

For more details, please check out our support website <http://www.quark.com/support>.

## Outside the Americas

For countries outside the Americas, please visit the following sites to access your support account:

Support Website

- France - <http://www.quark.com/fr/support>
- Germany - <http://www.quark.com/de/support>

# Legal Notices

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

Quark, the Quark logo, and QuarkXPress 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.