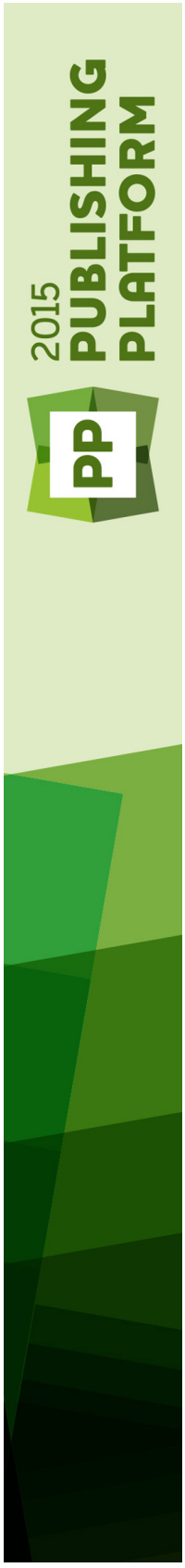


Quark Publishing Platform 2015 - 2016 年 9 月アップデート システム管理ガイド

目次

Quark Publishing Platform管理作業について

Quark® Publishing Platform™環境を管理するには、Quark Publishing Platformの制御だけでなく、ハードウェアおよびソフトウェアの幅広いメンテナンスが必要です。本書では、管理者が実行するセキュリティ、システム設定調整、および他のシステムとの統合の作業について説明します。Quark Publishing Platformインターフェイスでの管理については、『**A Guide to Quark Publishing Platform**』を参照してください。Quark Publishing Platformソフトウェアのインストールについての詳細は、『**Quark Publishing Platform ReadMe**』を参照してください。

Quark Publishing Platform Serverの外部サーバーコンテナへの配置

外部Tomcatインストール、IBM WebSphere、およびOracle WebLogicに、Quark Publishing Platformを配置できます。

環境変数の設定

以下の環境変数を設定し、配置を準備します。

Quark Publishing Platform Serverをサービスとして配置するには、これらをグローバルなシステム環境変数として設定します。

- **MAGICK_HOME**:[ImageMagick installation directory]¥bin
- **MAGICK_FILTER_MODULE_PATH**:%MAGICK_HOME%/modules/filters
- **MAGICK_CODER_MODULE_PATH**:%MAGICK_HOME%/modules/coders

JVMの設定

使用するアプリケーションサーバーの種類にかかわらず、Quark Publishing Platform Serverに対応するため、Tomcat、IBM WebSphere、およびOracle WeblogicでJVMに割り当てるメモリを下記のように増やす必要があります。

- Javaヒープメモリの最小値を2,048MBに設定します。
- PermGenSpaceの最小値を512MBに設定します。

EARファイルの配置準備

Platform EARビルドの構成内容は以下のとおりです。

- Platform Server設定ファイル
- 依存関係のバイナリ
- JEE EARファイル生成ツール

EARビルドには下記のフォルダが含まれます。

- conf：Platform Serverの設定ファイルが含まれます。
- database：データベースの作成ファイル、更新ファイル、サポートファイルが含まれます。
- ImageMagick：ImageMagickアプリケーションバイナリが含まれます。
- Jaws：JAWSアプリケーションバイナリが含まれます。
- DITA-OT1.6：DITA OTバイナリが含まれます。
- publishing：Quark Publishing Platformフレームワークの構成ファイルおよびリソースファイルが含まれます。

Platform Serverアプリケーションは、以下のモジュールで構成されます。

- admin.war：Quark Publishing Platform 管理Webアプリケーション
- workspace.war：Quark Publishing Platform ワークスペースWebアプリケーション
- webservices.war：Quark Publishing Platform Server Web サービスモジュール
- rest.war：Quark Publishing PlatformサーバーRESTインターフェイスモジュール
- messaging.war：HTTPベースメッセージングインターフェイス
- qxpsm.war：QuarkXPress Server Manager
- qxpsmadmin.war：QuarkXPress Server Manager管理Webアプリケーション
- pluginwiris_engine.war：Wirisプラグイン数式サポート

Quark Publishing Platform Server JEEエンタープライズアーカイブを生成するには、下記の手順に従ってください。

1. Quark Publishing Platform EARビルドをアプリケーションサーバーを実行しているコンピュータにコピーします。
2. 「*Quark Publishing Platform Server — 手動設定*」の説明に従って、設定ファイル（"Database.properties"、"managerconfig.xml"、"PluginContext.xml"、"QLA.properties.xml"を含む）を更新します。
3. "serverApp.properties"で、JMX Agentの**username**および**password**の値を更新します。**username**および**password**の値は、WebSphereのAdminコンソールへのログオンに使用されるものと同じです。
4. RunEARUpdater.batを実行します。配布可能なQuark Publishing Platform EARファイルが含まれる"For Deployment"フォルダが作成されます。

IBM WebSphereへの配置

IBM WebSphereへ配置を行うには、下記の手順に従ってください（サポートされているバージョンは8.5.5.8です）。

1. EARファイルが配置されるアプリケーションサーバーインスタンスの作業ディレクトリに、"For Deployment"フォルダから"qpp"フォルダをコピーします。IBM WebSphereの場合、このフォルダが対象プロファイルに相当します。

➡ Quark Publishing Platformまたはその他のリソースをIBM FileNetに配布するには、[作業ディレクトリ]¥properties¥wsjaas.confファイルの最後に次の内容を追加します。

```
FileNetP8WSI {
    com.filenet.api.util.WSILoginModule required;
};
```

2 http://localhost:port/ibm/consoleに移動して、IBM WebSphereアプリケーションサーバーコンソールを起動します。

3 **アプリケーション**から、**新規アプリケーション**をクリックします。

4 **新規エンタープライズアプリケーション**をクリックします。

5 **新規アプリケーションのパス**から、**ファイルを選択**をクリックし、"For Deployment"フォルダ内の.earファイルを選択します。（詳細は、「*EARファイルの配置準備*」を参照してください。）。

6 **アプリケーションのインストールの準備**から、**次へ**をクリックします。

7 手順1、2、3、および4で**次へ**をクリックします。設定はすべてデフォルトのままにします。

8 手順5で**完了**をクリックします。

9 **保存**をクリックします。

10 **アプリケーション > アプリケーションの種類 > WebSphereエンタープライズアプリケーション**に移動し、**qpp-server-12.0.ear**を再度クリックします。

11 **参照**から、**共有ライブラリ参照**をクリックします。

12 **qpp-server-12.0.ear**の横のボックスをチェックし、**共有ライブラリを参照する**をクリックします。

13 次の画面で、**新規**をクリックします。

14 任意の名前を選択します。classpathには、Platformパブリッシングおよび**ext**フォルダの相対パスを入力します。

たとえば、下記のディレクトリの場合

C:¥ProgramFiles¥IBM¥WebSphere¥AppServer¥profiles¥AppSrv01¥qpp¥publishing

./qpp/publishingおよび./qpp/extと入力します。

15 **適用**または**OK**をクリックして設定を保存します。

16 このライブラリ参照を**qpp-server-12.0.ear**アプリケーションに割り当てます。

17 **qpp-server-12.0.ear** Webアプリケーションを起動します。

18 リストに表示されるQPP EARをクリックし、**クラスの読み込みおよび更新の確認**を選択して、クラスの読み込み設定を変更します。

アプリケーション > アプリケーションの種類 > WebSphereエンタープライズアプリケーションへ移動して、下記の操作を行います。

- クラスローダーの順序を、**ローカルクラスローダーでロードしたクラスから（親が最後）**オプションへ変更します。
- WARクラスローダーポリシーを、**アプリケーション内の各WARファイルのクラスローダー**オプションへ変更します。

- **適用**をクリックし、変更をマスター設定へ保存します。
- QPP earを再起動します。

19 インストールを確認するには、次のURLに移動します。

http://localhost:9080/admin
http://localhost:9080/workspace

Oracle WebLogicへの配置

Quark Publishing Platform ServerをOracle WebLogicに配置するには、下記の手順に従ってください（サポートされているバージョンは12.1です）。

1 WebLogicサーバーを起動します。

2 「QPP_DOMAIN」という名前のWebLogicドメインを新規に作成します。

3 "krb5.conf"ファイルの場所を指定するJVMパラメータを設定するには、[WEB_LOGIC _DIR]¥domains¥base_domain¥bin¥setDomainEnv.cmdファイルを開き、下記の内容を最後に追加します。

set JAVA_OPTIONS=%JAVA_OPTIONS% -Djava.security.krb5.conf=./qpp/conf/krb5.conf

4 "For Deployment"フォルダから、"qpp"フォルダを対象のWebLogicサーバードメインディレクトリ（たとえば、C:¥oracle¥Middleware¥user_projects¥domains¥QPP_DOMAIN）にコピーします。

5 WebLogicコンピュータのグローバルシステム環境変数CLASSPATHを次の値に設定します。

[WEB_LOGIC _DIR]¥domains¥QPP_DOMAIN¥qpp¥publishing

および

[WEB_LOGIC _DIR]¥domains¥QPP_DOMAIN¥qpp¥ext

6 WebLogicを再起動します。

7 下記のURLから、Oracle WebLogicアプリケーションサーバーコンソールを開きます。

http://localhost:port/console

8 ドメイン（QPP_DOMAIN）を選択し、**Webアプリケーション**タブをクリックします。

9 **実パスのアーカイブを有効にする**をチェックします。

10 **保存**をクリックします。

11 **ドメイン構造**から、**配置**をクリックします。

12 右側の**配置**から、**インストール**をクリックし、"For Deployment"フォルダの.earファイルのパスを指定します （詳細は、「*EARファイルの配置準備*」を参照してください）。

13 次の画面で、デフォルトオプション（**アプリケーションとしてデプロイメントをインストール**）を選択して**次へ**をクリックします。

14 次の画面で、**次へ**をクリックします。

15 **完了**をクリックします。

16 **保存**をクリックします。

17 Quark Publishing Platform Web アプリケーションを起動します。

18 インストールを確認するには、次のURLに移動します。

http://localhost:7001/admin http://localhost:7001/workspace

外部Tomcatへの配置

Apache Software FoundationによってApache Tomcat™は、Java™サーブレットおよびJavaServer™ Pagesテクノロジーの標準リファレンス実装です。*Tomcat*™は、Webアプリケーション管理のためのサーブレットコンテナです。

Quark Publishing Platform Serverのスタンドアロンバージョンをインストールすると、Quark Publishing Platform Web ClientなどのQuark Publishing Platform Webアプリケーションを管理するため、Quark Publishing Platform Server Java Virtual Machine（JVM™）にTomcatのインスタンスが埋め込まれます。

しかし、他のWebアプリケーションのためにTomcatサーバーをすでに実行しており、Quark Publishing Platform Webアプリケーションに既存のTomcatサーバーを使用する場合には、TomcatのこのインスタンスにQuark Publishing Platform Serverを配置できます。Quark Publishing Platform Serverを既存の、すなわち外部のTomcatサーバーに配置するということは、サーバーコンピュータ上で別のQuark Publishing Platform Serverプロセスを実行する必要がないことを意味します。Quark Publishing Platform Serverを外部Tomcatに配置する場合、ソフトウェアパッケージの"Server (External Web Container)"にある別のビルドを使用できます。

要件とセットアップ

Quark Publishing Platform Serverには、JVM 1.5.xまたは1.6.x、およびWebサーバーとして設定されたApache Tomcat 7.0.41が必要です。 Tomcatが実行されているコンピュータは、64ビットコンピュータである必要があります。 Quark Publishing Platform Serverをすでにインストールされているに追加することもできます。

Windows上で実行するTomcatのセットアップ

➡ 下記の手順では、TOMCAT_HOMEが既存のApache Tomcatインストールフォルダです。

Tomcatのインストーラによって作成された、またはバイナリで展開された外部のTomcatインストール環境へQuark Publishing Platformを展開するには、下記の手順に従ってください。

1 [QPP_BUILD]/qppフォルダを[TOMCAT_HOME]フォルダへコピーします。

2 [QPP_BUILD]/webappsフォルダのコンテンツを[TOMCAT_HOME]/webappsフォルダへコピーします。

3 [TOMCAT_HOME]/endorsedフォルダがない場合は、このフォルダを作成します。

4 [QPP_BUILD]/endorsedフォルダのコンテンツを[TOMCAT_HOME]/endorsedフォルダへコピーします。

5 Tomcatがインストーラでインストールされた場合は、
[TOMCAT_HOME]/qpp/publishing/AS-Busdoc.xslt、
[TOMCAT_HOME]/qpp/publishing/BusDoc2QCD.xslt、
[TOMCAT_HOME]/qpp/publishing/BusDoc2QXPS.xslt、
[TOMCAT_HOME]/qpp/publishing/SmartDoc2QXPS.xsltの各ファイルで、以下に示すようにパスを更新します。

```
<xsl:include href="./qpp/publishing/xref-dita-anchors.xslt"/>
<xsl:include href="./qpp/publishing/AS-StyleSheets.xslt"/>
<xsl:include href="./qpp/publishing/AS-Transformations.xslt"/>
<xsl:include href="./qpp/publishing/BusDocsWordTableStyles.xslt"/>
```

6 Tomcatがバイナリで展開された場合は、
[TOMCAT_HOME]/qpp/publishing/AS-Busdoc.xslt、
[TOMCAT_HOME]/qpp/publishing/BusDoc2QCD.xslt、
[TOMCAT_HOME]/qpp/BusDoc2QXPS.xslt、
[TOMCAT_HOME]/qpp/publishing/SmartDoc2QXPS.xsltの各ファイルで、以下に示すようにパスを更新します。

```
<xsl:include href="../qpp/publishing/xref-dita-anchors.xslt"/>
<xsl:include href="../qpp/publishing/AS-StyleSheets.xslt"/>
<xsl:include href="../qpp/publishing/AS-Transformations.xslt"/>
<xsl:include href="../qpp/publishing/BusDocsWordTableStyles.xslt"/>
```

7 [TOMCAT_HOME]/qpp/conf/ServerApp.propertiesファイルで、以下のように設定します。

- webServer.portに、Tomcat用に設定した値、たとえば8080を入力します。
- webServer.embeddedWebContainerの値を**false**に設定します。

8 [TOMCAT_HOME]/qpp/conf/ManagerConfig.xmlファイルで、以下のように設定します。

- connectioninfoセクションのname要素に、QuarkXPress Serverの**IPアドレス**または**ホスト名**を入力します。
- port要素に、QuarkXPress Server用のポートを入力します。

➡ QuarkXPress ServerとTomcatが同じマシン上で実行されている場合、TomcatとQuarkXPress Serverは同じポート上で実行できないことに注意してください。

9 [TOMCAT_HOME]/qpp/conf/Qla.propertiesファイルで、以下のように設定します。

- 使用しているQLA Serverインスタンスの**ホスト名**、**ポート番号**、**シリアル番号**を設定します。

10 データベースを設定するため、[TOMCAT_HOME]/qpp/conf/Database.propertiesファイルで以下の値を編集または追加します。

- Oracleの場合

```
qpp.jdbc.driverClassName = oracle.jdbc.driver.OracleDriver
qpp.jdbc.url = jdbc:oracle:thin:@<hostname><:portnumber><:oracle_sid>
qpp.jdbc.userName = QppOracleDB
qpp.jdbc.password = QppPassword
```

- SQL Serverの場合

```
qpp.jdbc.driverClassName = com.microsoft.sqlserver.jdbc.SQLServerDriver
qpp.jdbc.url
=jdbc:sqlserver://<your-host-name>¥¥<instanceName>;databaseName=qppdb
qpp.jdbc.userName = QppMSSQLDB
qpp.jdbc.password = QppPassword
```

11 [TOMCAT_HOME]/qpp/conf/PluginsContext.xmlファイルを編集し、HSQL Daoコンテキストのデフォルトのエントリを、必要なデータベースDaoコンテキストに置き換えます。

- Oracleの場合

```
<import
resource="classpath:com/quark/qpp/common/dao/rdbms/oracle/OracleDaoContext.xml"/>
```

- SQL Serverの場合

```
<import
resource="classpath:com/quark/qpp/common/dao/rdbms/sqlserver/SqlServerDaoContext.xml"/>
```

12 マシンのグローバル環境変数を次のように設定します。

- **MAGICK_HOME**：[Tomcat_Home]¥qpp¥ImageMagick¥bin
- **MAGICK_FILTER_MODULE_PATH**：%MAGICK_HOME%/modules/filters
- **MAGICK_CODER_MODULE_PATH**：%MAGICK_HOME%/modules/coders

13 [TOMCAT_HOME]/conf/catalina.propertiesファイルを開き、以下の行を追加します。

```
Shared.Loader
.${catalina.home}/qpp/conf,${catalina.home}/qpp/lib/*.jar,
${catalina.home}/qpp/publishing, ${catalina.base}/qpp/ext
```

14 [TOMCAT_HOME]/conf/catalina.propertiesファイルを開いて org.apache.catalina.startup.TldConfig.jarsToSkip=を探し、次のように更新します。

```
org.apache.catalina.startup.TldConfig.jarsToSkip=a*.jar,b*.jar,c*.jar,d*.jar,e*.jar,f*.jar,g*.jar,
h*.jar,i*.jar,k*.jar,l*.jar,m*.jar,n*.jar,o*.jar,p*.jar,q*.jar,r*.jar,s*.jar,t*.jar,u*.jar,v*.jar,w*.jar,x*.
jar,y*.jar,z*.jar,ja*.jar,jc*.jar,jd*.jar,je*.jar,ji*.jar,jn*.jar
```

15 **Tomcatをインストーラでインストールした場合：**

Tomcatのモニタを起動します。**JAVA**タブで、Javaオプションの下のCATALINA_OPTSを以下のように設定します。

- Platform ServerがOracleデータベースで実行されている場合
  Doracle.jdbc.J2EE13Compliant=true
- LDAPの統合（または絶対パスの指定）
  Djava.security.krb5.conf=./qpp/conf/krb5.conf
- Java PermGenメモリスペース用：XX:MaxPermSize=256m

16 **Tomcatをバイナリで展開した場合：**

[TOMCAT_HOME]/bin/catalina.batファイルを開き、以下のパラメータを設定します。

- JRE_HOME=C:¥Program Files¥Java¥jdk1.7.0_76

- JAVA_OPTS=-server -Xmx2048m -XX:MaxPermSize=128m
- CATALINA_OPTS=-Doracle.jdbc.J2EE13Compliant=true -Djava.security.krb5.conf=../qpp/conf/krb5.conf

17 Tomcatのモニタを起動します。**JAVA**タブで、次のパラメータを設定します。

- Initial memory pool=1024 MB
- Maximum memory pool=2048 MB

Serverの起動と確認

Tomcatと共にQuark Publishing Platform Serverをインストールし、Quark Publishing Platform Serverアクセス用のポートを指定した後で、Quark Publishing Platform Serverを起動して設定を確認できます。 Quark Publishing Platform ServerとTomcatは連動しています。 Quark Publishing Platform Serverを起動および停止するには、Tomcatを起動および停止する必要があります。

Quark Publishing Platform Web Clientのアクセスを確認するには、ブラウザのアドレスフィールドにhttp://[マシン名]:[Web Serverポート]/workspaceと入力します。

Quark Publishing Platform Web adminのアクセスを確認するには、ブラウザのアドレスフィールドにhttp://[マシン名]:[Web Serverポート]/adminと入力します。

QuarkCopyDesk、QuarkXPress、Quark Publishing Platform Client、Quark Publishing Platform Web Client、およびQuark Publishing Platform Script Managerは、Tomcatポートを使用してQuark Publishing Platformサーバーにログオンします。

マルチサーバー展開でのQuark Publishing Platform Serverの展開

Quark Publishing Platformサーバークラスタは、共通データベース、共有アセットリポジトリ、およびメッセージキューを使用するために、Quark Publishing Platform Serverインストールの一部として設定されています。マルチサーバークラスタには次の利点があります。

- ハードウェアを追加することで、多数の要求を扱えます。
- アクティブロードバランシングを追加できます。
- 冗長的で信頼性のある設定が可能です。
- クラスタ内の特定のインスタンスの失敗により、全体のサービスが停止することはありません。インスタンスの失敗の影響を受けるのはごく一部のセッションに限られ、その後の要求は、利用可能でアクティブなQuark Publishing Platform Serverインスタンスにルーティングされます。
- ロードバランシングHTTPサーバーと通信するクライアントには透過的です。

要求をロードバランシングするには、HTTPサーバーと、"セッション維持"機能をサポートするHTTPロードバランサが必要です。以下のHTTPサーバーは、ロードバランシングサーバーとしてテスト済みです。

- Microsoft IIS 7と、IIS Tomcatコネクタの最新バージョン

- Apache 2.2と、Apache 2.2用JK Tomcat Connectorの最新バージョン
- Apache 2.2と、Apache 2.2に組み込みのmod_proxyおよびmod_proxy_balancer DSOモジュール

➡ HTTPロードバランサ経由でルーティングされた要求のみが、インスタンス間でロードバランシングされます。特定のインスタンスへの要求は、現在の負荷にかかわらず、必ず当該インスタンスで処理されます。

➡ RMIクライアントは、特定のサーバーインスタンスに直接接続する必要があります。

クラスタサーバーインスタンスまたはノードの設定

Quark Publishing Platform Serverはスタンドアロンのサーバーとして展開することも、初期コンピュータ上のデータベースを使用する外部Tomcatに展開することもでき、これはPlatformマルチサーバークラスタのノードと呼ばれます。クラスタサーバーをセットアップする最初の手順として、OracleまたはMS SQL Server用にQuark Publishing Platform Serverデータベースをセットアップします。

スタンドアロンサーバーとして展開されたPlatform Serverをノードとして動作させる

それぞれのQuark Publishing Platform Serverコンピュータで、下記の手順に従ってください。

1 qpp/confフォルダにある"ActiveMQ.xml"ファイルを開きます。

2 "ActiveMQ.xml"ファイルで、ネットワークコネクタのコメント化を解除して追加し、サーバーインスタンスのメッセージキューと、クラスタ内の1つ以上のサーバーインスタンスとを統合します。ここでqpp-node-Nは、Quark Publishing Platform Serverの別のインスタンスが展開されているコンピュータの名前です。

クラスタの各ノードについて、ノード自体を除いて、1つのコネクタが必要です。ここでは、3ノードのクラスタの例を示します。

ノード1には、次のエントリが含まれている必要があります。

```
<networkConnectors> <networkConnector dynamicOnly="true" duplex="false"
name="node1-1" networkTTL="1" uri="static:(tcp://qpp-Node2:61401)"/>
<networkConnector dynamicOnly="true" duplex="false" name="node1-2" networkTTL="1"
 uri="static:(tcp://qpp-node3:61401)"/> </networkConnectors>
```

ノード2には、次のエントリが含まれている必要があります。

```
<networkConnectors> <networkConnector dynamicOnly="true" duplex="false"
name="node2-1" networkTTL="1" uri="static:(tcp://qpp-Node1:61401)"/>
<networkConnector dynamicOnly="true" duplex="false" name="node2-2" networkTTL="1"
 uri="static:(tcp://qpp-node3:61401)"/> </networkConnectors>
```

ノード3には、次のエントリが含まれている必要があります。

```
<networkConnectors> <networkConnector dynamicOnly="true" duplex="false"
name="node3-1" networkTTL="1" uri="static:(tcp://qpp-Node1:61401)"/>
<networkConnector dynamicOnly="true" duplex="false" name="node3-2" networkTTL="1"
 uri="static:(tcp://qpp-node2:61401)"/> </networkConnectors>
```

➡ コネクタの名前（例：node1-1やnode1-2）は、クラスタ全体で固有な必要があります。

3 "ActiveMQ.xml"ファイルで、トランスポートコネクタのコメント化を解除し、サーバーインスタンスのメッセージキューを統合します。

次のようになります。

```
<transportConnector
uri="tcp://localhost:${jms.openWirePort}?wireFormat.maxInactivityDuration=0"
updateClusterClients="true" rebalanceClusterClients="true"
updateClusterClientsOnRemove="true"/> <transportConnector
uri="tcp://${server.machinename}:${jms.openWirePort}?wireFormat.maxInactivityDuration=0"/>
```

4 各WebアプリのWEB-INFフォルダにある、ワークスペースおよび管理Webアプリの"web.xml"ファイルを開き、SessionResetFilterのコメント化を解除して、ノードに障害が発生した場合にセッションを処理できるようにします。

```
<filter> <filter-name>SessionResetFilter</filter-name>
<filter-class>com.quark.web.security.servlet.SessionResetFilter</filter-class> <init-param>
<param-name>jvmRoute</param-name> <param-value>qpp1</param-value> </init-param>
 <init-param> <param-name>sessionCookieName</param-name>
<param-value>JSESSIONID</param-value> </init-param> <init-param>
<param-name>sessionInitUrls</param-name> <param-value> /workspace/reconnectUser.qsp
 /workspace/login.qsp </param-value> </init-param> </filter> <filter-mapping>
<filter-name>SessionResetFilter</filter-name> <url-pattern>/*</url-pattern>
</filter-mapping>
```

5 [QPP Server]/confフォルダにある"ServerApp.properties"ファイルで、server.id.prefixに固有の値を割り当て、サーバーインスタンスを識別します。

6 [QPP Server]/confフォルダにある"ServerApp.properties"ファイルで、マシンのIPを"server.machinename"に割り当て、サーバーインスタンスを識別します。

7 [server-path]/confフォルダにある"server.xml"ファイルで、<Engine>タグのjvmRoute属性に固有の値を割り当てます。

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="node1">
```

➡ jvmRouteは、"web.xml"ファイル（手順4で編集するファイル）と"server.xml"ファイルの両方で同じ値の必要があります。

8 {TOMCAT_HOME}/webapps/webServices/WEB-INFにある"services"フォルダの名前を"services_old"に変更します。

9 {TOMCAT_HOME}/webapps/webServices/WEB-INFにある"serviceArchivesMultiServer"フォルダの名前を"services"に変更します。

10 サーバーのインストールフォルダにある"ServerStartup.bat"ファイルに、-Dspring.profiles.active=multiserverを追加し、サーバーをクラスタモードで実行します。

```
java -Xmx2048m -XX:MaxPermSize=256m -classpath "%JavaClassPath%"
-Dspring.profiles.active=multiserver -Doracle.jdbc.J2EE13Compliant=true
-Djava.security.krb5.conf=conf/krb5.conf -Djava.endorsed.dirs="endorsed"
-Djavax.xml.stream.XMLInputFactory=com.ctc.wstx.stax.WstxInputFactory
com.quark.qpp.Server PluginsContext.xml
```

11 サーバーのインストールフォルダにある"wrapper.conf"ファイルで、Java Additional Parametersセクションの下にある次の行をコメント化解除します。

```
wrapper.java.additional.7=-Dspring.profiles.active=multiserver
```

➡ Quark Publishing Platformのすべてのノードについて、上記の手順を繰り返します。

12 作業を開始する前に、使用しているブラウザのCookieをクリアします。

13 次のセクション「*サーバーのキャッシュフラッシュ機構の設定*」に記載されている手順に従い、すべてのサーバーインスタンスにわたってキャッシュをフラッシュします。

サーバーのキャッシュフラッシュ機構の設定

Platform Serverでは、アップデートが行われたときに、サーバーのすべてのインスタンスでキャッシュをフラッシュするため、2つの機構が用意されています。

- 通知ベース - この機構では、キャッシュが非同期にクリアされます。通知ベースのキャッシュフラッシュは、タイプがCACHEでFLUSH_INTERCEPTORプロパティを持つメッセージを発行することで行われます。これは、デフォルトで設定されている機構です。
- REST要求ベース - この機構では、キャッシュが同期してクリアされます。RESTベースの機構では、すべてのピアノードへHTTP要求を行い、キャッシュフラッシュをトリガします。
  http://{peerNode}:61400/rest/service/xcache/flush/{interceptorName}この機構はデフォルトの設定ではありませんが、必要に応じて設定できます。

RESTベースのキャッシュフラッシュ機構を設定するには、下記の変更を行ってください。

1. {install_path}/Server/lib/qpp-server-common-{version}.jarに含まれている"CacheManager.xml"ファイルを開きます。
2. asyncRemoteMessengerをコメント化し、synchronousRemoteInvokerをコメント解除します。このBeanを、asyncRemoteMessengerの代わりにbaseCacheFlushingInterceptorへ挿入します。

```
<bean id="synchronousRemoteInvoker"
    class="com.quark.qpp.common.caching.SynchronousRemoteInvoker">
  <property name="remoteHosts" value="${server.peer.url}"/>
  <property name="httpContext" value="${server.cache.remoteFlush.context}"/>
</bean>

<!-- <bean id="asyncRemoteMessenger"
class="com.quark.qpp.core.messaging.service.impl.AsyncRemoteMessenger">
  <property name="messagePublisher" ref="messagePublisher"/>
</bean> -->
```

3. baseCacheFlushingInterceptorを見つけ、flushinginterceptorにsynchronousRemoteInvoker Beanを挿入します。

```
<bean id="baseCacheFlushingInterceptor"
class="com.quark.qpp.common.caching.BaseCacheFlushingAdvice" abstract="true">
  <property name="remoteInvoker" ref="synchronousRemoteInvoker"/>
</bean>
```

4. ファイルを保存して閉じます。
5. [QPP Server]/confフォルダにある"ServerApp.properties"ファイルで、以下の行をコメント解除します。

```
# QPPクラスタを同期キャッシュアップデートに設定するには、以下のプロパティをコメント解除します。
#クラスタの一部であり、ノード自体を除くピアサーバーインスタンスのURIを、カンマで区切って記載します。
server.peer.url=http://{peerHostN 1}:61400,http://{peerHostN 2}:61400

# キャッシュのフラッシュを起動するため使用可能なHTTPコンテキスト
server.cache.remoteFlush.context=rest/service/xcache/flush
```

6. それぞれのノードについて、[QPP Server]/webapps/rest/WEB-INFフォルダにある"rest-servlet.xml"ファイルを開き、以下の行をコメント解除します。

```
<!-- Following code should be uncommented for synchronous REST based cache flushing
 in clustered environment.
```

```
    This enables "RemoteFlushController" which handles HTTP requests from remote peers
 for flushing the cache.–>
<context:component-scan base-package="com.quark.qpp.common.caching"/>
```

7 サーバーを再起動します。

外部Tomcatとして展開されているPlatform Server

それぞれのQuark Publishing Platform Serverコンピュータで、下記の手順に従ってください。

1 最初のコンピュータのデータベースを使用して、外部のTomcatにQuark Publishing Platform Serverのスタンドアロンインスタンスを展開します。

2 qpp/confフォルダにある"ActiveMQ.xml"ファイルを開き、ネットワークコネクタを追加して、サーバーインスタンスのメッセージキューと、クラスタ内の1つ以上のサーバーインスタンスとを統合します。ここでqpp-node-Nは、Quark Publishing Platform Serverの別のインスタンスが展開されているコンピュータの名前です。

➡ これらのネットワークコネクタを、次のセクション「*スタンドアロンサーバーとして展開されたPlatform Serverをノードとして動作させる*」の手順2と同じ方法で追加します。

3 "ActiveMQ.xml"ファイル内のtransport connectorをコメント解除し、サーバーインスタンスのメッセージキューをクラスタ内のサーバーインスタンスのIPアドレスと名前に統合します。

```
<transportConnector
uri="tcp://localhost:${jms.openWirePort}?wireFormat.maxInactivityDuration=0"
updateClusterClients="true" rebalanceClusterClients="true"
updateClusterClientsOnRemove="true"/> <transportConnector
uri="tcp://${server.machinename}:${jms.openWirePort}?wireFormat.maxInactivityDuration=0"/>
```

4 [QPP Server]/confフォルダにある"ServerApp.properties"ファイルで、server.id.prefixに固有の値を割り当て、サーバーインスタンスを識別します。

5 [QPP Server]/confフォルダにある"ServerApp.properties"ファイルで、マシンのIPをserver.machinename属性に割り当て、サーバーインスタンスを識別します。

6 {TOMCAT_HOME}/confフォルダにある"server.xml"ファイルで、<Engine>タグのjvmRoute属性に固有の値を割り当てます。

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="node1">
```

7 tomcat_home/conf/context.xmlファイルを下記のように編集します。

```
<Context sessionCookiePath="/">
```

8 {TOMCAT_HOME}/webapps/webServices/WEB-INFにある"serviceArchivesMultiServer"フォルダの名前を"services"に変更します。

9 {TOMCAT_HOME}/binフォルダにある"Catalina.bat"ファイルを編集します。次のランタイムパラメータを設定し、サーバーをマルチサーバーモードで実行します。

```
Set CATALINA_OPTS=-Dspring.profiles.active=multiserver
```

➡ apache tomcatが"service"として実行されている場合、Apache tomcatモニタで、**Java**タブのJava optionテキストフィールドに、次のパラメータを追加します。
-Dspring.profiles.active=multiserver

10 各Webアプリの{TOMCAT_HOME}/webapps/ WEB-INFフォルダにある、ワークスペースおよび管理Webアプリの"web.xml"ファイルを開き、次のようにSessionResetFilterのコメント化を解除して、ノードに障害が発生した場合にセッションを処理できるようにします。

```
<filter> <filter-name>SessionResetFilter</filter-name>
<filter-class>com.quark.web.security.servlet.SessionResetFilter</filter-class> <init-param>
<param-name>jvmRoute</param-name> <param-value>qpp1</param-value> </init-param>
 <init-param> <param-name>sessionCookieName</param-name>
<param-value>JSESSIONID</param-value> </init-param> <init-param>
<param-name>sessionInitUrls</param-name> <param-value> /workspace/reconnectUser.qsp
 /workspace/login.qsp </param-value> </init-param> </filter> <filter-mapping>
<filter-name>SessionResetFilter</filter-name> <url-pattern>/*</url-pattern>
</filter-mapping>
```

➡ <param-value>qpp1</param-value>の値は、手順6の"server.xml"ファイルと同じにする必要があります。

11 Quark Publishing Platformのすべてのノードについて、上記の手順を繰り返します。

IISをHTTPロードバランサとして設定

IIS 7をHTTPロードバランサとして設定するには、下記の手順に従ってください。

1 最初にTomcatコネクタを配置します。方法の詳細については、*http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html*を参照してください。「TomcatConnector」という名前のフォルダを作成し、IISビルド用Tomcat Connectorから"isapi_redirect.dll"ファイルを配置します。

2 同じフォルダに"isapi_redirect.properties"ファイルを作成し、そのファイルに下記の内容を追加します。

```
xtension_uri=/jakarta/isapi_redirect.dll
log_file=C:¥TomcatConnector¥Log¥isapi.log
# "isapi_redirect.properties"ファイルにリストされたディレクトリが
# 存在することを確認してください。
log_level=info
worker_file=C:¥TomcatConnector¥worker.properties
worker_mount_file=C:¥TomcatConnector¥uriworker.properties
```

3 同じフォルダに"workers.properties"という名前の新しいファイルを作成し、次の内容を追加します。qpp-node1およびqpp-node2は、各Quark Publishing Platform Serverの"server.xml"ファイルで指定したjvmRoute属性の値です。

```
worker.list=TomcatBalancer
worker.TomcatBalancer.type=lb
worker.TomcatBalancer.balance_workers=qpp-node1,qpp-node2
worker.TomcatBalancer.sticky_session=True
worker.TomcatBalancer.sticky_session_force=True
worker.TomcatBalancer.method=Request
worker.TomcatBalancer.lock=Pessimistic

worker.qpp-node1.type=ajp13
worker.qpp-node1.host=qpp-node1

# Quark Publishing Platformが配置されたTomcatサーバーの
# "server.xml"ファイルに定義されたAJPコネクタの
# ポートと同じである必要があります。
worker.qpp-node1.port=61398

worker.qpp-node1.lbfactor=3

worker.qpp-node2.type=ajp13
worker.qpp-node2.host=qpp-node2

# Quark Publishing Platformが配置されたTomcatサーバーの
# "server.xml"ファイルに定義されたAJPコネクタの
# ポートと同じである必要があります。
worker.qpp-node2.port=61398
```

```
worker.qpp-node2.lbfactor=3
```

4 同じフォルダに"uriworker.properties"ファイルを作成し、そのファイルに下記の内容を追加します。

```
/admin/*=TomcatBalancer
/admin=TomcatBalancer
/workspace/*=TomcatBalancer
/workspace=TomcatBalancer
/webservices/*=TomcatBalancer
/webservices=TomcatBalancer
/rest/*=TomcatBalancer
/rest=TomcatBalancer
/messaging/*=TomcatBalancer
/messaging=TomcatBalancer
/qxpsmadmin/*=TomcatBalancer
/qxpsmadmin=TomcatBalancer
/favicon.ico=TomcatBalancer
/*=TomcatBalancer
/qxpsm/*=TomcatBalancer
/qxpsm=TomcatBalancer
/pluginwiris_engine/*=TomcatBalancer
/pluginwiris_engine=TomcatBalancer
```

➡ このファイルには、IISで処理して各Quark Publishing Platform Serverに転送する必要のあるURLマッピングが含まれます。

5 IIS管理コンソールを使用して、"isapi_redirect.dll"ファイルを配置したディレクトリの物理パスを使用して、「jakarta」という名前の仮想ディレクトリをIISウェブサイトに追加します。

6 仮想ディレクトリに実行権限を付与します。仮想フォルダを選択します。**ハンドラマッピング**をダブルクリックし、**アクション**ペインで**機能のアクセス許可の編集**をクリックします。**機能のアクセス許可の編集**ダイアログボックスで**実行**をチェックし、**OK**をクリックします。

7 ISAPIフィルタをIISウェブサイトに追加します。ウェブサイトを選択します。**ISAPIフィルタ**をダブルクリックし、**アクション**ペインで**追加**をクリックします。**ISAPIフィルタの追加**ダイアログボックスで、名前と"isapi_redirect.dll"ファイルのパスを入力し、**OK**をクリックします。

8 ISAPIおよびCGIの制限機能を設定します。サーバーホーム画面に移動します。**ISAPIおよびCGIの制限**をダブルクリックし、**アクション**ペインで**追加**をクリックします。**ISAPIまたはCGIの制限の追加**ダイアログボックスで、名前と"isapi_redirect.dll"ファイルのパスを入力し、**拡張パスの実行を許可する**をチェックして**OK**をクリックします。

9 Windowsn認証を有効化します。Webサイトに移動し、**認証**をダブルクリックします。**Windows認証**を右クリックし、**有効**を選択します。他の認証オプションを無効にしてIISを再起動します。

10 インストールを確認するには、すべてのノードのQuark Publishing Platform Serverを起動し、コンピュータ名またはIISサーバーのIPアドレスとポート番号を使用して各サーバーにアクセスします。IISサーバーは、Tomcatコネクタを使用して要求を処理およびロードバランスします。"isapi_redirect.properties"で識別されるログファイルで詳細を確認できます。

Apache Web ServerをHTTPロードバランサとして設定

Apache Web Server 2.2（Apache 2.2用JK Tomcat Connector最新版使用）をHTTPロードバランサとして設定するには、下記の手順に従ってください。

1 最初にTomcatコネクタを配置します。方法の詳細については、*http://tomcat.apache.org/connectors-doc/miscellaneous/faq.html*を参照してください。"mod_jk.so"ファイルを、Tomcatコネクタのダウンロード場所から{Apache_2.2_installation}/modulesへコピーします。

2 テキストエディタで{Apache_2.2_installation}/conf/httpd.confファイルを開き、下記の内容を追加します。

```
LoadModule jk_module modules/mod_jk.so

<IfModule jk_module>
  JkWorkersFile conf/workers.properties
     JkLogFile logs/mod_jk.log
 JkLogStampFormat "[%H:%M:%S] "
 JkRequestLogFormat "%T"
 JkLogLevel error
     JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
  <Directory />
  AllowOverride All
  <Limit GET HEAD POST PUT DELETE OPTIONS>
      Order Allow,Deny
      Allow from all
  </Limit>
 </Directory>
 JkMount /workspace TomcatBalancer
 JkMount /workspace/* TomcatBalancer
 JkMount /webservices TomcatBalancer
 JkMount /webservices/* TomcatBalancer
 JkMount /admin TomcatBalancer
 JkMount /admin/* TomcatBalancer
 JkMount /rest TomcatBalancer
 JkMount /rest/* TomcatBalancer
 JkMount /messaging TomcatBalancer
 JkMount /messaging/* TomcatBalancer
 JkMount /qxpsm TomcatBalancer
 JkMount /qxpsm/* TomcatBalancer
 JkMount /qxpsmadmin TomcatBalancer
 JkMount /qxpsmadmin/* TomcatBalancer
 JkMount /pluginwiris_engine/*=TomcatBalancer
 JkMount /pluginwiris_engine=TomcatBalancer
 </IfModule>
```

3 {Apache_2.2_installation}/conf/workers.propertiesファイルを作成し、下記の内容を追加します。qpp-node1およびqpp-node2は、各Quark Publishing Platform Serverの"server.xml"ファイルで指定したjvmRoute属性の値です。

```
worker.list=TomcatBalancer

worker.TomcatBalancer.type=lb
worker.TomcatBalancer.balance_workers=qpp-node1, qpp-node2
worker.TomcatBalancer.sticky_session=True

# この行はコメント化すべきですが、そうしない場合、ロードバランサとして機能する
# Apache Webサーバーへ接続できません。
worker.TomcatBalancer.sticky_session_force=True

worker.TomcatBalancer.method=Request
worker.TomcatBalancer.lock=Pessimistic

worker.qpp-node1.type=ajp13

# 最初のQPPノードのIPアドレス
worker.qpp-node1.host=Server 1

# 8009は外部のtomcatサーバー用
# 61398はQPPの組み込みtomcat用
worker.qpp-node1.port=61398

worker.qpp-node1.lbfactor=3

worker.qpp-node2.type=ajp13

# 最初のQPPノードのIPアドレス
worker.qpp-node2.host=Server 2

# 8009は外部のtomcatサーバー用
# 61398はQPPの組み込みtomcat用
```

worker.qpp-node2.port=61398

worker.qpp-node2.lbfactor=3

4 Apache 2.2 httpdを再起動します。

5 インストールを確認するには、すべてのノードのQuark Publishing Platform Serverを起動し、コンピュータ名またはApacheサーバーのIPアドレスとポート番号を使用して各サーバーにアクセスします。Apacheサーバーは、Tomcatコネクタを使用して要求を処理およびロードバランスします。{Apache_2.2_Installation}/logsにあるログファイルから詳細を確認できます。

Elasticsearch

複数のサーバーへの配置を行う場合、Elasticsearchを活用して、Platformのマルチサーバークラスタの各アプリケーションノードについて検索インデックスを再生成することを避けられます。

Elasticsearchのダウンロードとインストール

1 次の場所からElasticsearch v2.4.0をダウンロードし、解凍します。

www.elastic.co/downloads/elasticsearch

2 Elasticsearchをインストールするには、下記の操作を行ってください。

- Unixの場合：bin/elasticsearchを実行します。
- Windowsの場合：**bin¥elasticsearch.bat**を実行します。
- curl -X GET http://localhost:9200/を実行します。

Configuring Platform server to leverage Elasticsearch

A Multi-server deployment can leverage Elasticsearch to avoid regenerating the search index for each application node of the Platform multiserver cluster.

➡ If you are deploying Platform server as a single instance with Elasticsearch, there is no need to make any configuration changes as that deployment is covered by the default configuration.

Perform the following steps on each Quark Publishing Platform Server computer:

1 Open the "ESIndexingConfig.properties" file located in the [QPP Server]/conf folder.

2 Specify where Elasticsearch is running:

#Comma delimited list of host:tcp_port entries pointing to Elasticsearch cluster nodes
es.cluster.nodes=platform2k12:9300

➡ The default port for Elasticsearch is 9300.

3 Specify the name of the index to be used for attributes:

es.attribute.index.name=qps-attribute-index

4 Specify the name of the index to be used for text:

```
es.text.index.name=qps-text-index
```

5 In the "ESAttributeIndexSettings.json" file located in the [QPP Server]/conf folder, specify the number of shards and replicas required for the index:

```
{ "index":
  {"number of shards":5,
   "number of replicas":1
   }
...
...
...
}
```

➡ Shards are per index and replicas are per node.

6 In the "PluginsContext.xml" file located in the [QPP Server]/conf folder, comment out the following line to specify that you will no longer be using the Lucene indexer:

```
<!--import
resource="classpath:com/quark/qpp/textindexing/adapter/impl/LuceneTextIndexerContext.xml"/>-->
```

複数サーバー環境用にElasticsearchを設定する

複数サーバーの環境に展開し、各サーバーに独自のElasticsearchインスタンスが実行される場合、クラスタはマスターと子の概念で実行されます。Elasticsearchのインスタンスの1つがマスターとなり（どのノードが最初に展開されたかに基づいて決定されます）、他のインスタンスはすべて子インスタンスとなります。マスターインスタンスが終了すると、子インスタンスの1つがマスターになります。

各Elasticsearchインスタンスをクラスタの一部として設定するには、Elasticsearchのそれぞれの展開で、下記の手順を実行してください。

1 {Elasticsearch_install_path}/configフォルダにある"elasticsearch.yml"ファイルを開きます。

2 クラスタ名を指定します。

```
cluster.name: myESclustername
```

3 このノードの名前を指定します。

```
node.name: nameofnode1
```

4 ネットワークホスト名を指定します。

```
network.host: mynetworkhost
```

5 クラスタの他のノード（Elasticsearchのインスタンス）を識別します。

```
discovery.zen.ping.unicast.hosts:["nameofnode2", "nameofnode3"]
```

6 それぞれのQuark Publishing Platform Serverコンピュータで、[QPP Server]/conf フォルダにある"ESIndexingConfig.properties"ファイルに、クラスタの名前を指定します。

```
# このインスタンスのElasticsearchクライアントが接続する先のElasticsearchクラスタ名
es.cluster.name=myESclustername
```

Elastisticsearchの実行

Elasticsearchの実行には、Java 7 Update 55またはそれ以降が必要です。環境変数JAVA_HOMEを適切に設定する必要があります。

Elasticsearchは、次のいずれかの方法で開始できます。

- バッチファイルの使用
  Elasticsearchをフォアグラウンドで、新しいプロセスとして実行するには、/binフォルダにある"elasticsearch.bat"を実行します。

- サービスとして実行
  Elasticsearchをフォアグラウンドで、新しいプロセスとして実行するには、/binフォルダにある"elasticsearch.bat"を実行します。

  Elasticsearchをサービスとして実行するには、下記のコマンドを使用して、/binフォルダにある"service.bat"を実行します。

  - 下記のコマンドを実行して、Elasticsearchをサービスとしてインストールします。C:¥elasticsearch-2.4.0¥bin>service.bat install
  - インストール後に下記のコマンドを実行して、Elasticsearchをサービスとして開始します。C:¥elasticsearch-2.4.0¥bin>service.bat start
  - 下記のコマンドを実行して、Elasticsearchサービスを停止します。C:¥elasticsearch-2.4.0¥bin>service.bat stop
  - 下記のコマンドを実行して、インストールされているElasticsearchサービスを削除します。サービスが開始されていれば、停止されます。C:¥elasticsearch-2.4.0¥bin>service.bat remove
  - 下記のコマンドを実行して、インストールされているサービスを管理するGUIを開始します。C:¥elasticsearch-2.4.0¥bin>service.bat manager

➡ 詳細については、*www.elastic.co/guide/en/elasticsearch/reference/current/setup-service-win.html#setup-servic*覧ください。

ElasticsearchがPlatformとともに実行されるよう設定する方法については、『**Quark Publishing Platform Sys Admin Guide**』を参照してください。

Elasticsearch用インターフェイスの提供

Elasticsearchはユーザーインターフェイスを提供しないため、Quarkはサードパーティーのプラグインを使用しています。

プラグインを利用するには、Elasticsearchのそれぞれの展開で、下記の手順を実行してください。

1 headプラグインを{Elasticsearch_install_path}/pluginsフォルダへコピーアンドペーストします。

2 次の場所で、ユーザーインターフェイスにアクセスします。

{network host}:9300/_plugin/head/

Quark Publishing Platform Server用のSecure Sockets Layer（SSL）の有効化

Quark Publishing Platformに異なるセキュリティオプションを設定できます。お使いのネットワークセキュリティ仕様に加えて、Quark Publishing Platformクライアントアプリケーションに SSLプロトコルを適用できます。

SSLのサポート

アプリケーションサーバーコンテナと、すべてのQuark Publishing Platform Clientが、SSLテクノロジーを使用して、保護モードで実行するよう設定できます。本セクションでは、設定手順について説明します。

JVMにTomcatを埋め込まずにQuark Publishing Platformを実行することもできます。Tomcatを埋め込まずにQuark Publishing Platformを設定する方法については、「*外部Tomcatへの配置*」を参照してください。

Quark Publishing Platform Serverは、Quark Publishing Platform環境でWebアプリケーションを管理するため、JVMにApache Tomcat 7.0.61のインスタンスを埋め込みます。Quark Publishing Platformには、Quark Publishing Platform Web Client、Quark Publishing Platform Console、Quark Publishing Platform Renderer Manager、およびQuark Publishing Platform Web Servicesの4つのアプリケーションがあります。

SSLを有効にすると、Quark Publishing Platform Serverに配置されたすべてのQuark Publishing Platformクライアントアプリケーションに SSLが適用されます。

Quark Publishing Platform ServerでのSSLの有効化

下記の手順では、2つのシナリオについて説明しています。編集する"server.xml"ファイルには両方のシナリオ向けのXMLタグが含まれており、特定のタグを「コメント化」または「コメント化解除」することによって有効または無効にします。

すべてのQuark Publishing Platform WebアプリケーションのHTTPを保護するためにSSLを有効にするには、下記の手順に従ってください。

1 {install_path}/confにある"server.xml"ファイルを開きます。

2 下記のタグのコメント化します。

```
<Connector port="61400" maxHttpHeaderSize="8192"maxThreads="150"
minSpareThreads="25" maxSpareThreads="75"enableLookups="false" redirectPort="61399"
 acceptCount="100"connectionTimeout="20000" disableUploadTimeout="true" />
```

3 下記のタグのコメント化を解除します。

```
<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 61399を61400（またはセキュリティ保護された接続のためにTomcatがリッスンするポート）に置き換えます。

5 "server.xml"を保存して閉じます。

6 Quark Publishing Platform Serverマシンで、コマンドプロンプトを開き、次のコマンドを実行します。

```
%JAVA_HOME%¥bin¥keytool -genkey -alias tomcat -keyalg RSA
```

7 プロンプトで詳細を入力します。パスワードにはchangeitを指定します。

8 Quark Publishing Platform Serverを再起動します。

9 管理Webページhttps://servername:61399/admin61399にアクセスします。

➡ この設定では、プライベートとパブリックのキーのペアを持つ、1つのキーストアが作成されます。これは自己署名の証明書です。

10 SSL設定の詳細については、証明機関の使用法も含め、次のページにあるApache Tomcat SSLの情報を参照してください。
url:https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html

➡ この変更によって、Quark Publishing Platformクライアントアプリケーションで HTTPSを使用できるようになります。たとえば、Quark Publishing Platform Web Clientユーザーの URLは、https://[サーバー名]:61399/workspaceとなります。

SSL対応サーバーへログオンするためのPlatform Clientの構成

Mac OS X版のQuarkXPressとQuarkCopyDeskのユーザーは、ログオンの前に、Quark Publishing Platform ServerからSSL証明書を取得する必要があります。このためには、各ユーザーがターミナルを起動して下記のコマンドを実行し、[サーバー名]に対するQuark Publishing Platform Serverコンピュータの IPアドレスを置き換える必要があります。

```
echo ¦ openssl s_client -connect[サーバー名]:443 > [サーバー名].pem
```

El-CapitanおよびSierra MAC OSでは、コマンドは次のようになります。

```
echo ¦ openssl s_client -connect[サーバー名]:[Port] -servername[サーバー名] > [server
name.pem
```

このコマンドによって、"[サーバー名].pem"という名前のサーバー証明書のコピーが取得できます。このファイルを~/Library/Application Support/Quark/QPP/Certificatesフォルダに保存します（またはカスタマイズしたplistファイルが、~/Library/Application Support/Quark/QPP/[QPPフレームワークのバージョン]に存在する場合は、ファイルをそのフォルダに保存します）。

- 上記の場所に"Quark Publishing Platform"フォルダと"Certificates"フォルダがまだ存在しない場合は、手動で作成します。

- Windows上で実行されているQuark Publishing Platform Clientでは、サーバーベースのSSL証明書は必要ありません。

SSLの確認と使用

SSLを確認し、使用するには、下記の手順に従ってください。

1. Quark Publishing Platform Serverを開始します。
2. 以下の内容を入力してQuark Publishing Platform Web Clientアクセスをテストします。https://[マシンIPまたは名前]:61399/workspace.

Quark Publishing Platform Server — 手動設定

Quark Publishing Platform Serverをインストールした後、デフォルト設定を変更できます。Quark Publishing Platform Serverの実行中にJConsoleでパラメータを設定できるのに加えて、さまざまな.xmlファイルと.propertiesファイルの設定を調整できます。JVM設定のメモリ割り当ても調整できます。

"ServerApp.properties"の編集

"ServerApp.properties"を編集するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダ内の"ServerApp.properties"ファイルを開きます。
2. rmi.portの値に、RMIレジストリがリッスンするポートの番号を設定します。ScriptManagerなどのJavaベースのRMIクライアントは、このポートを通じて接続します。
3. rmi.servicePortの値に、Quark Publishing Platformサービスオブジェクトが登録されているRMIサーバーで使用するポート番号を設定します。
4. namingservice.portの値に、CORBAネーミングサービスでオブジェクトの解決要求をリッスンするために使用するポート番号を設定します。
5. serverORB.portの値に、Quark Publishing Platformサービスオブジェクトがアクティブ化されているORBで使用するポート番号を設定します。
6. jms.openWirePortの値に、OpenWireプロトコルを通じたJMS通信のために開くポート番号を設定します。Quark Publishing Platform Script ManagerなどのJavaクライアントは、このポートに接続して、サーバー通知をリッスンします。
7. webServer.portの値に、HTTP接続のためにTomcatがリッスンするポート番号を設定します。Quark Publishing Platform Script Manfager Web ClientとSOAPクライアントはこのポートを通じて接続されます。この値には、Tomcatの"server.xml"ファイルでHTTPコネクタに対して指定されている値を設定する必要があります。
8. socketStreaming.portの値に、ファイル転送（アップロード／ダウンロード）に使用するポート番号を設定します。
9. 特定のIPアドレスにQuark Publishing Platformをバインドするには、server.machinenameの値にそのIPアドレスを設定し、server.bindtoip=trueとして設定します。または、ネットワークカードが複数あり、Quark Publishing Platformをそのすべてのアドレスにバインドする場合は、server.machinename=localhost、

server.bindtoip=falseおよび
server.addtionalnames=[non-default-ip1],[non-default-ip2]のように設定します。

10 server.additionalnamesの値を設定して、ファイアウォールが置かれているグローバルIPアドレスを指定します。

11 webServer.port値にTomcatに設定した値、たとえば8080などを入力します。

12 ディレクトリサーバーが稼働していない場合でもユーザーがQuark Publishing Platform Serverにログオンできるようにするには、authentication.external.cacheTicket = trueのように設定します。

13 パスワードの大文字と小文字の区別を設定するには、server.password.case.sensitiveの値を、パスワードの比較で大文字と小文字を区別しないならfalseに設定します。LDAPを外部認証に使用する場合、このオプションは有効ではありません。

14 セッションタイムアウトを設定するには、session.maxIdleの値にセッションタイムアウトの時間を秒単位で指定し、session.eviction.thread.delayの値にセッション追い出しスレッドの周期を秒単位で指定します。

15 リポジトリ状況アップデータはバックグラウンドスレッドで、指定された周期で実行され、基礎となるリポジトリがアクティブかをテストし、サーバーデータベースで状況を同じに更新します。リポジトリ状況アップデータのスリープ周期を設定するには、respository.status.updator.sleepIntervalの値に、リポジトリ状況アップデータスレッドが実行されるまでの間隔を秒単位で指定します。

16 QuarkXPress ServerにRealm verif.For Admin. Requestsが設定されている場合、QuarkXPress Serverのユーザー名とパスワードをこのファイルに設定する必要があります。QuarkXPress Serverを設定するには、下記のプロパティを設定します。

- QuarkXPress Serverのユーザー名を指定するには、qxps.usernameの値を設定します。
- QuarkXPress Serverのパスワードを指定するには、qxps.passwordの値を設定します。
- QuarkXPress ServerXPSの場所を指定するには、qxps.localeの値を設定します。

"PublishingPool.properties"の編集

"PublishingPool.properties"を編集するには、下記の手順に従ってください。

1 Serverのインストールに含まれるconfフォルダ内の"PublishingPool.properties"ファイルを開きます。

2 publishingThread.pool.maxActiveの値を、同時に実行できるバックグラウンドのパブリッシングスレッドの最大数に設定します。

3 publishingThread.pool.maxIdleの値を、プール内のアイドルスレッドの最大数に指定します。

4 publishingThread.pool.minIdleの値を、プール内のアイドルスレッドの最小数に指定します。

5 publishingThread.pool.maxWaitの値に、パブリッシング要求がプールからスレッドを借りるときに待つ時間を、ミリ秒単位で指定します。

6 publishingThread.pool.minEvictableIdleTimeMillisの値に、プール内のスレッドが排除されるまでの時間を、ミリ秒単位で指定します。

7 publishingThread.pool.timeBetweenEvictionRunsMillisの値に、排除実行スレッドがアイドルスレッドを取り除くまでの時間を、ミリ秒単位で指定します。

Quark Publishing Platform RendererをQuark Publishing Platformで使用するための設定

Quark Publishing PlatformでQuark Publishing Platform Rendererを使用する設定は、下記の手順に従ってください。

1 ウェブブラウザで下記のURLに移動してQXPSM Admin Clientを開きます。http://[QPPサーバー名]:[ポート]/qxpsmadmin

2 **サーバーの管理**ウィンドウで、**サーバーの追加**をクリックし、Quark Publishing Platform RendererのQuark Publishing Platformインスタンスを追加します。詳細は、『**A Guide to Quark Publishing Platform Renderer**』を参照してください。

"Qla.properties"の編集

"Qla.properties"を編集するには、下記の手順に従ってください。

1 [QPP Server]/confフォルダ内の"Qla.properties"ファイルを開きます。

2 QlaServer.machinename=フィールドにQLA ServerのIPアドレスとホスト名を入力します。

3 QlaServer.port=フィールドにQLA Serverのポート番号を入力します。

4 バックアップQLA Serverがある場合には、Backup.QlaServer.machinename=フィールドとBackup.QlaServer.port=フィールドにIPアドレスまたはホスト名とポート番号を入力します。

5 Qla.SerialNumber=フィールドにQuark Publishing Platformシリアル番号を入力します。

➡ Quark Publishing Platformシリアル番号は、QLA Server ConsoleアプリケーションとQLA Clientアプリケーションに表示されています。

6 "Qla.properties"を保存して閉じます。

拡張設定ファイル

設定ファイルは、展開しやすくするため **base**と **ext**に分割されています。

ソフトウェアをアップグレードした後で、カスタムの拡張子を分離して保守するため、サーバーインストールのextフォルダの **.ext**ファイルに、ユーザー固有のカスタム**ビーン**、**プロセス**、**パブリッシングチャンネル**を定義する必要があります。

- ChannelConfig-ext.xml
- content-mimetype-mappings-ext.xml
- custom-xml-types-ext.xml
- IndexingChannels-ext.xml
- PluginsContext-ext.xml
- ProcessConfig-ext.xml
- PublishingConfig-ext.xml

Admin Web Clientのユーザーアクティビティペインに表示されるユーザー名の設定

Quark Publishing Platformでは、**ユーザーアクティビティ**ペインでのユーザー名の表示方法を、次の3とおりに設定できます。

- [ユーザー名]
- [ユーザー名]（[名] [姓]）
- [ユーザー名]（[姓], [名]）

この設定を変更するには、[QPP Server]/webapps/admin/WEB-INF/classesフォルダにある"WebAdminConfig.properties"設定ファイルを開きます。 userNameFormattingプロパティの値を、下記のいずれかに設定します。

- 0は、[ユーザー名]を表示します。
- 1は、[ユーザー名]（[名] [姓]）を表示します。
- 2は、[ユーザー名]（[姓], [名]）を表示します。

Quark XML Author - SharePoint Adapterによるパブリッシングで使用するようにPlatform Serverを設定する方法

Quark XML Author - SharePoint Adapterによるパブリッシングで使用するようにQuark Publishing Platform Serverを設定するには、下記の手順に従ってください。

1. [QPP Server]/publishingフォルダ内の"sharepoint.properties"ファイルを開きます。
2. sharepoint.usernameの値を、SharePointサイトにアクセスするユーザーのログイン名に設定します。
3. sharepoint.userpasswordの値を、上記で指定したユーザーのログインパスワードに設定します。
4. sharepoint.userdomainの値を、上記で指定したユーザーのドメインに設定します。

5 sharepoint.sitecollectionの値を、パブリッシング時に必要なドキュメントを含むSharePointサイトのコレクションのURLに設定します。

6 サーバーを再起動します。

Quark XML Author - FileNet AdapterによるパブリッシングでPlatform Serverを設定する方法

Quark XML Author - FileNet AdapterによるパブリッシングでQuark Publishing Platform Serverを設定するには、下記の手順に従ってください。

1 [QPP Server]/publishingフォルダ内の"contentengine.properties"ファイルを開きます。

2 filenet.stanzaの値を、FileNetのコンテンツエンジン接続のスタンザに設定します。 パラメータを編集する必要があるのは、FileNetサーバーがFileNet以外のスタンザを使用するように設定されている場合のみです。

3 filenet.usernameの値を、FileNetユーザー名に設定します。

4 filenet.userpasswordの値を、上記で指定したユーザーのFileNetパスワードに設定します。

5 filenet.connectionuriの値を、FileNetコンテンツエンジンWebServiceの接続URIに設定します。

6 サーバーを再起動します。

7

WindowsでのJVMメモリ割り当て

Windowsでは、Quark Publishing Platform Serverの起動方法に応じて、JVMのメモリ割り当てを異なる場所に指定できます。64ビットのオペレーティングシステムでは、Quark Publishing PlatformのJavaプロセス用にそれ以上のメモリを割り当てできます。いずれの場合も、使用可能なメモリの50パーセントを超えるメモリを割り当てるべきではありません。

Quark Publishing Platform Server ConsoleまたはQuark Publishing Platform Server Windowsサービスの使用

1 Quark Publishing Platform Serverを停止します。

2 Quark Publishing Platform Server ConsoleまたQuark Publishing Platform Server Windowsサービスを使用してQuark Publishing Platform Serverを起動する場合は、"wrapper.conf"ファイルを開きます。

3 wrapper.java.maxmemoryプロパティを探します。

4 値を調整します。64ビットのオペレーティングシステムでは、それ以上の値を指定できます。

5 変更を保存し、Quark Publishing Platform Serverを再起動します。

"Serverstartup.bat"の使用法

1. Quark Publishing Platform Serverを停止します。
2. Quark Publishing Platform Serverのインストールフォルダに"ServerStartup.bat"ファイルを置いてQuark Publishing Platform Serverを起動する場合は、"ServerStartup.bat"を開きます。
3. java -server -Xmx2048m -XX:MaxPermSize=256m -classpath.2048mを探します。これは、1536MBのRAMがQuark Publishing Platform Serverに割り当てられることを示します。
4. 64ビットのオペレーティングシステムでは、それ以上の値を指定できます。
5. 変更を保存し、Quark Publishing Platform Serverを再起動します。

Windowsn認証の設定

Windowsユーザーは、ダイアログボックスを表示せずにWindowsユーザー資格情報を使用して透過的にQuark Publishing Platformにログインできます。Quark Publishing Platformは、NTLM-v1/NTLM-v2、Negotiate/Kerberosなど、すべてのWindows 認証スキームに対応しています。混在モードにも対応し、Windows認証とPlatform認証は共存できます。

Windows認証は、脱着可能なHTTPサーブレットフィルタで設定します。各Webアプリケーションの"web.xml"ファイルにセキュリティフィルタを追加することで、配置時に認証を簡単に有効化および無効化できます。

```
<filter>
    <filter-name>SecurityFilter</filter-name>
    <filter-class>com.quark.web.security.servlet.ApplicationSecurityFilter</filter-class>
    <init-param>
        <param-name>provider</param-name>
        <param-value>com.quark.web.security.waffle.WaffleAuthenticationProvider</param-value>
    </init-param>
    <init-param>
        <param-name>provider/protocols</param-name>
        <param-value>Negotiate NTLM</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>SecurityFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

Quark Publishing PlatformのWebアプリケーションごとに、以下のWindows認証スキームを設定できます。

Webアプリケーション	NTLMのみ	Negotiate/Kerberosのみ	NTLMおよびKerberos
ワークスペース	NTLM	Negotiate	NTLM Negotiate
管理者	NTLM	Negotiate	NTLM Negotiate
Web サービス	NTLM	Kerberos	NTLM Negotiate
QXPSM	該当なし	該当なし	該当なし
メッセージング	該当なし	該当なし	該当なし
REST	該当なし	該当なし	該当なし

```
    <param-name>provider</param-name>
    <param-value>com.quark.web.security.waffle.WaffleAuthenticationPr
</init-param>
<init-param>
    <param-name>provider/protocols</param-name>
    <param-value>Negotiate NTLM</param-value>
</init-param>
filter>
lter-mapping>
 <filter-name>SecurityFilter</filter-name>
```

Platform ClientでSSO（シングルサインオン）を有効にする手順

Admin Webアプリの場合

1. C:¥Program Files¥Quark¥Quark Publishing Platform¥Server¥webapps¥admin¥WEB-INFに移動します。
2. Web.xmlを開き、以下のスニペットを探します。

    <filter> <filter-name>SecurityFilter</filter-name> <filter-class>com.quark.web.security.servlet.ApplicationSecurityFilter</filter-class> <init-param> <param-name>provider</param-name> <param-value>com.quark.web.security.waffle.WaffleAuthenticationProvider</param-value> </init-param> <init-param> <param-name>provider/protocols</param-name> <param-value>NTLM</param-value> </init-param> </filter> <filter-mapping> <filter-name>SecurityFilter</filter-name> <url-pattern>/*</url-pattern> </filter-mapping>

3. 上記のスニペットのコメント化を解除して、ファイルを保存します。

   ワークスペースやデスクトップクライアントの場合は、それぞれのWebアプリに移動して、web.xmlファイルを開き、同じスニペットのコメント化を解除します。

   - デスクトップクライアントの場合 C:¥Program Files¥Quark¥Quark Publishing Platform¥Server¥webapps¥webservices¥WEB-INF
   - ワークスペースの場合 C:¥Program Files¥Quark¥Quark Publishing Platform¥Server¥webapps¥workspace¥WEB-INF

➡ Platformサーバーを、ローカルシステムのアカウントによるサービスとして実行します。

混在モード認証用にWeb Clientを設定する手順

1. C:¥Program Files¥Quark¥Quark Publishing Platform¥Server¥webapps¥workspaceに移動します。PreLogin.jspのコピーを、local.jspという名前で作成します。
2. C:¥Program Files¥Quark¥Quark Publishing Platform¥Server¥webapps¥workspace¥WEB-INFに移動します。
3. Web.xmlを開き、以下のスニペットを探します。

   < filter > < filter-name > SecurityFilter </ filter-name > < filter-class > com.quark.web.security.servlet.ApplicationSecurityFilter </ filter-class > < init-param > < param-name > provider </ param-name > < param-value > com.quark.web.security.waffle.WaffleAuthenticationProvider </ param-value > </ init-param > < init-param > < param-name > provider/protocols </ param-name > < param-value > NTLM </ param-value > </ init-param > **<init-param> <param-name>exclude-url-patterns</param-name> <param-value>/local.jsp,/Login.jsp</param-value> </init-param>** </ filter > < filter-mapping > < filter-name > SecurityFilter </ filter-name > < url-pattern > /* </ url-pattern > </ filter-mapping >

4. 上記のスニペットのコメント化を解除して、ファイルを保存します。上記のスニペットで、太字の行は追加する必要があります。

5 サーバーを再起動し、次のURLを使用して、対応するWebアプリケーションにアクセスします。

- ネイティブのPlatformユーザー認証でアクセスする場合、下記のURLを使用します。 http://localhost:61400/workspace/local.jsp
- Windows認証でアクセスする場合、下記のURLを使用します。 http://localhost:61400/workspace/

➡ 上記の手順は、Web Adminや他のWebアプリについても繰り返すことができます。

"log4j.xml"の編集

ログ出力されたイベントにユーザー情報を追加する

ログ出力される各イベントにユーザーIDとユーザー名を記録するように、log4j.xml fileを設定できます。

1 {QPP Server}/confフォルダにある"log4j.xml"ファイルを開きます。

2 conversionPatternパラメータを設定します。

```
 <appender name="QpsServerAllFileAppender"
class="org.apache.log4j.RollingFileAppender"> <param name="file"
value="log/QppServer.log"/> <param name="maxFileSize" value="10MB"/> <param
name="maxBackupIndex" value="10"/> <layout
class="org.apache.log4j.EnhancedPatternLayout"> <param name="conversionPattern"
value="%d %p [%c][%t]
[User(Id:%properties{qpp.user.id},Name:%properties{qpp.user.name})] - % m%n"/>
</layout> </appender>
```

ログレベルの設定

"log4j.xml"ファイルを編集してログレベルを調整できます。また、Quark Publishing Platform Serverの起動後にJConsoleを使用してログレベルを変更できます。例外に対して異なるログレベルを設定することもできます。

"log4j.xml"でのログレベルの変更

Quark Publishing Platform Web ClientおよびQuark Publishing Platform Serverのログレベルを変更できます。オプションにはERROR、INFO、WARN、DEBUG、SQLTRACE、およびTRACEがあります。

- ERRORは、中断された要求または失敗した要求を示すメッセージが記録されます。
- INFOは、サービスの状態を示すメッセージが記録されます。
- WARNは、致命的でないサービスエラーメッセージが記録されます。
- DEBUGは、サーバーリソースの使用を示すメッセージが記録されます。
- SQL_TRACEは、SQL要求に関連したアクティビティによるメッセージが記録されます。
- TRACEは、要求に関連したアクティビティによるメッセージが記録されます。

ログレベルについての詳細は、Javaの資料を参照してください。

ログレベルを変更するには以下の手順に従ってください。

1 {QPP Server}/confフォルダにある"log4j.xml"ファイルを開きます。

2 Quark Publishing Platform Web Clientアクティビティのログレベルを定義するには、<logger name=com.quark.qpp.web.webeditorまでスクロールします。構造は下記の通りです。

```
<logger name="com.quark.qpp.web.webeditor" additivity="false"> <level value="INFO" />
 <appender-ref ref="WebHubAsyncAppender" /> </logger>
```

3 Quark Publishing Platform Serverアクティビティのログレベルを定義するには、<logger name=com.quark.qppまでスクロールします。構造は下記の通りです。

```
<logger name="com.quark.qpp"> <level value="INFO" /> </logger>
```

4 その他のアクティビティのログレベルを定義するには、「<root>」までスクロールします。構造は下記の通りです。

```
<root> <priority value="ERROR" /> <appender-ref ref="QppServerAsyncAppender" />
</root>
```

5 "log4j.xml"を保存して閉じます。

Quark Publishing Platform Server起動後のログレベルの変更

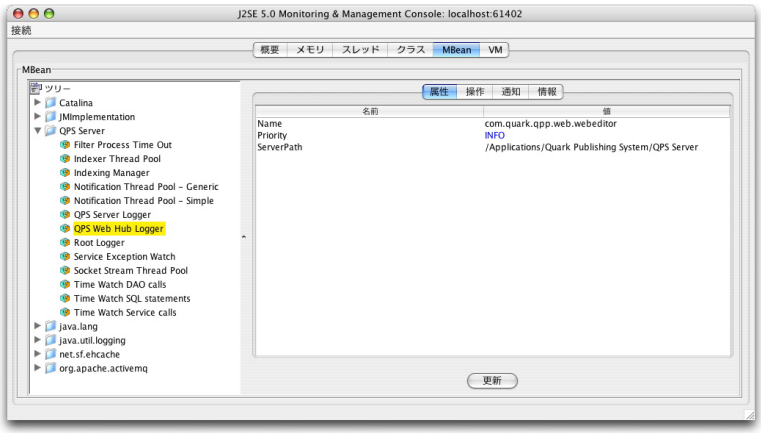
1 Quark Publishing Platform Serverの実行中に**Platform Server Console**を表示します。

2 **JConsole**をクリックしてQuark Publishing Platform Serverパフォーマンスのさまざまな側面を監視するウィンドウを表示します。

3 **MBeans**タブをクリックします。**MBeans**タブの左側にさまざまなQuark Publishing Platform Server機能がツリー形式で表示されます。

4 ツリーから**Platformサーバー > ログ**を開きます。

5 **Operations**タブをクリックします。



JConsoleを使用してログの優先レベルを調整します

6 Quark Publishing Platform ServerまたはQuark Publishing Platform Web Clientのログレベルを編集するには、テキストボックスにログレベルを設定し、対応するボタンをクリックします。

➡ JConsoleで行った変更は、すぐに有効になりますが、Quark Publishing Platform Serverを再起動すると"log4j.xml"の設定が適用されます。

例外に対するログ記録の変更

"ServerApp.properties"ファイル内の2つの値を編集することによって、未知および既知の例外に対するログ記録を設定できます。

1. Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。
2. "ServerApp.properties"をテキスト編集アプリケーションで開きます。
3. Quark Publishing Platformの例外をQuark Publishing Platform Serverログに記録しない場合は、server.logqppserviceexceptionをfalseに設定します。
4. 未知の例外をログに記録しない場合は、server.logthrowableをfalseに設定します。
5. "ServerApp.properties"を保存して閉じます。

検索通知評価設定の変更

すべての開いている**検索結果**パレットは、パレットに表示されているアセットが変更された場合にQuark Publishing Platform Serverから通知を受けます。いくつかのパラメータを編集して、これらのクエリー通知の条件を評価し配信するために使用される計画とリソースに影響を与えることができます。多くのパラメータの最適値は、Quark Publishing Platform Serverに使用するデータベースとハードウェアによって異なります。

➡ 経験のある管理者のみが下記で説明する設定を変更するようにしてください。サポートが必要な場合には、Quark Enterprise Supportにお問い合わせください。

1. Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。
2. "Query.properties"をテキスト編集アプリケーションで開きます。
3. HSQLデータベースを使用している場合、Quark Publishing Platform Serverがクエリー通知を配信する前にアセットのメタデータを一時テーブルに保存するように設定することで、パフォーマンスを最適化できます。HSQLデータベース使用時の検索パフォーマンスを最適化するには、query.notification.useTempTable=パラメータまでスクロールし、値をtrueに設定します。

➡ HSQLデータベースを使用しない場合には、値をfalseに設定します。デフォルトでは、ユーザーがインストール時にQuark Publishing Platform Databaseオプションを選択すると、Quark Publishing Platform Server Installerによってこのパラメータが正しく設定されます。

4. QPQuark Publishing Platform Serverでは、クエリー通知の評価に2つのスレッドプールが使用されます。「一般通知評価スレッドプール設定」と「単純通知評価スレッドプール設定」です。「一般」スレッドプールではデータベースを使用してクエリー通知が評価されます。「単純」スレッドプールでは、データベースを使用せずに、より単純な方法でクエリー通知が評価されます。「Generic Notification Evaluator Thread Pool Configuration」領域で、下記のプロパティを調整します。

- データベースアクセスを必要とする通知を評価するためにバックグラウンドで同時に実行されるスレッド数の最大値を指定するには、query.notification.generic.pool.maxActive値を調整します。複数のプロセッサと拡張システムメモリがあるハードウェアを使用している場合には、この値を大きくしてパフォーマンスを向上させます。
- プール内のアイドルスレッド数の最大値を指定するには、query.notification.generic.pool.maxIdleプロパティを調整します。
- プール内のアイドルスレッド数の最小値を指定するには、query.notification.generic.pool.minIdleプロパティを調整します。
- アイドル状態になったスレッドがプールから排除されるまでの最短時間を指定するには、query.notification.generic.pool.minEvictableIdleTimeMillisプロパティに時間をミリ秒単位で設定します。
- バックグラウンド排除スレッドが実行されてアイドルスレッドが排除されるまでのミリ秒数を指定するには、query.notification.generic.pool.timeBetweenEvictionRunsMillisプロパティを調整します。

5 特定のアセット変化のクエリー通知は、データベースアクセスを必要とせずに、より単純な方法を使用して効果的に評価できます。「Simple Notification Evaluator Thread Pool Configuration」領域で、下記のプロパティを調整します。

- バックグラウンドで同時に実行されるスレッド数の最大値を指定するには、query.notification.simple.pool.maxActive値を調整します。複数のプロセッサと拡張システムメモリがあるハードウェアを使用している場合には、この値を大きくしてパフォーマンスを向上させます。
- プール内のアイドルスレッド数の最大値を指定するには、query.notification.simple.pool.maxIdleプロパティを調整します。
- プール内のアイドルスレッド数の最小値を指定するには、query.notification.simple.pool.minIdleプロパティを調整します。
- アイドル状態になったスレッドがプールから排除されるまでの最短時間を指定するには、query.notification.simple.pool.minEvictableIdleTimeMillisプロパティに時間をミリ秒単位で設定します。
- バックグラウンド排除スレッドが実行されてアイドルスレッドが排除されるまでのミリ秒数を指定するには、query.notification.simple.pool.timeBetweenEvictionRunsMillisプロパティを調整します。

6 クエリー通知にJMSトポロジーを設定するには、query.notification.topicPerSessionプロパティを調整します。バージョン8.1以前のQuark Publishing Platformと同じ動作をさせるには、この値にfalseを設定します。

7 "Query.properties"設定ファイルのその他のプロパティは編集しないでください。

8 "Query.properties"を保存して閉じます。

9 Quark Publishing Platform Serverを再起動してこれらの設定を有効にします。

データベースのプロパティ

データベース接続URL、データベースユーザー名、データベースユーザーパスワード、およびデータベース接続プールサイズを手動で指定できます。

データベースのプロパティを変更するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダにある"Database.properties"ファイルを開きます。
2. Database related configurationセクションまでスクロールします。
3. ドライバクラス名を指定するには、qpp.jdbc.driverClassNameの値を置き換えます。たとえば、HSQLデータベースの場合は、qpp.jdbc.driverClassName=org.hsqldb.jdbcDriverを使用します。
4. データベース接続URLを指定するには、qpp.jdbc.urlの後のパスを置き換えます。
5. データベースユーザー名を指定するには、qpp.jdbc.userNameの値を置き換えます。
6. データベースユーザーパスワードを指定するには、qpp.jdbc.passwordの値を置き換えます。
7. データベース接続プールサイズを指定するには、qpp.jdbc.maxActiveの値を変更します。
8. アイドル接続の最大数と最小数を設定するには、qpp.jdbc.maxIdleおよびqpp.jdbc.minIdleの値を変更します。
9. 接続がプールから排除されるまでの最短時間（ミリ秒）を指定するには、qpp.jdbc.minEvictableIdleTimeMillisの値を変更します。
10. アイドル状態の接続を排除する排除スレッドが実行されるまでの時間（ミリ秒）を指定するには、qpp.jdbc.timeBetweenEvictionRunsMillisの値を変更します。
11. プリペアドステートメントのプーリングを調整するには、qpp.jdbc.poolPreparedStatementsの値を変更します。
12. 自動コミットではないプールでの接続数の最大値を指定するには、qpp.jdbc.maxActive_noAutoCommitを指定します。
13. "Database.properties"を保存して閉じます。

➡ Platform ServerとDatabase Serverとの間にファイアウォールが設定されており、長時間のアイドル後にTCPセッションを終了できる場合は、"Database.properties"ファイルに次の設定変更を加える必要があります。

- qpp.jdbc.minIdleの値を **0**に設定します。
- qpp.jdbc.minEvictableIdleTimeMillisの値を **60000**に設定します。
- qpp.jdbc.timeBetweenEvictionRunsMillisの値を **60000**に設定します。

これらの設定を有効にするため、Platform Serverを再起動する必要があります。

変換プロパティ

変換プロパティを使用してさまざまな定義済み変換出力を変更できます。"Transformation.properties"ファイルを変更することで、アセットの変換に呼び出す外部コマンドが変更されます。このファイルのコマンドには、<temp>、<source>、<extension>、<output>などの定義済みマクロが含まれます。

変換プロパティを変更するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダにある"Transformation.properties"ファイルを開きます。
2. 他のプロパティの値または値の一部とならないパターンを指定するには、transformer.spaceEncodingPatternの値を指定します。このパターンは、コマンドや他のプロパティ値におけるスペースのエスケープに使用されます。
3. ImageMagickが処理できるファイルタイプを指定するには、imTransformer.extensionsにカンマ区切りで、大文字小文字を区別してリストを入力します。
4. ImageMagickTransformerで使用される一時ファイルの場所を指定するには、imTransformer.tempDirのパスを指定します。
5. ImageMagickTransformerで出力を生成するためのImageMagickコマンドを指定するには、imTransformer.jpg.imCommandに値を入力します。
6. JawsTransformerが処理できるファイルタイプを指定するには、jawsTransformer.extensionsにカンマ区切りで、大文字小文字を区別してリストを入力します。
7. JawsTransformerで使用される一時ファイルの場所を指定するには、jawsTransformer.tempDirのパスを指定します。
8. jawsTransformer.tempDirで出力を生成するためのJawsコマンドを指定するには、jawsTransformer.jpg.commandに値を入力します。
9. "Transformation.properties"を保存して閉じます。

セッションタイムアウト

個々のセッションがアイドルになってからタイムアウトになるまでの時間とアイドルセッションを確認する頻度を指定できます。

タイムアウトパラメータを指定するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダにある"ServerApp.properties"ファイルを開きます。
2. session.maxIdle=エントリまでスクロールします。
3. セッションがアイドルになってからタイムアウトになるまでの秒数を入力します。
4. Quark Publishing Platform Serverがバックグラウンドでアイドルセッションの確認を実行する頻度を指定するには、session.eviction.thread.delayプロパティに秒数を入力します。
5. "ServerApp.properties"を保存して閉じます。

➡ Web Clientのタイムアウト設定は、"Workspace-Config.xml"ファイルのajaxTimeoutエントリを調整して変更できます。このエントリの値はミリ秒単位です。このファイルは{install_folder}/webapps/workspace/WEB-INF/classesにあります。

リポジトリ状況アップデータ

Quark Publishing Platform File Serverの状況を確認するために頻繁に実行されるバックグラウンドスレッドを指定できます。

リポジトリ状況を更新する間隔を指定するには、下記の手順に従ってください。

1. Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。
2. "ServerApp.properties"をテキスト編集アプリケーションで開きます。
3. repository.status.updator.sleepInterval=エントリーまでスクロールします。
4. リポジトリ状況アップデータスレッドが実行される間隔を秒数で指定する値を入力します。
5. "ServerApp.properties"を保存して閉じます。

Quark Publishing Platform Rendererの移行

Quark Publishing Platform Rendererを別のコンピュータに移動する必要がある場合、Quark Publishing Platform Serverを再インストールする必要はありません。代わりに"ManagerConfig.xml"ファイルを下記の手順に従って編集できます。

1. [QPS Server]/confフォルダ内の"ManagerConfig.xml"ファイルを開きます。
2. "ManagerConfig.xml"の最下部にある<connectioninfo>セクションまでスクロールします。
3. nameエントリーの値を新しいQuark Publishing Platform RendererのIPアドレスまたはホスト名に変更します。
4. portエントリーの値を新しいQuark Publishing Platform Renderer用に指定したポート番号に変更します。
5. "ManagerConfig.xml"を保存してQuark Publishing Platform Serverを起動します。
6. 変更を確認するには、"QppServer.log"ファイルでSuccessfully registered with QXPSという行を探します。

➡ またはURLhttp://[サーバー]:[ポート]/qxpsmadminでQuark Publishing Platform Renderer Managerクライアントを開き、そこでこれらの変更を行うこともできます。

➡ QuarkXPress**のサーバー設定**ダイアログボックスの**HTTP**タブで**管理要求に対してレルムを確認する**にチェックを付けてユーザー名とパスワードを入力した場合以外は、<user>エントリーと<password>エントリーは空白のままにしておくことができます。これは、"ServerApp.properties"ファイルのユーザー名フィールドqxps.usernameとパスワードフィールドqxps.passwordにもあてはまります。

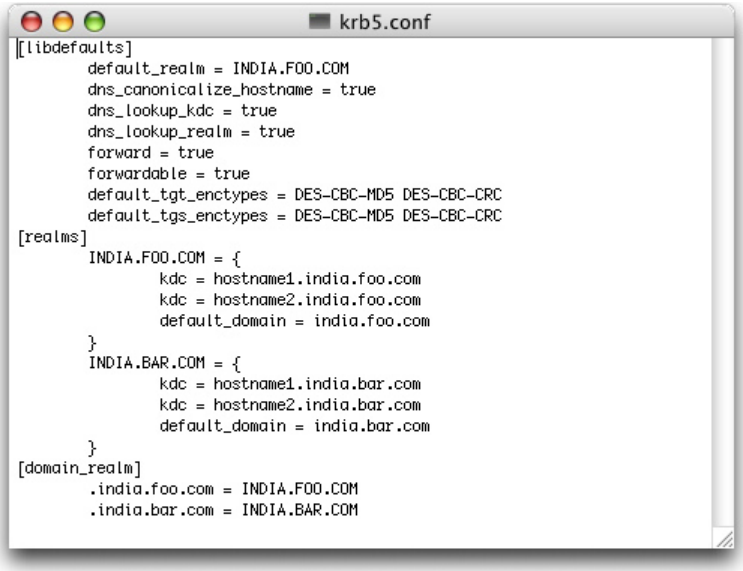
Quark Publishing PlatformとLDAPの統合

Quark Publishing PlatformとLDAPを統合する方法の詳細は、『**A Guide to Quark Publishing Platform**』の「LDAPによるユーザーリストの管理」を参照してください。

Kerberos認証の使用

Kerberos認証を使用するには、下記の手順に従ってください。

1. "krb5.conf"をテキスト編集アプリケーションで開きます。"krb5.conf"にはKerberos設定情報が含まれています。これにはKey Distribution Center（KDC）やKerberosレルムの管理サーバーの場所が含まれます。Kerberosは、コンピュータネットワーク上のサービスへの要求を認証するためのセキュリティ保護された方法です。



"krb5.conf"ファイルでレルム設定を指定します

2. "krb5.conf"を編集してすべてのレルム設定を指定します。これにはレルム名、デフォルトレルム、レルムドメイン間マッピング、およびレルムのKDCの設定が含まれます。
3. "krb5.conf"を保存して閉じます。

➡ 慣習として、レルム名は大文字で入力し、ドメイン名は小文字で入力します。

シンプル認証の使用

WindowsでActive Directory以外のディレクトリサーバーを使用している場合、ほとんどの場合はシンプル認証スキームを使用します。 シンプル認証を使用するには、下記の手順に従ってください。

1. LDAP属性のマッピングで、Authentication Name= :uid: User Name=uidを設定します。
2. シンプル認証の場合、Realm Nameは省略できます。

LDAPユーザーパスワードのQuark Publishing Platform Serverへの接続

下記の手順に従ってQuark Publishing Platformを設定すると、Quark Publishing Platform Serverとディレクトリサーバーの間の接続が切断された場合でも、ユーザーはディレクトリサーバーの資格情報を使用してQuark Publishing Platform Serverにログオンできます。

ユーザーがQuark Publishing Platform Serverへのログオンに初めて成功すると、暗号化されたユーザーの資格情報がQuark Publishing Platformに保存されます。Quark Publishing Platform Serverとディレクトリサーバー間に接続がない場合、これらの資格情報が以降のログオン操作に使用できます。

LDAPユーザーパスワードの取得、暗号化、Quark Publishing Platform Serverデータベースへの保存ができるように設定するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダにある"ServerApp.properties"ファイルを開きます。
2. Quark Publishing Platform Serverとディレクトリサーバーの間の接続が切断された場合にもユーザーがQuark Publishing Platform Serverにログオンできるようにするには、authentication.external.cacheTicket=の値をtrueに変更します。authentication.external.cacheTicket=のデフォルトの値はfalse（無効）です。
3. コンテキスト内のすべてのコレクションについて、実行時に"CollectionInfo#isAccessibleChildrenAvailable()"フラグを評価する必要があることを示すため、collections.evaluateAccessibleChildrenFlag=の値はtrue（デフォルト）にする必要があります。
4. LDAPサーバーからフェッチされた要素は、Platform Serverへキャッシュされ、指定された秒数の間だけキャッシュから提供され、その後で再度フェッチされます。LDAPキャッシュ要素の持続時間を指定するには、ldap.cache.timetolive=の値を秒単位で設定します。
5. LDAP同期遅延時間を秒単位で指定するには、ldap.synchronization.thread.delay=の値をtrueに設定します。LDAP同期が必要ない場合、値を-1に設定します。
6. "ServerApp.properties"を保存して閉じます。

MSSQL Server Windows認証用にPlatformを設定する

Quark Publishing Platform ServerをSQL Serverデータベースとともに、Windows認証が有効な状態で実行するには、下記の変更を加える必要があります。

1. [QPP Server]/confフォルダ内の"database.properties"ファイルを開きます。
2. 次のプロパティをコメント解除し、該当の情報を入力します。

   qpp.jdbc.url=jdbc:sqlserver://<your-host-name>¥¥<instanceName>;databaseName=qppdb;integratedSecurity=true;

3. Javaのclasspath変数に、Microsoft JDBCドライバに付属するsqljdbc_auth.dll.sqljdbc_auth.dllへのポインタを追加します。

   これは、*msdn.microsoft.com/en-us/library/mt683464(v=sql.110).aspx*でダウンロードできます。

4. サーバーを起動します。

ワークスペースブラウザパレットの制限

ユーザーが同時に開くことのできる**ワークスペースブラウザ**ウィンドウの数を制限するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダ内の"Query.properties"ファイルを開きます。
2. query.maxWatchedQueryCountPerSessionパラメータの値を指定します。
3. "Query.properties"を保存して閉じます。

クエリーの結果設定の変更

- **割り当て**ベースの検索で返される結果の数を制限するには、query.maxAssignmentFetchCountパラメータに0でない正の値を指定します。
- **ユーザー**ベースの検索で返される結果の数を制限するには、query.maxRowFetchCountパラメータに0でない正の値を指定します。

カスタムコンテンツタイプ検出の設定

デフォルトでは、Quark Publishing PlatformはQuark形式、共通画像形式、共通コンテンツ形式（Microsoft Office、プレーンテキスト、RTF、PDF、HTML）XML、XHTML、DITAドキュメント、BusDocドキュメントを認識するように設定されています。 ただし、追加のカスタムコンテンツタイプを自動的に使用するようQuark Pulishingを設定できます。 これを行うには、下記の手順に従ってください。

1. ファイルのMIMEタイプの検証により特定のコンテンツタイプが認識されるようにするには、"tika-mimetypes.xml" fileのリストにMIMEタイプを追加します。 MIMEタイプをこのファイルに含めない場合、"custom-mimetypes"ファイルに追加できます。 ファイルの種類は、ファイル拡張子、マジックバイト、またはルート要素（XMLファイルの場合）によって識別できます。 例：

   ```
   <mime-type type="application/dita+xml; format=busdoc"> <sub-class-of type="application/dita+xml;format=topic"/> <_comment>Business Documents</_comment> </mime-type>
   ```

   構造的に類似していても意味的に異なるXMLファイルには異なるMIMEタイプを割り当てることができます。 当該XML内に適用される処理命令にもとづいて、XMLファイルの種類を識別できます。 このためには、Platformサーバーの"ext"フォルダに存在する"custom-xml-types-ext.xml"ファイルにMIMEタイプのエントリを追加します。 例：

   ```
   <mime-type type="application/dita+xml; format=busdoc" sub-class-of="application/dita+xml; format=topic"> <pi name="Xpress">productLine="busdoc"</pi> </mime-type>
   ```

   詳細は、*http://tika.apache.org*を参照してください。

2. 識別したMIMIタイプに基づいてQuark Publishing Platformコンテンツタイプを設定します。 このためには、Platformサーバーの"ext"フォルダに存在する"content-mimetype-mappings-ext.xml"ファイルにマッピングのエントリを追加します。 指定したカスタムMIMEタイプのアセットは、指定したカスタムコンテンツタイプでチェックインされます。 例：

   ```
   <content-type name="Business Document"> <mime-type>application/dita+xml; format=busdoc</mime-type> </content-type>
   ```

3 Platformサーバーの"ext"フォルダに存在する"Indexing Channels-ext.xml"ファイルに、対応するマッピングのエントリを追加します。 例：

```
<mapping contenttype="Ratings Document" channel="ratingsPreviewChannel"> <parameter
 name="channelParam1">param value</parameter> <parameter
name="channelParam2">param value</parameter> </mapping>
```

デフォルトPDF出力スタイルの指定

Quark Publishing Platform Web Clientユーザーは、QuarkXPressプロジェクトやQuarkCopyDeskアーティクルのコピーをPDFとして取得できます。 これらのPDFの設定は、QuarkXPress Serverのデフォルト PDF出力スタイルのみで定義されます。 デフォルトのPDF出力スタイルを指定するには、『QuarkXPress Serverガイド』の「ジョブジャケットダイアログボックス」の説明に従って、デフォルトのジョブジャケットファイルを更新します。 デフォルトのPDFスタイルは、PDFを取得するため使用されるパブリッシングチャンネルでも指定できます。 QuarkXPressプロジェクトから PDFを取得するデフォルトのチャンネルは"qxpPdf"です。

Web Clientでの、配信チャネルの表示設定の制御

デフォルトでは、配信チャネルは表示されません。 配信チャネルを利用可能にするには、QPP_HOME¥webapps¥workspace¥WEB-INF¥classes¥Workspace-Config.xmlファイルでdisplaydeliverychannelsプロパティをtrueに設定します。

```
<property name="enableDeliveryChannels" value="true" />
```

Quark Publishing Platformパスワードの大文字と小文字の区別の指定

Quark Publishing Platformユーザーパスワードの大文字と小文字の区別を指定するには、下記の手順に従ってください。

1 [QPP Server]/confフォルダにある"ServerApp.properties"ファイルを開きます。

2 ユーザーパスワードの大文字と小文字を区別するように指定するには、server.password.case.sensitiveをtrueに設定します。区別しない場合にはfalseと入力します。

3 "ServerApp.properties"を保存して閉じます。

➡ server.password.case.sensitiveのデフォルト設定はtrueです。Quark Publishing Platform管理者は、パスワードで大文字と小文字の区別が必要な場合にはユーザーに通知する必要があります。

フィルタと索引サービス設定の管理

Quark Publishing Platform Serverでは、プレビューを生成するために4つのフィルタが使用されます。各フィルタの".properties"ファイルで値を変更できます。これには下記のファイルが含まれます。

- "AsposeFilterServiceConfig"ファイルは、Word、RTF、Excel、PowerPoint、およびプレーンテキストファイルのプレビューとサムネールを提供します。
- "QxpsFilterServiceConfig.properties"ファイルは、QuarkXPressプロジェクトおよびQuarkCopyDeskアーティクルのプレビューとサムネールを提供します。
- "IMFilterServiceConfig.properties"ファイルは、ほとんどの画像ファイルのプレビューとサムネールを提供します。
- "JawsFilterServiceConfig.properties"ファイルは、特定の画像ファイル、すなわちPDFファイル、EPSファイル、およびAdobe® Illustrator®ファイルのプレビューとサムネールを提供します。
- "XADocFilterServiceConfig.properties"ファイルは、XMLファイルのプレビューとサムネールを提供します。

下記のコマンドは、4つのフィルタすべてに共通です。

- <フィルタ名>filter.generateAttributes：フィルタ固有の属性を生成するには、これをtrueに設定します。
- <フィルタ名>filter.generateAttributes：プレビューを生成するには、これをtrueに設定します。
- <フィルタ名>filter.generateAttributes：サムネールを生成するには、これをtrueに設定します。
- <フィルタ名>filter.generateAttributes：テキストを取り出してフルテキスト検索を有効にするには、これをtrueに設定します。
- <フィルタ名>filter.generateAttributes：プレビューの幅をピクセル単位で指定するにはこれを設定します。
- <フィルタ名>filter.previewHeight：プレビューの高さをピクセル単位で指定するにはこれを設定します。
- <フィルタ名>filter.thumbnailHeight：サムネールの高さをピクセル単位で指定するにはこれを設定します。
- <フィルタ名>filter.thumbnailWidth：サムネールの幅をピクセル単位で指定するにはこれを設定します。
- <フィルタ名>filter.previewPages：生成するプレビューページ数を指定するにはこれを設定します。
- <フィルタ名>filter.processTimeOut：不完全な索引プレビュープロセスが終了するまでのミリ秒数を指定するにはこれを設定します。

索引サービス設定

アセット索引作成のためのパラメータを指定できます。索引作成はQuark Publishing Platform Server内部でバックグラウンドプロセスとして実行され、多くのシステムリソースを消費する場合があります。下記の手順に従って「索引プロパティ」ファイルを編集することにより、「索引作成スレッドプール」と呼ばれる索引プロセスを、同時に実行されるスレッドプール数とスレッドプール間の間隔を調整することによって管理できます。

アセット索引作成の間隔とスレッドを管理するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダにある"Indexing.properties"ファイルを開きます。
2. バックグラウンドで同時に実行される索引作成スレッド数の最大値を指定するには、indexingThread.pool.maxActive=値を調整します。
3. プールから使用可能なスレッドを待つ時間をミリ秒単位で指定するには、indexingThread.pool.maxWait=値を調整します。ここで、-1は、無制限に待つことを示します。
4. プール内のアイドルスレッド数の最大値を指定するには、indexingThread.pool.maxIdle=値を調整します。
5. プール内のアイドルスレッド数の最小値を指定するには、indexingThread.pool.minIdle=値を調整します。
6. スレッドがプールから排除されるまでのミリ秒数を指定するには、indexingThread.pool.minEvictableIdleTimeMillis=を調整します。
7. アイドルスレッドを排除する排除スレッドが実行されるまでのミリ秒数を指定するには、indexingThread.pool.timeBetweenEvictionRunsMillis=値を調整します。排除スレッドはバックグラウンドで実行されます。

➡ 「排除」スレッドが実行されると、indexingThread.pool.maxIdle値を超えたすべてのアイドルスレッドがプールから解放されます。しかし、indexingThread.pool.minIdle値で指定された数のアイドルスレッドはプール内に保持されます。

8. Quark Publishing Platform Server起動前に索引付けされていなかったアセットの索引を作成するには、indexing.indexPendingAssetsOnStartup=値にtrueを入力します。
9. "Indexing.properties"を保存して閉じます。

ASPOSEフィルタ

ASPOSEフィルタは、テキストファイル、Microsoft Word、Excel、およびPowerPointのドキュメント、RTFファイル、プレーンテキストファイルを処理します。APSOSEフィルタの設定を調整するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダ内の"AsposeFilterServiceConfig.properties"ファイルを開きます。
2. asposefilter.previewType=png。サポートされているプレビュータイプは**png**と**jpg**です。
3. asposefilter.generatePreview=true
4. asposefilter.generateThumbnail=true
5. asposefilter.generateTableImages=false
6. asposefilter.generateChartImages=false
7. asposefilter.thumbnailHeight=エントリ。
8. asposefilter.thumbnailWidth=エントリ。
9. asposefilter.previewDpi=エントリ。
10. asposefilter.reditionPreviewDpi=エントリ。

11 一定時間以内に索引付けが完了しない場合にプロセスを終了するには、asposefilter.processTimeOut=エントリにミリ秒単位で値を入力します。

12 プレビューを生成するページの最大数を指定するには、asposefilter.previewPages=エントリに値を入力します。

13 RTF、XML、HTML、HTM、TXTタイプのファイルについてプレビューページレイアウトのマージンを設定するには、次のエントリに値を入力します。

- asposefilter.text.topMargin=エントリ。
- asposefilter.text.bottomMargin=エントリ。
- asposefilter.text.rightMargin=エントリ。
- asposefilter.text.leftMargin=エントリ。

14 "AsposeFilterServiceConfig.properties"を保存して閉じます。

PDFプレビュー用のASPOSEフィルタの設定

Platform Serverでは、PDFドキュメントの索引を作成するため、2つのフィルタ実装が用意されています。

- Jawsフィルタ
- ASPOSEフィルタ

デフォルトでは、PDFドキュメントの索引用にJAWSフィルタが設定されています。PDFドキュメントの索引用にASPOSEフィルタを設定するには、下記の手順に従ってください。

1 [QPP Server]/confフォルダ内の"AsposeFilterServiceConfig.properties"ファイルを開きます。

2 次の行を探して、コメント化を解除します。

```
<!-- <supported-file-type> <mac-os-type/> <file-extension>pdf</file-extension> <mime-type/>
 <creator-code/> </supported-file-type> <supported-file-type> <mac-os-type/>
<file-extension/> <mime-type>application/pdf</mim-type> <creator-code/>
</supported-file-type> <supported-file-type> <mac-os-type/> <file-extension/>
<mime-type>application/x-pdf</mime-type> <creator-code/> </supported-file-type> -->
```

3 "JawsFilterServiceConfig.properties"をテキスト編集アプリケーションで開きます。

4 次のような、サポートされているファイルタイプを探し、コメント化します。

```
<supported-file-type> <mac-os-type/> <file-extension>pdf</file-extension> <mime-type/>
<creator-code/> </supported-file-type>
```

5 "AsposeFilterServiceConfig.properties"を保存して閉じます。

6 "JawsFilterServiceConfig.properties"を保存して閉じます。

APSフィルタ

APSフィルタの設定を調整するには、下記の手順に従ってください。

1 [QPP Server]/confフォルダにある"ApsFilterServiceConfig.properties"ファイルを開きます。

2. apsfilter.generatePreview=true
3. apsfilter.generateThumbnail=true
4. apsfilter.generateAttributes=true
5. apsfilter.generateText=true
6. 索引の作成が終了しない場合にプロセスが終了するまでの待ち時間を指定するには、apsfilter.processTimeOut=エントリに値をミリ秒単位で入力します。
7. プレビューを生成するページの最大数を指定するには、apsfilter.previewPages=エントリに値を入力します。
8. "ApsFilterServiceConfig.properties"を保存して閉じます。

POIフィルタ

POIフィルタの設定を調整するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダにある"POIFilterServiceConfig.properties"ファイルを開きます。
2. poifilter.approximateCharactersPerPage。テキストのページを構成するだいたいの文字数を設定します。この値により、テキストファイルの改ページが決定されます。
3. poifilter.previewMaxLineLength。テキストファイルからプレビュー画像を生成するとき、1行に配置する文字の最大数を設定します。
4. poifilter.fontName。テキストから画像を生成するときに使用するフォントの名前。
5. poifilter.previewFontSize。画像内のテキストに必要なフォントのサイズ。
6. poifilter.lineSpacing。行の間隔を、poifilter.previewFontSizeプロパティに定義されているフォントサイズの倍数で指定します。
7. "POIFilterServiceConfig.properties"を保存して閉じます。

MS Officeドキュメントの処理の設定

MS Officeフィルタの設定を調整するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダ内の"OfficeServiceConfig.properties"ファイルを開きます。
2. office.defaultImageFormatプロパティを設定し、デフォルトの出力形式タイプを指定します。デフォルトの形式はPNGです。
3. office.defaultHorizontalResolutionプロパティを設定し、プレビューのデフォルト水平解像度を指定します。
4. office.defaultVerticalResolutionプロパティを設定し、プレビューのデフォルト垂直解像度を指定します。
5. office.onePagePerSheetプロパティを**true**に設定し、シートのすべてのコンテンツが1つのページにのみ出力されるようにします。
6. office.showHiddenWorksheetsプロパティを設定し、非表示ワークシートの可視性をコントロールします。

7 出力形式HTML、XHTML、SMARTTABLE、CALSで、隠しまたは非表示の列／行を取り除くオプションを設定するには、次のプロパティを設定します。

- office.defaultHiddenColumnDisplayType=エントリ。このオプションに使用できる値は、hideまたはremoveです。
- office.defaultHiddenRowDisplayType=エントリ。このオプションに使用できる値は、hideまたはremoveです。

8 office.defaultHorizontalResolution office.ppt.document.defaultImageFormatプロパティを設定し、PowerPointドキュメント出力のデフォルト画像形式を指定します。

9 office.defaultHorizontalResolution office.ppt.document.image.defaultHorizontalScalingFactorプロパティを設定し、PowerPointドキュメント出力のデフォルトの水平倍率を指定します。このプロパティは、JPEG、GIF、PNG、BMPの画像フォーマットに適用されます。

10 office.defaultHorizontalResolution office.ppt.document.image.defaultVerticalScalingFactorプロパティを設定し、PowerPointドキュメント出力のデフォルトの垂直倍率を指定します。このプロパティは、JPEG、GIF、PNG、BMPの画像フォーマットに適用されます。

11 office.defaultHorizontalResolution office.ppt.component.defaultImageFormatプロパティを設定し、PowerPointドキュメントのスライド出力のデフォルト画像形式を指定します。

12 office.defaultHorizontalResolution office.ppt.component.image.defaultHorizontalScalingFactorプロパティを設定し、PowerPointドキュメントのスライドのデフォルト水平倍率を指定します。このプロパティは、JPEG、GIF、PNG、BMPの画像フォーマットに適用されます。

13 office.defaultHorizontalResolution office.ppt.component.image.defaultVerticalScalingFactorプロパティを設定し、PowerPointドキュメントのスライドのデフォルト垂直倍率を指定します。このプロパティは、JPEG、GIF、PNG、BMPの画像フォーマットに適用されます。

14 office.defaultHorizontalResolution office.visio.document.defaultImageFormatプロパティを設定し、Visioドキュメント出力のデフォルト画像形式を指定します。

15 office.defaultHorizontalResolution office.visio.document.image.defaultResolutionプロパティを設定し、Visioドキュメント出力のデフォルト解像度を指定します。このプロパティは、JPEG、GIF、PNG、BMPの画像フォーマットに適用されます。

16 office.defaultHorizontalResolution office.visio.component.defaultImageFormatプロパティを設定し、Visioドキュメントのページ出力のデフォルト画像形式を指定します。

17 office.defaultHorizontalResolution office.visio.component.image.defaultResolutionプロパティを設定し、Visioドキュメントのページ画像のデフォルト解像度を指定します。このプロパティは、JPEG、GIF、PNG、BMPの画像フォーマットに適用されます。

OfficeServiceとChartingServiceによる並列要求のスロットル機構

OfficeServiceとChartingServiceが並列に要求を行えるよう設定するには、次の2つのプロパティファイルを変更します。

- "ChartingPool.properties"
- "OfficePool.properties"

"ChartingPool.properties"の設定

1. [QPP Server]/confフォルダにある"ChartingPool.properties"ファイルを開きます。
2. chartingThread.pool.maxActiveプロパティの設定により、いつでもアクティブにできるスレッドの最大数を指定します。
3. chartingThread.pool.whenExhaustedActionプロパティの設定により、プールのサイズが限界に達したときどのような動作を行うかを指定します。
   - 0に設定すると、NoSuchElement例外をスローしてただちに戻ります。
   - 1に設定すると、maxWaitの値に応じて、スレッドが利用可能になるまでしばらくブロックします。
   - 2に設定すると、プールのサイズを増やし、新しいスレッドインスタンスを返します。
4. chartingThread.pool.maxWaitプロパティの設定により、スレッドが利用可能になるのを待つ時間をミリ秒単位で指定します。-1に設定すると、無期限に待ちます。
5. chartingThread.pool.maxIdleプロパティの設定により、プール内のアイドルスレッドの最大数を指定します。
6. chartingThread.pool.minIdleプロパティの設定により、プール内のアイドルスレッドの最小数を指定します。
7. chartingThread.pool.minEvictableIdleTimeMillisプロパティの設定により、プール内のスレッドが排除されるまでの最小時間をミリ秒単位で指定します。
8. chartingThread.pool.timeBetweenEvictionRunsMillisプロパティの設定により、排除実行スレッドによりアイドルスレッドが削除されるまでの時間をミリ秒単位で指定します。-1に設定すると、削除は行われません。

"OfficePool.properties"の設定

1. [QPP Server]/confフォルダ内の"OfficePool.properties"ファイルを開きます。
2. officeThread.pool.maxActiveプロパティの設定により、同時にアクティブにできるスレッドの最大数を指定します。
3. officeThread.pool.whenExhaustedActionプロパティの設定により、プールのサイズが限界に達したときどのような動作を行うかを指定します。
   - 0に設定すると、NoSuchElement例外をスローして、ただちに戻ります。
   - 1に設定すると、maxWaitの値に応じて、スレッドが利用可能になるまでしばらくブロックします。
   - 2に設定すると、プールのサイズを増やし、新しいスレッドインスタンスを返します。

4 officeThread.pool.maxWaitプロパティの設定により、スレッドが利用可能になるまで待つ時間をミリ秒単位で指定します。-1に設定すると、無期限に待ちます。

5 officeThread.pool.maxIdleプロパティの設定により、プール内のアイドルスレッドの最大数を指定します。

6 officeThread.pool.minIdleプロパティの設定により、プール内のアイドルスレッドの最小数を指定します。

7 officeThread.pool.minEvictableIdleTimeMillisプロパティの設定により、プール内のスレッドが排除されるまでの最小時間をミリ秒単位で指定します。

8 officeThread.pool.timeBetweenEvictionRunsMillisプロパティの設定により、排除実行スレッドによりアイドルスレッドが削除されるまでの時間をミリ秒単位で指定します。-1に設定すると、削除は行われません。

QuarkXPress Serverフィルタ

QuarkXPressフィルタの設定を調整するには、以下の手順に従ってください。

1 [QPP Server]/confフォルダ内の"QxpsFilterServiceConfig.properties"ファイルを開きます。

2 qxpsfilter.useSpreadForArticles=値をtrueに設定してアーティクルのプレビューが各スプレッドで生成されることを指定します。

3 qxpsfilter.useSpreadForProjects=値をtrueに設定してプロジェクトプレビューにスプレッドを使用することを指定します。

4 デフォルトでは、XML分解は有効になっています。qxpsfilter.generateDeconstructedXML=エントリーのデフォルト設定はtrueです。XML分解を無効にするには、このエントリーをfalseに設定します。

5 デフォルトでは、全文検索（FTS）は有効になっています。qxpsfilter.generateText=エントリーのデフォルト設定はtrueです。FTSを無効にするには、このエントリーをfalseに設定します。

➡ qxpsfilter.generateDeconstructedXMLをtrue値に設定することによっても、QuarkXPressプロジェクトとQuarkCopyDeskアーティクルのFTSが有効になります。テキスト生成はXML生成に依存します。generateDeconstructedXMLをfalseに設定する場合、qxpsfilter.generateTextもfalseに設定することをお勧めします。

6 プレビュー画像の拡大縮小率を指定するには、qxpsfilter.previewScale=値を調整します。デフォルト値の1は、100パーセントでプレビューが表示されることを示します。2、3、6の値はそれぞれ200パーセント、300パーセント、600パーセントを示します。

7 サムネール画像の拡大縮小率を指定するには、qxpsfilter.thumbnailScale=値を調整します。デフォルト値の1は、100パーセントでプレビューが表示されることを示します。.5、2、3、6の値はそれぞれ50パーセント、200パーセント、300パーセント、600パーセントを示します。

8 JPEGプレビューの画質を指定するには、qxpsfilter.jpegQuality=値を調整します。最高画質は1、高画質は2、中画質は3、最低画質は4を入力します。

9 "QxpsFilterServiceConfig.properties"を保存して閉じます。

JAWSフィルタ設定

JAWSフィルタはEPSファイル、PDFファイル、Adobe® Illustrator®ファイル用のImageMagick® Filter Service Configurationsの一部です。JAWSフィルタの設定を指定するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダ内の"JawsFilterServiceConfig.properties"ファイルを開きます。
2. サムネールとプレビューの解像度をインチあたりのドット数で指定するには、jawsfilter.resolution=を調整します。

➡ jawsfilter.resolution=値は、正しい解像度が計算できなかった場合にプレビュー画像を特定のサイズに拡大縮小する代替解像度です。

3. "JawsFilterServiceConfig.properties"を保存して閉じます。

XML Authorフィルタ設定

XML Authorフィルタは、XMLファイルの設定の一部です。XADocFilterは、XMLファイルのプレビューとサムネールを生成するために使用されます。XML Authorフィルタの設定を指定するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダにある"XADocFilterServiceConfig.properties"ファイルを開きます。
2. XADocFilterによる索引付けの完了までのプロセス待機時間を指定するには、xaDocfilter.processTimeOut=の値を調整します。
3. プレビューの高さと幅を指定するには、xaDocfilter.previewHeight=とxaDocfilter.previewWidth=の値を調整します。
4. プレビューを生成するページの最大数を指定するには、xaDocfilter.previewPages=の値を調整します。
5. "XADocFilterServiceConfig.properties"を保存して閉じます。

ImageMagick、Jaws、およびDITA OTディレクトリ

ImageMagick、Jaws、DITA Open Toolkitは、デフォルトでQuark Publishing Platformに付属しています。だたし、ImageMagickまたはJawsが既にインストールされている場合には、以下の手順に従ってください。

1. [QPP Server]/confフォルダにある"ServerApp.properties"ファイルを開きます。
2. IMAGE_MAGICK_HOME=エントリまでスクロールします。
3. 既存のImageMagickのbinフォルダへのパスを入力します。
4. JAWS_HOME=エントリまでスクロールします。
5. 既存のJawsのbinフォルダへのパスを入力します。
6. DITA_HOME=エントリまでスクロールします。
7. 既存のDITA Open Toolkitフォルダへのパスを入力します。
8. "ServerApp.properties"を保存して閉じます。

全文索引の設定

FTS索引フィルタを設定するには、以下の手順に従ってください。

1. [QPP Server]/confフォルダにある"LuceneTextIndexingConfig.properties"ファイルを開きます。
2. デフォルトでは、索引ファイルのストレージ場所はQuark Publishing Platformインストールフォルダです。しかし、lucene.index.dir=パラメータを変更して索引ファイルを保存するフォルダを指定することによって場所を変更できます。
3. lucene.analyzerClass=パラメータを変更して、テキスト索引と検索語分析に最も頻繁に使用する言語（クラス）を指定します。デフォルト設定であるStandardAnalyzerを指定すると、全文検索の意味認識に英語が使用されます。しかし、ワークフローのテキストの大半が英語でない場合は、言語の意味論に従ってより正確に検索できるように別の言語を指定できます。言語のリストは"LuceneTextIndexingConfig.properties"ファイルにあります。

➡ "LuceneTextIndexingConfig.properties"には各パラメータの情報とApache文書へのリンクが含まれます。

4. 索引の圧縮が必要になる変更数を指定するには、lucene.maxModificationWithoutOptimisationの値を設定します。
5. "LuceneTextIndexingConfig.properties"を保存して閉じます。

チャート作成サービス

チャート作成サービスのプロパティを設定するには、下記のプロパティを変更します。

1. charting.defaultOutputFormatパラメータに、デフォルトの出力フォーマット（PDF／HTML／IMAGE／JSON）を入力します。
2. 次のパラメータに、画像出力プロパティのデフォルト値を入力します。
   - charting.defaultImageFormat=png
   - charting.defaultWidth=600
   - charting.defaultScale=
3. デフォルトのHTMLテンプレートへのパスを入力します。
   charting.defaultHTMLTemplateURI=classpath¥:DefaultHTMLTemplate.html

Quark Publishing PlatformへのQLAの統合

Quark® License Administrator（QLA）プライマリサーバーおよびバックアップサーバーをQuark Publishing Platformで再設定するには、下記の手順に従ってください。

1. [QPP Server]/confフォルダ内の"Qla.properties"ファイルを開きます。
2. QlaServer.machinename=パラメータにQLAをインストールしたコンピュータの現在のIPアドレスとホスト名を入力します。

3 QLAServer.port=パラメータにポート番号を入力します。

4 バックアップQLA Serverがある場合には、Backup.QlaServer.machinename=パラメータとBackup.QlaServer.port=パラメータに現在のIPアドレスまたはホスト名とポート番号を入力します。

5 Qla.SerialNumber=パラメータにQuark Publishing Platform ServerのQLAシリアル番号を入力します。シリアル番号は、QLA Server ConsoleアプリケーションとQLA Clientアプリケーションに表示されています。

➡ "Qla.properties"内のシリアル番号は、Quark Publishing Platform Serverをインストールまたは更新したときに入力したバリデーションコードに基づいて更新されます。

6 "Qla.properties"を保存して閉じます。

動的設定

設定サービスは、パブリッシング設定の管理に使用されます。

- 新しいパブリッシングチャンネルの追加、または更新
- 新しいパブリッシングプロセスの追加、または更新
- コンテンツタイプの新しいMIMEタイプの追加、または更新

設定サービスは、次の目的に使用されます。

- 設定ファイルの管理
- 更新のキャプチャ
- 対応するビーン定義を再初期化し、再起動を回避する
- APIの再初期化を使用し、設定ファイルがサーバー上で手作業により変更された後でも再起動を回避する

マルチサーバー環境では、クラスタの各ノードが単一の.extフォルダへアクセスでき、共有の場所がclassptheシステム環境変数に追加される必要があります。

カスタムXML MIMEタイプの作成

https://http://localhost:61400/rest/service/config/xmlmimetypes?
op=create&mimetype=application/xml;
format= researchreport&parentmimetype=application/xml;
format= smartcontent&xpath =/*[local-name()='section']/@type=' researchreport

XML MIMEタイプからPlatformのコンテンツタイプへのマッピング

https://http://localhost:61400/rest/service/config/xmlmimetypes?
op=mapcontenttype&contenttype=Research Report&mimetypes=application/xml;
format=researchreport

パブリッシングチャネルの作成

https://http://localhost:61400/rest/service/config/publishingchannels?
op=create&publishingchannellist=<publishingChannelList><publishingChannelInfoid="researchReportPdf"

アセット索引付け用のパブリッシングチャンネルの定義

https://http://localhost:61400/rest/service/config/publishingchannels?contenttype=Research Report&loginname=Admin&loginpassword=Admin

IPTCサポートの有効化

画像ファイル内のIPTC値を識別して、画像の属性リストに含めることができます。IPTC機能を有効化または無効化するには、¥Server¥conf¥IMFilterServiceConfig.propertiesファイルを開いて、imfilter.generateIPTCAttributesプロパティをtrueまたはfalseに設定します。imfilter.resolutionの値を設定することによって、プレビュー／サムネールの解像度を指定することもできます。

RMIクライアントおよびCORBAクライアントのみ

デフォルトでは、Quark Publishing Platformクライアントで必要なポートは61400のみです。ただし、RMIクライアントやCORBAクライアントを使用または開発する場合は、下記のトピックを参照することをお勧めします。

Quark Publishing Platform Server使用ポートの変更

デフォルトでは、Quark Publishing Platformクライアントで必要なポートは61400のみです。ただし、RMIクライアントやCORBAクライアントを使用または開発する場合は、Quark Publishing Platform環境内のサーバーとファイルストリーミングのポート設定を調整できます。デフォルトでは、Quark Publishing Platform Serverのサービスは61400から61407までのネットワークポート上で提供されますが、これらのポートをすべて設定できます。

デフォルトポートの変更

デフォルトでは、Quark Publishing Platformクライアントで必要なポートは61400のみです。ただし、RMIクライアントやCORBAクライアントを使用または開発する場合は、下記の手順に従ってポートを変更します。

1. [QPP Server]/confフォルダにある"ServerApp.properties"ファイルを開きます。
2. Quark Publishing Platform Script ManagerなどのJavaベースのRMIクライアントを実行するポートを指定するには、rmi.port=プロパティに値を入力します。Remote Method Invocation（RMI）は、Javaオブジェクトをネットワーク上でリモートに実行するための標準的なリモートプロシージャコールです。
3. サービスをバウンドするポートを指定するには、rmi.serviceport=プロパティに値を入力します。
4. IIOP要求を管理するポートを指定するには、namingservice.port=プロパティに値を入力します。Internet Inter-Orb Protocol（IIOP）は、TCP/IPネットワーク上のオブジェクトを操作するためのメッセージプロトコルです。QuarkXPress、QuarkCopyDesk、およびQuark Publishing Platform Clientはこのポートを通じて接続されます。
5. CORBAオブジェクトをエクスポートするポートを指定するには、serverORB.port=プロパティに値を入力します。

6 すべてのQuark Publishing Platformクライアントアプリケーション通信のポートを指定するには、jms.openWirePort=プロパティに値を入力します。Quark Publishing Platform環境では、Java Messaging Service（JMS）にはOpenWire®プロトコルが使用されます。

7 Tomcatサーバーを実行するポートを指定するには、webServer.port=プロパティに値を入力します。Quark Publishing Platform Web Clientにはこのポートを通じて接続されます。

8 "ServerApp.properties"を保存して閉じます。

Tomcatサーバーポートの指定

デフォルトポートを変更した後、Tomcatサーバーを実行するポートも別のファイルで指定する必要があります。下記の手順に従ってください。

1 [QPP Server]/confフォルダにある"server.xml"ファイルを開きます。

2 Webサーバーポートを変更します。

3 "server.xml"を保存して閉じます。

4 サーバーを再起動します。

ファイルストリーミングポートの変更

ファイルストリーミングポートを変更するには、下記の手順に従ってください。

1 Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。

2 "SocketStreaming.properties"をテキスト編集アプリケーションで開きます。

3 ファイルのアップロードとダウンロードに使用するポートを指定するには、socketStreaming.port=プロパティに値を入力します。

4 "SocketStreaming.properties"を保存して閉じます。

複数のネットワークカード

デフォルトでは、Quark Publishing Platformクライアントで必要なポートは61400のみです。ただし、RMIクライアントやCORBAクライアントを使用または開発する場合や、Quark Publishing Platform Serverを実行しているコンピュータに複数のネットワークインターフェイスカードがある場合には、Quark Publishing Platform Serverに特定のカードまたはコンピュータ上のすべてのIPアドレスをバインドできます。

特定のIPアドレスへのバインド

特定のIPアドレスにバインドするには、下記の手順に従ってください。

1 Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。

2 "ServerApp.properties"をテキスト編集アプリケーションで開きます。

3 server.machinename=エントリーまでスクロールします。バインドするネットワークカードのIPアドレスを入力します。

4 server.bindtoip=エントリーまでスクロールします。trueと入力します。これによりQuark Publishing PlatformServerサーバーは「server.machinename=」エントリーで指定されたIPアドレスと名前にのみバインドされます。

5 "ServerApp.properties"を保存して閉じます。

"server.xml"の編集

"server.xml"も編集する必要があります。下記の手順に従ってください。

1 Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。

2 "server.xml"をテキスト編集アプリケーションで開きます。

3 「connector」タグにIPアドレスを追加します。たとえば、<Connector port=61400を<Connector address = "<IP Address>" port="61400"に変更します。

4 "server.xml"を保存して閉じます。

単一コンピュータ上のすべてのIPアドレスへのバインド

単一コンピュータ上のすべてのIPアドレスにバインドするには、下記の手順に従ってください。

1 Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。

2 "ServerApp.properties"をテキスト編集アプリケーションで開きます。

3 server.machinename=エントリーまでスクロールします。localhostと入力します。

4 server.bindtoip=エントリーまでスクロールします。falseと入力します。これによりQuark Publishing Platform Serverはコンピュータ上のすべてのIPアドレスにバインドされます。

5 server.additionalnames=エントリーまでスクロールします。バインドするネットワークカードのIPアドレスを入力します。複数のIPアドレスがある場合には、エントリーをカンマで区切ります。たとえばserver.additionalnames= 10.91.43.266,10.X.Y.Zとなります。ネットワークカードが1つだけしかない場合には、このフィールドは空白のままにします。

➡ このリストには、デフォルトでないIPのみを含めます。デフォルトIPアドレスは自動的に使用されるため、デフォルトIPはここで指定しないでください。ここにデフォルトIPを追加することはお勧めしません。

6 "ServerApp.properties"を保存して閉じます。

NATを使用したファイアウォール

RMIクライアントやCORBAクライアントを使用または開発する場合に、Network Address Translation（NAT）サービスを実行しているファイアウォールを通じてインターネット上でQuark Publishing Platform Serverにアクセスできるようにするには、Quark Publishing Platform ServerのプライベートIPアドレスをマップするパブリックIPアドレスを指定する必要があります。

パブリックIPアドレスを指定するには、下記の手順に従ってください。

1 Quark Publishing Platform Serverフォルダ内の"conf"フォルダを開きます。

2 "ServerApp.properties"をテキスト編集アプリケーションで開きます。

3 server.additionalnames=エントリーまでスクロールします。

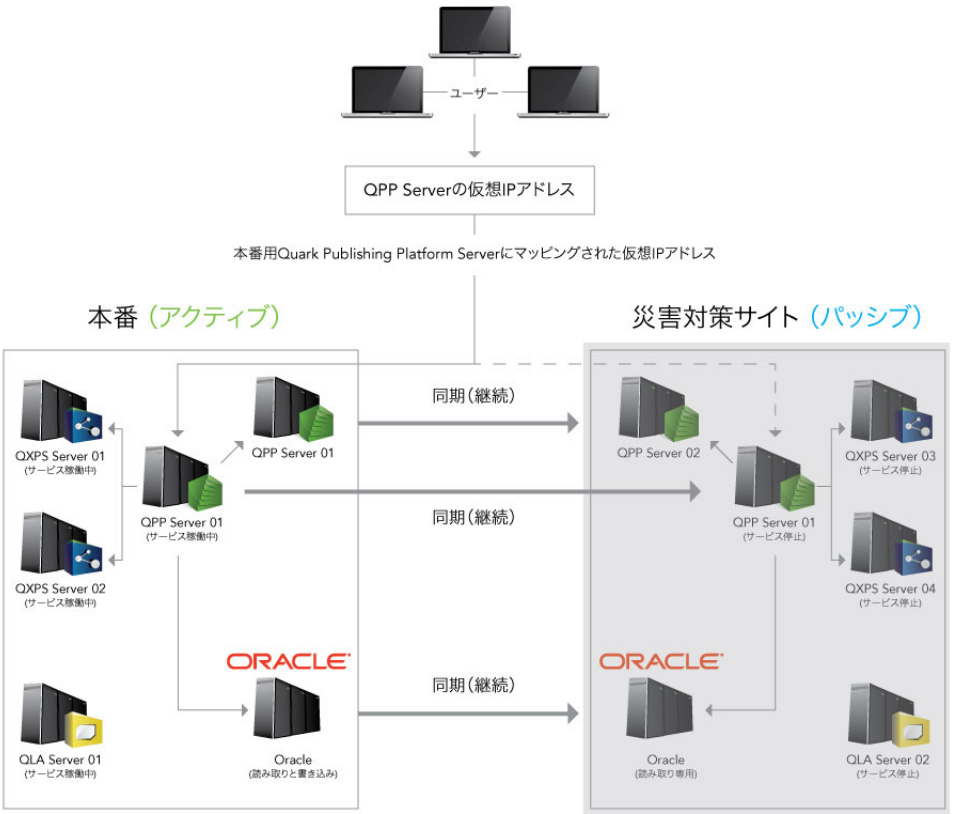
4 Quark Publishing Platform ServerのプライベートIPアドレスをマップするパブリックIPアドレスを入力します。

5 "ServerApp.properties"を保存して閉じます。

フェイルオーバー設定

このトピックは、Quark Publishing Platformインストールにおけるフェイルオーバーの設定方法を説明します。

Quark Publishing Platform Serverコンピュータにおける仮想IPアドレスの使用が、このような設定の鍵となります。



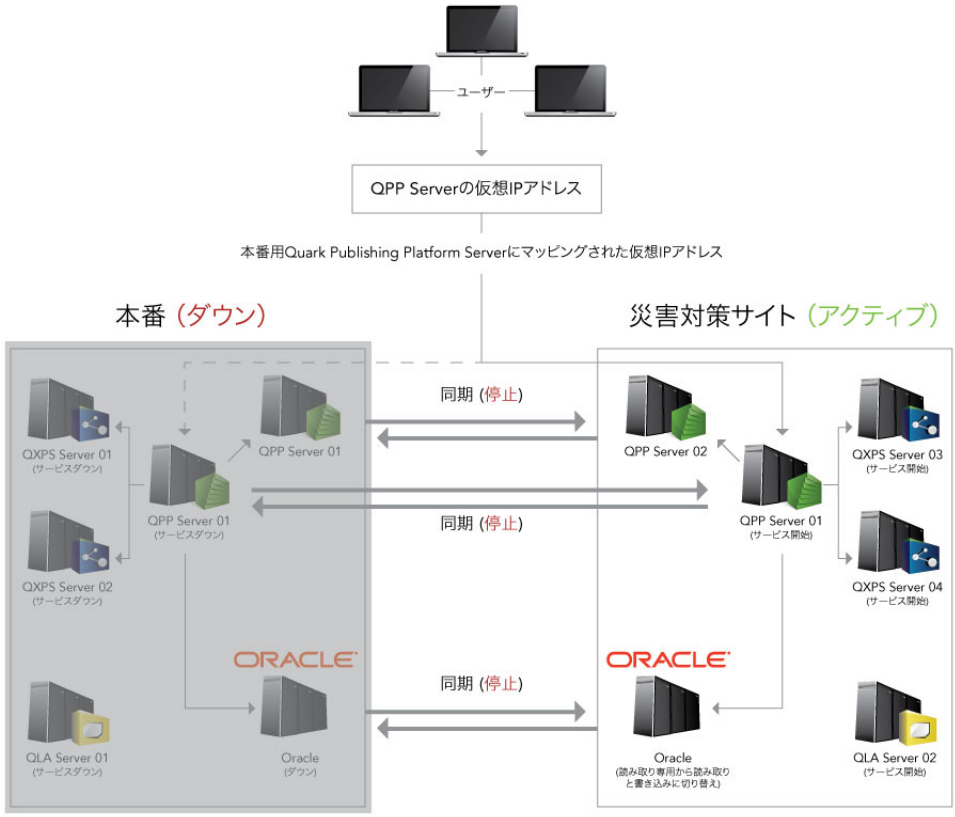
フェイルオーバー設定：通常動作

通常動作では、実稼働サイトが作動し、仮想IPアドレスがそのサーバーにマップされます。DR（障害復旧）サイトは休止モードとなり、すべてのQuark Serviceが停止します。実稼働サーバーからDRサーバーに対して,下記の項目が同期されています。

- Quark Publishing Platformリポジトリフォルダ
- Quark Publishing Platform データベース
- QPP Server/confフォルダ
- QPP Server/Indexフォルダ

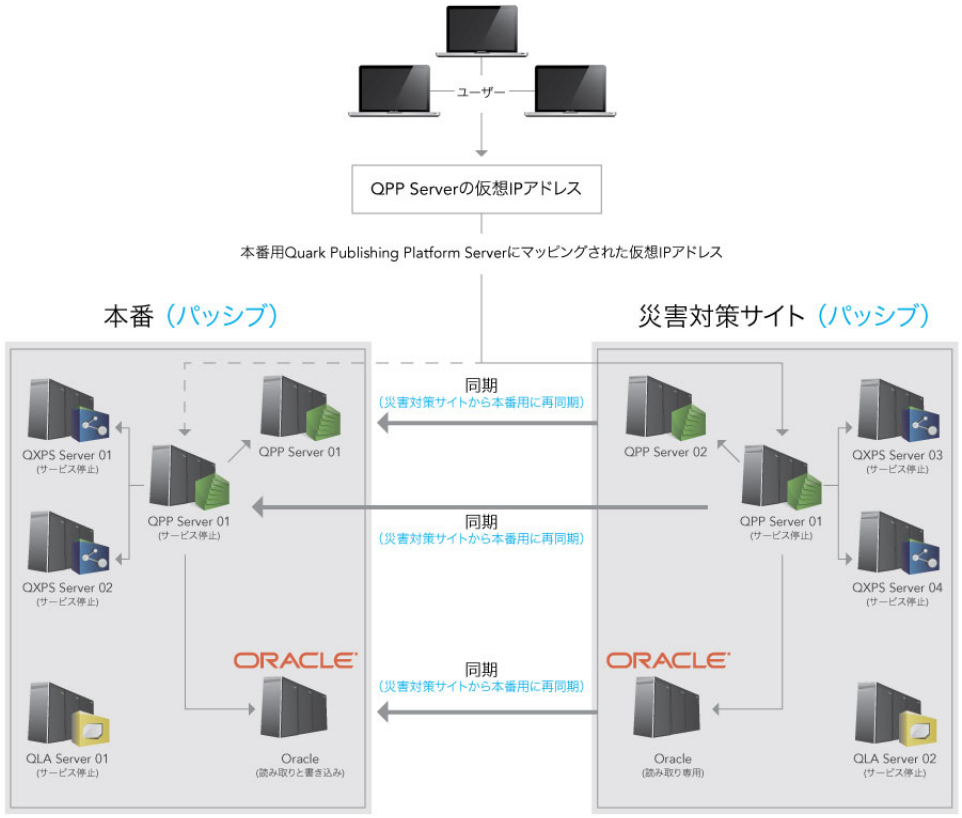
Quark Publishing Platform File Server フォルダおよびQuark Publishing Platform Oracleデータベースの同期は、データ整合性を維持するため、同じサイクルで同時に行う必要があります。

実稼働環境がダウンした場合、同期は停止し、仮想IPアドレスがDRサイトに再度マップされます。エンドユーザーは、同じ仮想IPアドレスを使用しているため、変更は通知されません。



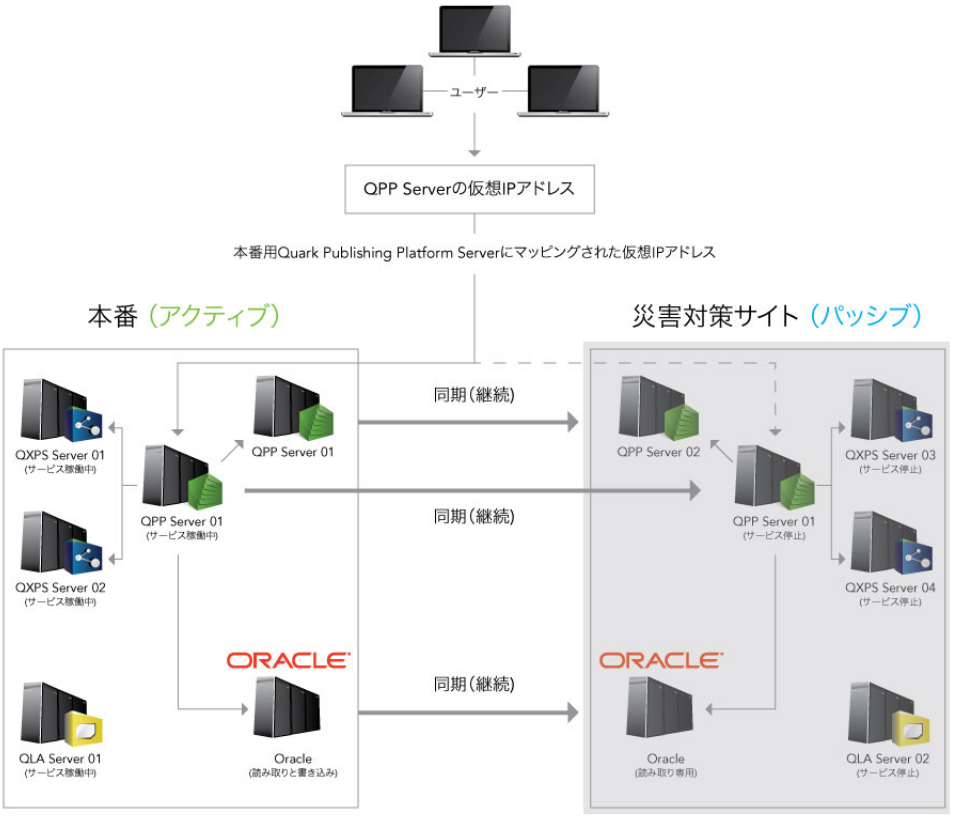
フェイルオーバー設定：実稼働サーバーのダウン

この場合、管理者はDRサイトを正しい順序でシャットダウンし、DRサイトから実稼働サイトへ再度同期を行う必要があります。



フェイルオーバー設定：DRサイトから実稼働サイトへの同期

DRサイトと実稼働サイト間の同期が完了したら、仮想IPアドレスを実稼働サイトに再度マップし、同期を反転して実稼働サイトからDRサイトへ変更が適用されるようします。次に、すべてのQuark Publishing Platformコンポーネントを起動します。



フェイルオーバー設定：通常動作の復元

プレーンテキストパスワードの暗号化

Quark Publishing Platform 10.1およびそれ以降では、プレーンテキストのパスワードをすべて暗号化できます。たとえば、SQL ServerまたはOracleをPlatform Severのデータベースとして使用している場合、データベースのユーザーパスワードは"database.properties"ファイルにプレーンテキストで保存されます。このパスワードを暗号化できます。

プレーンテキストのパスワードを暗号化するには、下記の手順に従ってください。

1. コマンドプロンプトを開き、Platform Serverのインストールフォルダに移動します。
2. 自分のパスワードで"encrypt.bat"ファイルを起動し（encrypt.bat password）、暗号化された値を取得します。
3. パスワードに対して生成された暗号化済みの値をコピーし、その暗号化された値を、いずれかのプロパティファイル（たとえば、"database.properties"）に使用します。
4. Quark Publishing Platform Serverフォルダ内の"lib"フォルダを開きます。
5. "qpp-server-app-11.2.jar"を開き、jarファイル内の¥com¥quark¥qpp¥appに移動します。

6 テキスト編集アプリケーションで"ServerStartupContext.xml"を開き、securePlaceholderConfigノードの下に、暗号化した値を使用した各プロパティファイルに対応するエントリを追加します。このエントリがplaceholderConfigノードに存在する場合、このノードからは削除します。

7 ファイルを保存し、jarを更新して、サーバーを再起動します。

➡ "ServerStartupContext.xml"ファイルに記載されているアルゴリズムはどれでも使用でき、パスワードの暗号化に使用されているキーも変更できます（デフォルトはQUARKです）。

非アクティブ時の強制ログオフの有効化

WebAdminに強制ログオフを許可する設定

強制ログオフを有効にするには、[QPP Server]/webapps/adminフォルダの"Web.xml"ファイルで定義済みのサーブレットフィルタを有効にします。

- session-idle-time： ユーザーがこの時間（秒単位）だけ非アクティブ状態であると、WebAdminにより強制ログオフされます。
- exclude-url-patterns： ユーザーのアクティビティとはみなされないURLパターンを指定します。
- pre-expiry-mag-time： 非アクティブ状態によるログオフが保留中のとき、この時間（秒単位）だけログオフより前に、警告またはカウントダウンメッセージが表示されます。

```
<filter> <filter-name>SessionTimeoutCookieFilter</filter-name>
<filter-class>com.quark.web.activity.servlet.SessionTimeoutCookieFilter</filter-class>
<init-param> <param-name>session-idle-time</param-name>
<param-value>60</param-value> </init-param> <init-param>
<param-name>exclude-url-patterns</param-name>
<param-value>/admin/keepAlive.qsp</param-value> </init-param> <init-param>
<param-name>pre-expiry-mag-time</param-name> <param-value>40</param-value>
</init-param> </filter> <filter-mapping>
<filter-name>SessionTimeoutCookieFilter</filter-name> <url-pattern>/*</url-pattern>
</filter-mapping>
```

QPP Home/webapps/admin/jspフォルダの"Admin-Home.jsp"ファイルでも、強制ログオフを有効にできます。

```
<!-- Uncomment for "Forced Logoff on Inactivity" --> <script type="text/javascript"
src="resources/js/SessionTimer.js></script> <link type="text/css"
href="resources/js/SessionTimer.css re;="stylesheet></link>

function logoutDueToInactivity () { window.location.href = "logout.qsp"; }; <%-- Initialise
 inactivity monitor --%> SessionTimer.init(logoutDueToInactivity);
```

ワークスペースで強制ログオフを許可する設定

強制ログオフを有効にするには、[QPP Server]/webapps/workspaceフォルダの"Web.xml"ファイルで定義済みのサーブレットフィルタを有効にします。

- session-idle-time： ユーザーがこの時間（秒単位）だけ非アクティブ状態であると、WebAdminにより強制ログオフされます。

- exclude-url-patterns： ユーザーのアクティビティとはみなされないURLパターンを指定します。
- pre-expiry-mag-time： 非アクティブ状態によるログオフが保留中のとき、この時間（秒単位）だけログオフより前に、警告またはカウントダウンメッセージが表示されます。

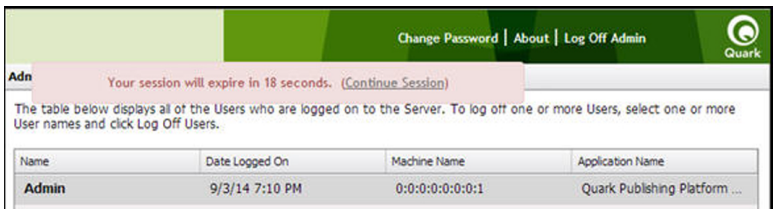
```
<filter> <filter-name>SessionTimeoutCookieFilter</filter-name>
<filter-class>com.quark.web.activity.servlet.SessionTimeoutCookieFilter</filter-class>
<init-param> <param-name>session-idle-time</param-name>
<param-value>300</param-value> </init-param> <init-param>
<param-name>exclude-url-patterns</param-name>
<param-value>/workspace/keepAlive.qsp/workspace/assetHeaderUpdate.qsp</param-value>
 </init-param> <init-param> <param-name>pre-expiry-mag-time</param-name>
<param-value>60</param-value> </init-param> </filter> <filter-mapping>
<filter-name>SessionTimeoutCookieFilter</filter-name> <url-pattern>/*</url-pattern>
</filter-mapping>
```

QPP Home/webapps/workspace/jspフォルダの"User-Home.jsp"ファイルでも、強制ログオフを有効にできます。

```
<!-- Uncomment for "Forced Logoff on Inactivity" --> <script type="text/javascript"
src="resources/js/SessionTimer.js></script> <link type="text/css"
href="resources/js/SessionTimer.css rel="stylesheet></link>

<%-- Initialise inactivity monitor --%> SessionTimer.init(UserHomeUI.logout);
```

カウントダウンメッセージには、残りの秒数が表示されます。 ユーザーは**セッションの続行**をクリックして強制ログオフを防止できます。



強制ログオフのカウントダウン

メッセージングの設定

Quark Publishing Platform Serverで、関連するアセットにおいてリストされている属性のいずれかが変更されたとき、Asset Workflow Changedメッセージを表示するよう設定するには、下記の手順に従ってください。

1. [install_path]/Server/conf/フォルダにある"AssetWorkflowChangedMessageAttributes.xml"ファイルを開きます。
2. 関連するアセットにおいて、リストされている属性のいずれかが変更されたとき、Asset Workflow Changedメッセージをトリガする属性IDまたは名前を設定します。

```
<!-- List of attribute ids or names to be considered for "ASSET WORKFLOW CHANGED
MESSAGE" The "ASSET WORKFLOW CHANGED MESSAGE" will be published if there is any
 changes in the attributes of the mentioned asset/> <util:list id="assetWorkflowAttributes">
 <value>name</value> <value>Status</value> <value>Workflow</value> <value>Routed
 to</value> </util:list> </beans>
```

Quark Publishing Platform Web Client： 手動設定

以下のセクションでは、Quark Publishing Platform Web Clientの詳細設定を行う方法について解説します。

設定の概要

ワークスペース設定は、2つのカテゴリに分類されます。

- アプリケーションレベルの設定
- ドキュメントレベルの設定

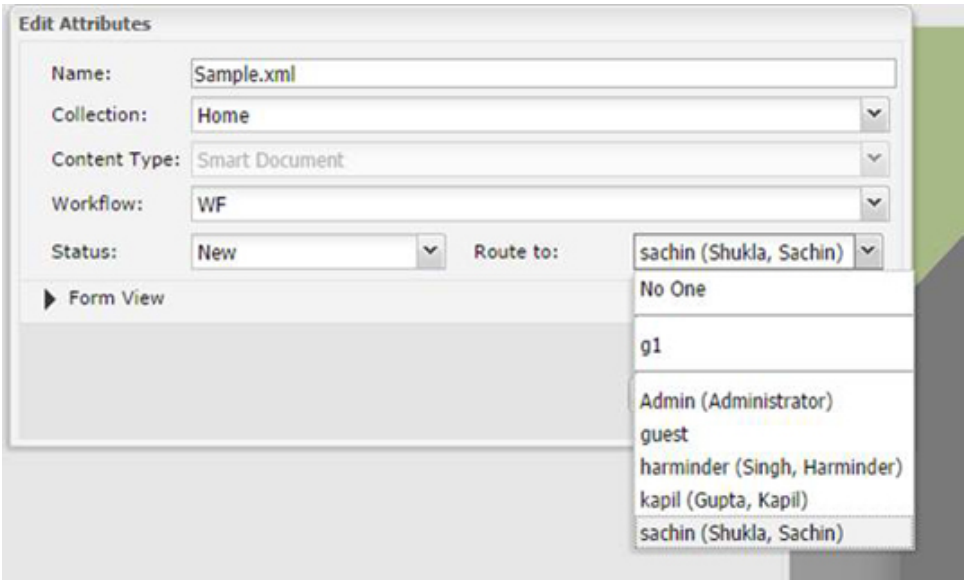
これらは、[QPP Server]/ webapps /workspace/WEB-INF/classesフォルダにある"Workspace-Config.xml"設定ファイルにより設定されます。

アプリケーションレベルの設定

ApplicationSettingsの属性要素は、アプリケーションレベルの各種の設定を定義します。

- userNameFormatting： すべてのダイアログに表示される**ユーザー名**の形式を指定します。

1. 0は、ログイン名<username>を表示します。
2. 1は、<名> <姓>を表示します。
3. 2は、<姓> <名>を表示します。



- layoutPreview： QuarkXPressのコンテンツタイプに属するアセットの、プレビューウィンドウの外観と操作性を指定します。

1. Trueは、プレビューを別のブラウザウィンドウで表示します。
2. Falseは、プレビューをブラウザの代わりにjavascriptウィンドウで表示します。

- viewrevision_expandAll： リビジョンの表示ダイアログで、すべてのリビジョンコメントを展開する必要があるかどうかを指定します。
- supported_picture_extension： アーティクル／プロジェクトエディタへ画像／テキストを取り込むときに許容される、可能なファイルタイプを指定します。
- picture_editing_extension： アーティクル／プロジェクトエディタで画像を編集するときに許容される、可能なファイル形式を指定します。
- defaultPreviewScale： アーティクル／プロジェクトの編集およびアセットの表示（ライブプレビュー）を行うときに使用される、デフォルトのズームプレビュー設定を指定します。 このプロパティの有効な値の範囲は0.1～5です。
- ajaxTimeout： ブラウザから発生するすべてのバックグラウンド要求について、制限時間を指定します。この時間が経過した要求はタイムアウトとしてマークされます。 値はミリ秒単位です。
- showFormView： **チェックイン**ダイアログのフォームビューパネルを表示するかどうかを指定します。 trueに設定すると、フォームビューパネルが表示されます。
- topBannerJspPath： 割り当てられたページのトップバナーを表示するために含める.jspファイルへのパスを指定します。
- logoFilePath： 割り当てられたページのトップバナーに表示されるロゴ画像へのパスを指定します。
- enabledPublishingChannels： アセットを選択するとき優先する必要があるパブリッシングチャンネルのリストを定義します。
- enableDeliveryChannels： 配信チャンネルを有効にする必要があるかどうかを定義します。

- enabledDeliveryChannels：アセットを選択するとき優先する必要がある配信チャンネルのリストを定義します。 これは、enableDeliveryChannelsオプションがtrueに設定されているとき有効になります。
- allowPublishedRenditionDownload： 選択したアセット用に定義されているパブリッシングを使用して、各種のプレビューに表示されるダウンロード出力を許可するかどうかを定義します。

```
<ApplicationSettings> <Add key="viewrevision_expandAll" value="false"/> <Add
key="supported_picture_extension" value="bmp;jpg;jpeg;tif;tiff;gif;"/> <Add
key="picture_editing_extension" value="jpg;jpeg;tif;tiff;eps"/> <Add
key="userNameFormatting" value="0"/> <Add key="layoutPreview" value="true"/> <Add
key="defaultPreviewScale" value="0.8"/> <Add key="ajaxTimeout" value="300000"/> <Add
 key="showFormView" value="false"/> <Add key="topBannerJspPath" value="Header.jsp"/>
 <Add key="logoFilePath" value="images/login/topbanner-innerpage-left.png"/> <Add
key="enabledPublishingChannels"
value="qxpPdf;qxpEpub;qxpAppStudio;qxpAppStudioPackage;busDocPdf;busDocHtml;busDocQxp/>
 <Add key="enableDeliveryChannels" value="false"/> <Add key="enabledDeliveryChannels"

value="checkInToSharepoint;checkInToFileNet;checkInToDocumentum;sendEmail;sendToFTPServer"/>
 <Add key="allowPublishedRenditionDownload" value="true"/> </ApplicationSettings>
```

マルチチャンネルのプレビュー

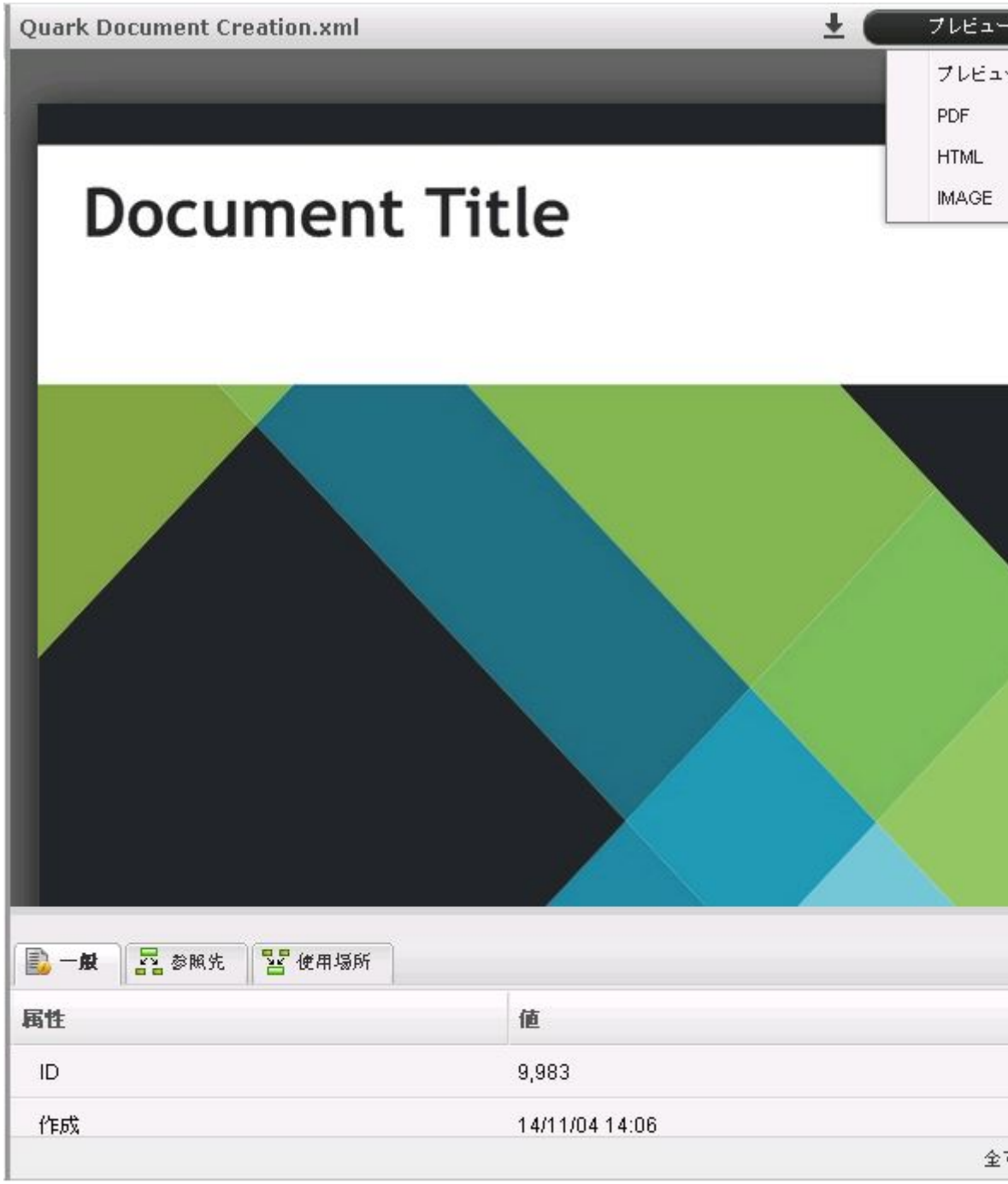
それぞれのコンテンツタイプについて、ユーザーが利用可能なパブリッシングチャンネルを、割り当てページの**プレビュー**タブで指定します。 プレビュー用のパブリッシングチャンネルは、<PreviewSettings>要素を使用して設定します。

- displayName： ユーザーインターフェイスに表示されるチャンネル名を指定します（オプション）。指定されていない場合、outputFormatが使用されます。
- Id： Platform Serverで定義されているパブリッシングチャンネルIDを指定します。
- ContentType： アセットのコンテンツタイプを指定します。
- ApplyToChildContentTypes： 子コンテンツタイプを含めるかどうかを指定します。
- outputFormat： 次の値がサポートされます。

1. IMAGE_ARCHIVE： 発行される出力のイメージアーカイブで、Webページ内にレンダリングされます。
2. HTML_ARCHIVE： 発行される出力のHTMLアーカイブで、変更なしにレンダリングされ、HTMLアーカイブ内に存在するファイル名を指し示します。
3. PDF_ARCHIVE： 発行される出力PDFで、変更なしにレンダリングされます。

- downloadChannel： （オプション）選択されたチャンネルプレビューをダウンロードするため、別のチャンネルを起動する必要があるときに使用します。

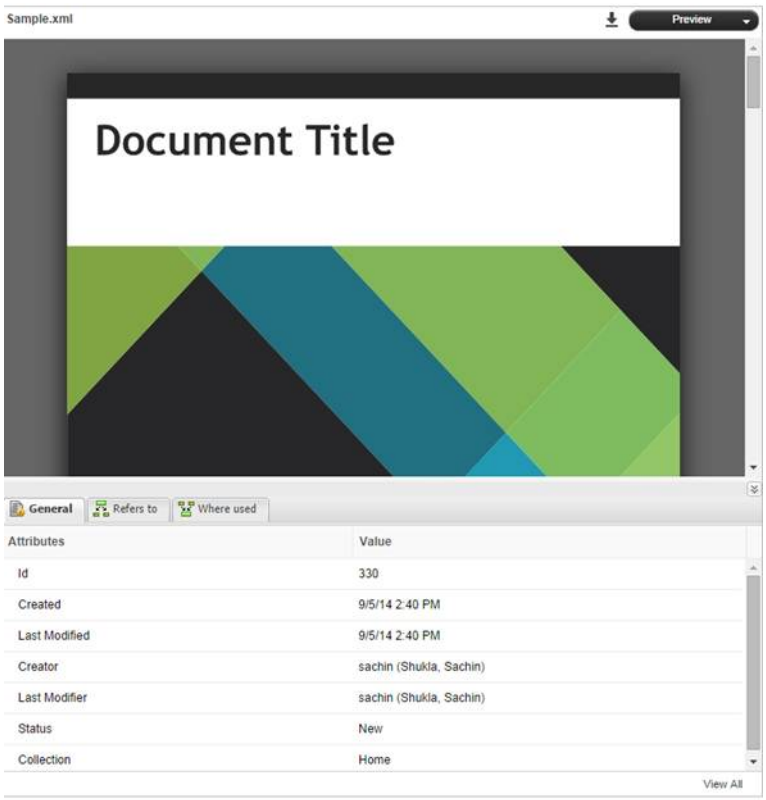
```
<PreviewSettings> <ChannelMappings> <ChannelMapping contentType ="Business
Document" applyToChildContentTypes ="true"> <Channels> <Channel id="busDocPdf"
outputFormat="PDF_ARCHIVE" displayName="PDF"/> <Channel id="busDocHtml"
outputFormat="HTML_ARCHIVE" displayName="HTML"/> <Channel id="busDocJpeg"
outputFormat="IMAGE_ARCHIVE" displayName="IMAGE"/> </Channels>
</ChannelMapping> <ChannelMapping contentType="Smart Content"
applyToChildContentTypes="true"> <Channels> <Channel id="smartDocPdf"
outputFormat="PDF_ARCHIVE" displayName="PDF"/> <Channel id="smartDocHtml"
outputFormat="HTML_ARCHIVE" displayName="HTML"/> <Channel id="smartDocJpeg"
outputFormat="IMAGE_ARCHIVE" displayName="IMAGE"/> </Channels>
</ChannelMapping> </ChannelMappings> </PreviewSettings>
```



全般ペインの属性

全般タブの属性は、<PreviewAttributes>要素を使用して設定されます。 これらの値は、選択したアセットについて**全般**タブに表示される属性のリストを指定します。

<PreviewAttributes> <PreviewAttribute>Id</PreviewAttribute>
<PreviewAttribute>Created</PreviewAttribute> <PreviewAttribute>Last modified</PreviewAttribute> <PreviewAttribute>Creator</PreviewAttribute>
<PreviewAttribute>Last modifier</PreviewAttribute>
<PreviewAttribute>Status</PreviewAttribute>
<PreviewAttribute>Collection</PreviewAttribute> ..... </PreviewAttributes>



ロールベースのツールバー設定

ここでは、特定のロール用のツールバーアイテムを定義する方法について説明します。ツールバーアイテムをロールベースで設定するには、<ToolbarConfig>要素を使用します。

- ロールの名前：設定を定義するロールの名前を指定します。
- アイテムID：外見の設定が必要なアイテムのIDを指定します。
- showInToolbar：ツールバーのメニューアイテムを隠すには、この属性をfalseに設定します。
- showInContext：コンテキストメニューのメニューアイテムを隠すには、この属性をfalseに設定します。
- showInNewMenu：新規アセットメニューのメニューアイテムを隠すには、この属性をfalseに設定します。
- showInTemplateMenu：メニューアイテムを新規アセットの作成テンプレートに表示しないようにするには、この属性をfalseに設定します。

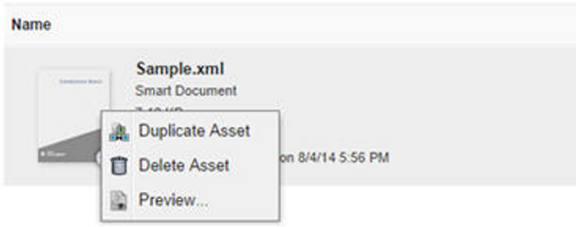
ツールバー設定のXML構造は次のとおりです。

```
<Role name ="[ROLE NAME]"> <Item id ="[ITEM ID]" showInToolbar ="[true/false (defaults to true)]" showInContextMenu ="[true/false (defaults to true)]" showInNewMenu ="[true/false (defaults to true)]" showInTemplateMenu ="[true/false (defaults to true)]"/> </Role>

<ToolbarConfig> <Role name ="Guest"> <Item id ="new_qcd_menu_item"/> <Item id ="new-search-btn"/> <Item id ="duplicate-asset" showInToolbar="false"/> <Item id
```

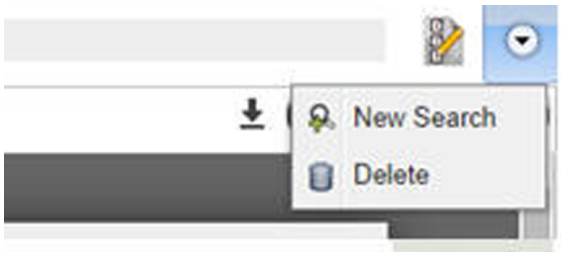
="delete-asset"/> <Item id ="show-edit-attributes" showInContextMenu="false"/> <Item id ="show-Asset_preview" showInToolbar="false"/> .....</Role> </ToolbarConfig>

ゲストロールのコンテキストメニューは次のようになります。



ゲストロールのコンテキストメニュー

ゲストロールのツールバーアイテムは次のようになります。



ゲストロールのツールバーアイテム

アイテムIDのリストは次のとおりです。

- checkin：アセットのチェックイン
- show-check-out：アセットのチェックアウト
- cancel-checkout：アセットのチェックアウトのキャンセル
- get-asset：アセットの取得
- new-search-btn：新しい検索（コンテキストメニューのアイテムとしては利用できません）
- show-Asset-Preview：アセットのプレビュー
- show-edit-attributes：アセットの属性の編集
- attachment-info：リンク情報の表示
- view-revisions：アセットのリビジョンの表示
- duplicate-asse：アセットの複製
- delete-asset：アセットの削除
- reindex-asset：アセットの索引の再作成（デフォルトではツールバーに表示されません）
- publish-menu-btn：アセットの発行
- deliver-menu-btn：アセットの配信
- restore：アセットの復元
- archive-asset：アセットのアーカイブ

- open-collection：アセットが属するコレクションを開く
- refresh-datadoc：datadocのリフレッシュ
- unlink-datadoc：datadocのリンクの解除
- open-readonly：アセットを編集不可で開く
- new_qcd_menu_item：新しいQuarkCopyDeskアーティクルの作成
- new_qxp_menu_item：テンプレートから新しいQuarkXpressProjectを作成

➡ showInToolbarおよびshowInContextMenu属性は、ツールバーおよびコンテキストメニューに表示されるメニューアイテムにのみ適用可能です。**ui-extension.xml**ファイルに設定されているボタンにはshowInToolbarを適用できません。この設定は、ツールバーのボタン自体を追加することが目的なためです。さらに、ボタンをコンテキストメニューに追加できるため、showInContextMenuのみが **ui-extension.xml**ボタンに適用可能です。showInNewMenuおよびshowInTemplateMenu属性は、**新規アセット**ドロップダウンメニューのメニューアイテムにのみ適用可能で、このセクションで**ui-extension.xml**メニュー用にロールベースの設定をセットアップするときには適用できません。

➡ "Workspace-config.xml"に指定される他の設定可能な要素は、Quark Author Web Editionとの統合に使用されます。詳細は、『Quark Author Web Edition Configuration Guide』を参照してください。

Web Client／Adminへのアクセスの制限

ワークスペースや管理ページが自社のネットワーク内からのみアクセス可能になるようにWebアプリを制限するには、Tomcat Valveを使用し、IPアドレス範囲に基づいてアクセスを制限します。

1. {install_path}/confにある"server.xml"ファイルを開きます。
2. RemoteAddrValveに関係する管理およびワークスペースアプリケーションに対応するコンテキストエレメントを編集します。イントラネットトラフィックのみを許容するように、valveを設定する必要があります。最終的な設定は次のようになります。

```
<Context path="/qxpsm" docBase="qxpsm"> <Manager pathname="" /> </Context>
<Context path="/workspace"docBase="workspace"> <Manager pathname="" /> <Valve
className="org.apache.catalina.valves.RemoteAddrValve" allow=" 10¥.91¥..*¥..* "/>
</Context> <Context path="/webservices"docBase="webservices"> <Manager pathname=""
/> </Context> <Context path="/messaging" docBase="messaging"> <Manager pathname=""
/> </Context> <Context path="/admin"docBase="admin"> <Manager pathname="" />
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="
10¥.91¥..*¥..*  " /> </Context> <Context path="/rest"docBase="rest"> <Manager
pathname="" /> </Context>
```

➡ allow属性の値は、一致するIPアドレスの正規表現になります。ここでは、イントラネットのIPが10.91.x.xの範囲であることを想定しています。この値は、お使いのネットワークIPの範囲に応じて変更する必要があります。

Quark Publishing Platform クライアント — 手動設定

インストール後に、Quark Publishing Platformクライアントおよび Platform XTensions のさまざまな設定オプションを変更できます。 詳細は、下記のトピックを参照してください。

ログファイルの作成と管理（Mac OS Xのみ）

ログファイルの作成と管理の設定は、com.quark.qpp.client.Quark.QPP.Client.config.plist ファイル内にあります。このファイルは次にあります。~/Library/Preferences¥Quark¥QPP¥{version}。

ログファイルの場所を変更するには、LogFileName属性を使用して新しいログファイルの場所を定義します。デフォルトのログファイルの場所は~/Library/Logs/Quark Publishing Platform Client.Logです。

ログファイルのサイズを変更するには、LogFileSize属性を使用して新しいサイズを定義します。

LogException属性を使用して、例外のログ記録を有効または無効にできます。

下記の値がサポートされています。

- (NO)例外のログ記録を無効にします。
- (YES)例外のログ記録を有効にします。

ログファイルの作成と管理（Windowsのみ）

ログファイルの作成と管理の設定は、Quark.QPP.Client.configファイル内にあります。このファイルは、アプリケーションがインストールされている場所にあります。QPSLogIsEnabled属性を使用して、ログの記録を有効または無効にできます。

下記の値がサポートされています。

- 0（無効）
- 1（有効）

ログファイルの場所を変更するには、QPSLogFile属性を使用して新しいログファイルの場所を定義します。 デフォルトのログの場所は次のとおりです。

- Quark Publishing Platform Client用は、AppData¥Quark Inc¥Quark Publishing Platform¥{version}¥Logs
- QuarkXPressでのPlatform Client関連の操作用は、AppData¥Roaming¥Quark Software Inc¥QuarkXPress {version}¥{version}¥Logs

ログファイルのサイズを変更するには、QPSLogFileSize属性を使用して新しいサイズを定義します。 サイズは、バイトで定義します。 初期のconfigファイルのデフォルト値は、5242880バイトです。

QPSLogFileMode属性を使用して、ログファイルの書き出し方法を指定します。

下記の値を指定できます。

- 1（新規作成）。 オペレーティングシステムで新しいファイルを作成するように指定されます。 既にファイルがある場合は、System.IO.IOExceptionがスローされます。
- 2（作成）。 オペレーティングシステムで新しいファイルを作成するように指定されます。 既にファイルがある場合は、そのファイルが上書きされます。
- 3（開く）。 オペレーティングシステムで既存のファイルを開くように指定されます。 ファイルがない場合は、System.IO.FileNotFoundExceptionがスローされます。
- 4（開くまたは作成）。 ファイルがある場合はオペレーティングシステムでそのファイルを開き、ファイルがない場合は新規作成するように指定されます。
- 5（破棄）。 オペレーティングシステムで既存のファイルを開くように指定されます。 ファイルを開くと、サイズが0バイトになるようにファイルの内容が破棄されます。
- 6（追加）。 ファイルがある場合は、そのファイルを開いてファイルの終わりの位置に移動します。または、新しいファイルを作成します。

FileMode - 開くまたは作成（4）では、FileAccess.Read、FileAccess.Write、FileAccess.ReadWrite、FileAccess.Appendのアクセス権でファイルを開くことができます。 FileMode - 破棄（5）でファイルを開いてから読み取りを試みると、例外がスローされます。

QPSLogFileShareには、他のプロセスで作成されたログファイルを制限または共有するオプションがあります。 デフォルト値は1（"読み取り"）です。

下記の値を指定できます。

- 0（なし）。 現在のファイルの共有を拒否します。
- 1（読み取り）。 デフォルト値です。 読み取りのためにファイルを開くことを許可します。
- 2（書き出し）。 書き出しのためにファイルを開くことを許可します。
- 3（読み取りと書き出し）。 読み取りまたは書き出しのためにファイルを開くことを許可します。
- 4（削除）。 続けてファイルを削除することを許可します。

- 16（継承可能）。 ファイルハンドルを子プロセスによって継承できるようにします。

バックアップログファイルの最大値を指定するには、QPSLogFileBackupNumber属性を使用します。

ログファイルの作成と管理（Platform用Quark XML Author）

Quark.CMSAdapters.configファイルが下記のように構成されている場合、システムインテグレータやユーザーはログファイルのパスを設定できます。 <!-- Defines log file path.--> <add key="LogFilePath" value="%APPDATA%¥Quark¥XML Author¥Logs¥CMS Adapter Log.txt"/>

ログファイルの作成と管理の設定は、Quark.QPP.Client.configファイル内にあります。このファイルは、アプリケーションフォルダに含まれています。QPSLogIsEnabled属性を使用して、ログの記録を有効または無効にできます。

下記の値がサポートされています。

- 0（無効）
- 1（有効）

ログファイルの場所を変更するには、LogFilePath属性を使用して新しいログファイルの場所を定義します。 ログファイルの場所はAppData¥Quark¥XML Author¥Logs¥CMS Adapter Log.txtです。

ログファイルのサイズを変更するには、QPSLogFileSize属性を使用して新しいサイズを定義します。サイズは、バイトで定義します。初期のconfigファイルのデフォルト値は、5242880バイトです。

QPSLogFileMode属性を使用して、ログファイルの書き出し方法を指定します。

下記の値を指定できます。

- 1（新規作成）。オペレーティングシステムで新しいファイルを作成するように指定されます。既にファイルがある場合は、System.IO.IOExceptionがスローされます。
- 2（作成）。オペレーティングシステムで新しいファイルを作成するように指定されます。既にファイルがある場合は、そのファイルが上書きされます。
- 3（開く）。オペレーティングシステムで既存のファイルを開くように指定されます。ファイルがない場合は、System.IO.FileNotFoundExceptionがスローされます。
- 4（開くまたは作成）。ファイルがある場合はオペレーティングシステムでそのファイルを開き、ファイルがない場合は新規作成するように指定されます。
- 5（破棄）。オペレーティングシステムで既存のファイルを開くように指定されます。ファイルを開くと、サイズが0バイトになるようにファイルの内容が破棄されます。
- 6（追加）。ファイルがある場合は、そのファイルを開いてファイルの終わりの位置に移動します。または、新しいファイルを作成します。

➡ FileMode - 開くまたは作成（4）では、FileAccess.Read、FileAccess.Write、FileAccess.ReadWrite、FileAccess.Appendのアクセス権でファイルを開くことができます。 FileMode - 破棄（5）でファイルを開いてから読み取りを試みると、例外がスローされます。

QPSLogFileShareには、他のプロセスで作成されたログファイルを制限または共有するオプションがあります。デフォルト値は1（"読み取り"）です。

下記の値を指定できます。

- 0（なし）。現在のファイルの共有を拒否します。
- 1（読み取り）。デフォルト値です。読み取りのためにファイルを開くことを許可します。
- 2（書き出し）。書き出しのためにファイルを開くことを許可します。
- 3（読み取りと書き出し）。読み取りまたは書き出しのためにファイルを開くことを許可します。
- 4（削除）。続けてファイルを削除することを許可します。
- 16（継承可能）。ファイルハンドルを子プロセスによって継承できるようにします。

アクセシビリティサービスの警告の非表示

デフォルトでは、Mac OS X用Quark Publishing Platform Clientの起動時に、「アクセシビリティサービスが有効になっていません。」という警告が表示されます。 この警告が表示されないようにするには、次の手順に従ってください。

1. Controlキーを押しながらQuark Publishing Platformt Clientアプリケーションアイコンをクリックし、**パッケージのコンテンツを表示**を選択します。新しいウィンドウが表示されます。
2. "Info.plist"ファイルをテキストエディタで開きます。
3. 次の行を探します。

   <key>QPPDisableAccessibilityWarning</key> <string>0</string>
4. <string>要素の「0」を「1」に変更します。

リビジョンコメントの表示

デフォルトでは、**リビジョンの表示**ダイアログボックスを表示したときにリビジョンコメントを見るには、リビジョンごとに展開する必要があります。 Mac OS Xでリビジョンコメントを自動的に表示するには、下記の手順に従ってください。

1. ~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]に移動します。
2. テキストエディタで、"com.quark.qpp.client.[アプリケーション名].config.plist"ファイルを開きます。

3 次の行を探します。

<key>ExpandAllRevisionComments</key> <false/>

4 要素<false/>を<true/>に変更します。

Windowsの場合、"［アプリケーション名].exe.config"ファイルを開き、次の行を探して、「0」を「1」に変更します。

<add key ="ExpandAllRevisionComments" value="0"/>

姓名の表示

Quark Publishing Platformでは、ユーザー名を以下の3種類のいずれかの方法で表示するように設定できます。

- [ユーザー名]
- [ユーザー名]（[名] [姓]）
- [ユーザー名]（[姓], [名]）

この設定を変更するには、下記の操作を行ってください。

- **Mac OS X用Quark Publishing Platform Client**では、ファイル ~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<string>の値に、[ユーザー名]の場合は0、[ユーザー名]（[名] [姓]）の場合は1、[ユーザー名]（[姓], [名]）の場合は2を設定します。
<key>UserNameFormattingStyle</key> <string>0</string>

- **Windows用Quark Publishing Platform Client**では、Quark Publishing Platform Clientアプリケーションフォルダ内の"Quark Publishing Platform Client.exe.config"ファイルを開き、下記のテキストを探して、value属性に、[ユーザー名]の場合はDEFAULT、[ユーザー名]（[名] [姓]）の場合はFIRSTNAME_LASTNAME、[ユーザー名]（[姓], [名]）の場合はLASTNAME_FIRSTNAMEを設定します。
<add key="UserNameFormattingStyle" value="DEFAULT"/>

- **Mac OS X用QuarkXPress**では、ファイル ~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<string>の値に、[ユーザー名]の場合は0、[ユーザー名]（[名] [姓]）の場合は1、[ユーザー名]（[姓], [名]）の場合は2を設定します。
<key>UserNameFormattingStyle</key> <string>0</string>

- **Mac OS X用QuarkCopyDesk**では、ファイル ~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<string>の値に、[ユーザー名]の場合は0、[ユーザー名]（[名] [姓]）の場合は1、[ユーザー名]（[姓], [名]）の場合は2を設定します。
<key>UserNameFormattingStyle</key> <string>0</string>

- **Windows用QuarkXPressおよびQuarkCopyDesk**では、アプリケーションフォルダ内の"Quark.QPP.client.config"ファイルを開き、下記のテキストを探して、value属性に、[ユーザー名]の場合はDEFAULT、[ユーザー名]（[名] [姓]）の場合はFIRSTNAME_LASTNAME、[ユーザー名]（[姓], [名]）の場合はLASTNAME_FIRSTNAMEを設定します。
  <add key="UserNameFormattingStyle" value="DEFAULT"/>

プレビューのフォントとサイズの変更（Windowsのみ）

属性値に使用されるフォントとサイズを、デフォルトの検索結果パレットのフォントとサイズ以外のフォントに変更するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>セクションに下記のキーを追加して、"value"属性に必要なフォントとサイズを指定します。

<add key="FontName_Text Preview" value="Arial, 18"/>

フェッチ可能なアセットの最大数の設定（Windowsのみ）

ユーザーがGet Assetコマンドや Get Collectionコマンドを使用したとき、コレクションごとにフェッチできるアセットの最大数を変更するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>セクションに下記のキーを追加し、"value"属性に数を設定します。

<add key="MaximumAssetFetchPerCollection" value="50"/>

➡ デフォルト値は50です。

チャンクエンコーディングを使用するかどうかの指定（Windowsのみ）

ファイルをHTTPで転送するときに**チャンクエンコーディング**を使用するかどうかを指定するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>セクションに下記のキーを追加します。 サポートされる値は0と1です。

<add key="UseChunkedEncoding" value="0"/>

➡ デフォルト値は0です。

検索時の遅延読み込みサポートの指定（Windowsのみ）

検索時の遅延読み込み方式のサポートを指定するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>セクションに下記のキーを追加し、"value"属性に方式を指定します。

<add key="LazyLoadingMode" value="LAZY_LOADING_SCROLLBAR"/>

➡ 下記の値がサポートされています。

- NO_LAZYLOADING： すべてのアセットは、チャンクサイズによって、ただちに設定されます。

- LAZY_LOADING_HYPERLINK： 指定された結果はチャンクサイズに基づいて取得され、チャンクサイズによって指定された結果が取得されたとき、結果の次のチャンクをフェッチするためのハイパーリンクが表示されます。
- LAZY_LOADING_SCROLLBAR： 指定された結果はチャンクサイズに基づいて取得され、スクロールバーを使用して、結果の次のチャンクをフェッチできます。

➡ デフォルト値はLAZY_LOADING_SCROLLBARです。

遅延読み込みのチャンクサイズの設定（Windowsのみ）

遅延読み込みの検索結果をフェッチするチャンクサイズを変更するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>に下記のキーを追加し、"value"属性にサイズを設定します。

```
<add key="LazyLoadingChunkSize" value="50"/>
```

➡ デフォルト値は50です。

すべてのリモートサービス参照についてサービスのタイムアウト値を設定する（Windowsのみ）

すべてのリモートサービス参照について、サービスのタイムアウト値を変更するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>セクションに下記のキーを追加し、"value"属性に時間を秒単位で設定します。

```
<add key="ServiceTimeoutInSeconds" value="0"/>
```

➡ デフォルト値は0です。 Platform Serverの設定に指定されているデフォルト値が使用されます。

パブリッシングサービスのサービスタイムアウト値の設定（Windowsのみ）

パブリッシングサービスについて、サービスのタイムアウト値を変更するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>セクションに下記のキーを追加し、"value"属性に時間を秒単位で設定します。

```
<add key="PublishingServiceTimeoutInSeconds" value="600"/>
```

➡ デフォルト値は600です。 値が0の場合、ServiceTimeoutInSecondsで指定された値が有効になることを示します。

コピーテイスティング行のフォントサイズの指定（Windowsのみ）

コピーテイスティング行に使用するフォントサイズを変更するには、"Quark Publishing Platform Client.exe.config"ファイルの<appSettings>セクションに下記のキーを追加し、"value"属性にフォントサイズを設定します。

```
<add key="CopyTastingRowFontSize" value="25"/>
```

ファイル拡張子のアイコンの指定（Windowsのみ）

指定したファイル拡張子とアイコンとを関連付けるには、"Quark Publishing Platform Client.exe.config"の<appSettings>セクションに下記のキーを追加し、"value"属性に目的のアイコンへのパスを指定します。

<add key="icon_<file extension>" value="Path of icon"/>

➡ キーには"file extension"を指定し、値にはファイルパスを指定します。

パスワード保存機能の制御（Mac OSのみ）

前回のログインで使用したユーザー名をQuark Publishing Platformクライアントに記憶させ、ユーザーパスワードは記憶させないように設定できます。このオプションを設定するには、下記の操作を行ってください。

- **Quark Publishing Platform Client**では、ファイル~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<false/>を<true/>に変更します。
  <key>RememberPassword</key> <false/>

- **QuarkXPress for Mac OS X**では、ファイル~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<false/>を<true/>に変更します。
  <key>RememberPassword</key> <false/>

- **QuarkCopyDesk for Mac OS X**では、ファイル~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<false/>を<true/>に変更します。
  <key>RememberPassword</key> <false/>

プロキシサーバー経由のMacクライアントの使用

ファイアウォールの外側にあるMac OS Xクライアントから、プロキシサーバー経由でQuark Publishing Platform Serverに接続できるようにするには、下記の操作を行ってください。

- **Quark Publishing Platform Client**では、ファイル~/Library/Application Support/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<false/>を<true/>に変更します。
  <key>UseProxy</key> <false/>

- **QuarkXPress for Mac OS X**では、ファイル~/Library/Application Support/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<false/>を<true/>に変更します。
  <key>UseProxy</key> <false/>

- **QuarkCopyDesk for Mac OS X**では、ファイル~/Library/Application Support/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.{Application}.config.plistを開き、下記のテキストを探して、<false/>を<true/>に変更します。
  <key>UseProxy</key> <false/>

プロキシサーバー経由のWindowsクライアントの使用

プロキシサーバー経由でWindowsクライアントを使用するには、下記の手順に従ってください。

1. "[アプリケーション名].exe.config"ファイルを開き、次の行を探します。
   <!-- <add key="ProxyAddress" value="http://<プロキシ名>:<ポート番号>"/> -->
2. 行のコメント化を解除して、適切なプロキシの詳細を挿入します。
3. ファイルを保存して閉じます。

チェックアウト／取得におけるコレクション階層のミラーリング

デフォルトでは、Quark Publishing Platformクライアントは、アセットのチェックアウトまたは取得時にローカルドライブのコレクション階層をミラーリングします。ただし、このオプションは変更できます。

コレクションミラーリングの停止：Mac OS X

Mac OS Xでコレクションミラーリングを停止するには、下記の手順に従ってください。

1. Quark Publishing Platform Clientでは、ファイル~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.Quark Publishing Platform Client.config.plistを開き、下記のテキストを探して、<true/>を<false/>に変更します。
   <key>MirrorCollectionHierarchy</key> <true/>
2. QuarkXPressでは、ファイル~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.QuarkXPress.config.plistを開き、下記のテキストを探して、<true/>を<false/>に変更します。
   <key>MirrorCollectionHierarchy</key> <true/>
3. QuarkCopyDeskでは、ファイル~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.client.QuarkCopyDesk.plistを開き、下記のテキストを探して、<true/>を<false/>に変更します。
   <key>MirrorCollectionHierarchy</key> <true/>

コレクションミラーリングの停止：Windows

Windowsでコレクションミラーリングを停止するには、下記の手順に従ってください。

1 Quark Publishing Platform Clientの場合、"Quark Publishing Platform Client.exe.config"を開き、次のkeyを探して、valueを0に設定します。

```
<add key="MirrorCollectionHierarchy" value="1"/>
```

2 QuarkXPressの場合、"Quark.QPP.Client.config"を開き、次のkeyを探して、valueを0に設定します。

```
<add key="MirrorCollectionHierarchy" value="1"/>
```

3 QuarkCopyDeskの場合、"Quark.QPP.Client.config"を開き、次のkeyを探して、valueを0に設定します。

```
<add key="MirrorCollectionHierarchy" value="1"/>
```

配信チャネルの設定

Quark Publishing Platformでサポートされる配信チャネルを制御できます。

配信チャネルの設定：Mac OS X

MAC OS Xで配信チャンネルを設定するには、下記の手順に従ってください。

1 Mac OSクライアントの場合、次のファイルを開きます。

```
~/Library/Preferences/Quark/QPP/[QPP
Framework  Version]/com.quark.qpp.publishing.preferences.v7.plist
```

2 クライアントが優先する配信チャネルのリストを指定する下記のkeyを探し、チャネルを追加または削除します。

```
<key>EnabledPublishingChannels</key>
<array>
  <string>qxpPdf</string>
  <string>qxpEpub</string>
  <string>qxpAppStudio</string>
  <string>qxpAppStudioPackage</string>
  <string>busDocPdf</string>
  <string>busDocHtml</string>
  <string>busDocQxp</string>
  <string>busDocAppStudio</string>
  <string>busDocAppStudioPackage</string>
  <string>ditaDocPdf</string>
  <string>ditaDocWordRTF</string>
  <string>ditaDocHtml</string>
  <string>collectBusdocForOutput</string>
  <string>collectDitaForOutput</string>
  <string>smartDocHtml5Publication</string>
  <string>busDocHtml5Publication</string>
  <string>visioDocPdf</string>
  <string>fetchVisioPage</string>
  <string>fetchPowerPointSlide</string>
</array>
```

3 配信チャネルを追加または削除します。

配信チャネルの設定：Windows

Windowsで配信チャネルを設定するには、下記の手順に従ってください。

1 Quark Publishing Platform Clientの場合、"Quark Publishing Platform Client.exe.config"を開き、クライアントが優先する配信チャネルのリストを指定する下記のkeyを探します。

```
<add key="EnabledPublishingChannels"
value="qxpPdf,qxpEpub,qxpAppStudio,qxpAppStudioPackage,
```

busDocPdf, busDocHtml,busDocQxp,busDocAppStudio,
busDocAppStudioPackage, ditaDocPdf,ditaDocWordRTF,ditaDocHtml,
qxpAppStudio, collectBusdocForOutput, collectDitaForOutput, smartDocHtml5Publication,
busDocHtml5Publication, visioDocPdf, fetchVisioPage, fetchPowerPointSlide"/>

2 配信チャネルを追加または削除します。

3 QuarkXPressおよびQuarkCopyDeskの場合、"Quark.QPP.Client.config"ファイルを開き、同様の変更を行います。

配信チャネルの設定

Quark Publishing Platformでサポートされる配信チャネルを制御できます。

配信チャネルの設定：Mac OS X

Mac OS Xで配信チャネルを設定するには、下記の手順に従ってください。

1 Mac OS Xクライアントの場合、次のファイルを開きます。

~/Library/Preferences/Quark/QPP/[QPPフレームワークのバージョン]/com.quark.qpp.publishing.preferences.v5.plist

2 次の行を探します。

<key>EnableDeliveryChannels</key> <false/>

3 要素<false/>を<true/>に変更します。

4 クライアントが優先する配信チャネルのリストを指定する下記のkeyを探します。

<key>EnabledDeliveryChannels</key> <array> <string>checkInToSharepoint</string>
<string>checkInToFileNet</string> <string>sendEmail</string>
<string>checkInToDocumentum</string> <string>sendToFileSystem</string>
<string>sendToFTPServer</string> </array>

5 配信チャネルを追加または削除します。

配信チャネルの設定：Windows

Windowsで配信チャネルを設定するには、下記の手順に従ってください。

1 Windowsクライアントの場合、次のファイルを開きます。

~application name.exe.config

2 次の行を探します。

<add key ="EnableDeliveryChannels" value="0">

3 0を1に変更します。

4 クライアントが優先する配信チャネルのリストを指定する下記の行を探します。

<add key="EnabledDeliveryChannels"
value="checkInToSharepoint,checkInToFileNet,sendEmail,checkInToDocumentum,sendToFTPServer,sendToFileSystem"/>

5 配信チャネルを追加または削除します。

Quark XML Author for Platformの環境の設定

チェックアウト場所の設定

チェックアウト場所の環境設定を指定するには、"Quark.CMS.Adapters.config"ファイルの<appSettings>セクションの下にあるCheckOutLocationキーを使用します。このキーは、アプリケーションを最初に起動したときの最初の環境設定を指定します。

チェックアウト場所を変更するには、**ファイル > 環境設定 > 一般**へ移動し、チェックアウト場所として設定する場所へ移動します。

この環境設定は、下記の方法でリセットできます。

- デフォルトにリセット：ユーザーは、"Quark.CMS.Adapters.config"ファイルのappSettingsセクションの下にあるCheckOutLocationキーを変更し、詳細設定 > デフォルトにリセット機能を使用して、環境設定を変更できます。

保存して閉じるのファイル削除の環境設定

ファイル削除の環境設定を指定するには、"Quark.CMS.Adapters.config"ファイルの<appSettings>セクションの下にあるFileDeletionOptionキーを使用します。 このキーは、アプリケーションを最初に起動したときの最初の環境設定を指定します。

ファイル削除の環境設定を変更するには、**ファイル > 環境設定 > 一般**へ移動し、次のいずれかの値を選択します。

1. 警告を表示しないで削除する
2. 削除を許可しない
3. 削除する前に常に尋ねる

この環境設定は、下記の2つの方法でリセットできます。

- デフォルトにリセット： ユーザーは、"Quark.CMS.Adapters.config"ファイルのappSettingsセクションの下にあるFileDeletionOptionキーを変更し、詳細設定 > デフォルトにリセット機能を使用して、環境設定を変更できます。
- 設定の変更： 管理者は、ファイルを更新してから、更新された"Quark.CMS.Adapters.config"ファイルを任意の場所で共有し、詳細設定 > 設定の変更機能を使用して新しい環境設定を読み込むようユーザーに指示できます。

クイック検索の環境設定

クイック検索の環境設定を指定するには、"Quark.CMS.Adapters.config"ファイルの<appSettings>セクションの下にあるQuickSearchOptionキーを使用します。このキーは、アプリケーションを最初に起動したときの最初の環境設定を指定します。

ファイル削除の環境設定を変更するには、**ファイル > 環境設定 > 検索**へ移動し、次のいずれかの値を選択します。

1. 名前
2. 内容
3. 名前とコンテンツ

この環境設定は、下記の方法でリセットできます。

- デフォルトにリセット：ユーザーは、"Quark.CMS.Adapters.config"ファイルのappSettingsセクションの下にあるQuickSearchOptionキーを変更し、詳細設定 > デフォルトにリセット機能を使用して、環境設定を変更できます。

保存して閉じるのリビジョンコメントの表示方法の環境設定

リビジョンコメントの表示方法の環境設定を指定するには、"Quark.CMS.Adapters.config"ファイルの<appSettings>セクションの下にあるRevisionCommentsDisplayOptionキーを使用します。 このキーは、アプリケーションを最初に起動したときの最初の環境設定を指定します。

ファイル削除の環境設定を変更するには、**ファイル > 環境設定 > 検索**へ移動し、次のいずれかの値を選択します。

1 常に
2 常に表示しない
3 新規割り当てのときのみ表示する

この環境設定は、次の2つの方法でリセットできます。

- デフォルトにリセット： ユーザーは、"Quark.CMS.Adapters.config"ファイルのappSettingsセクションの下にあるRevisionCommentsDisplayOptionキーを変更し、詳細設定 > デフォルトにリセット機能を使用して、環境設定を変更できます。
- 設定の変更： 管理者は、ファイルを更新してから、更新された"Quark.CMS.Adapters.config"ファイルを任意の場所で共有し、詳細設定 > 設定の変更機能を使用して新しい環境設定を読み込むようユーザーに指示できます。

Microsoft Officeコンポーネント用のPlatformアダプタでWeb共有を設定する方法

Web共有の設定は、スタンドアロンのサーバーインストールにのみ使用し、外部Webコンテナの展開や、EARベースの展開には使用しないことをお勧めします。これらのシナリオでWeb共有設定が必要な場合、Quarkテクニカルサポートと協力して作業を行う必要があります。

この章では、次のMicrosoft Officeコンポーネントをインストールする手順について説明します。

- Microsoft Office用のQuark Publishing Platform Adapter - Word
- Microsoft Office用のQuark Publishing Platform Adapter - Excel
- Microsoft Office用のQuark Publishing Platform Adapter - PowerPoint

展開場所は、Platformのサーバーマシンの"Quark Publishing Platform Server" webappsフォルダです。

コンポーネントをインストールする前に、下記の手順を完了する必要があります。

1 アーカイブ"Platform<*version* and *build*>_Adapter_for_Office.zip"をコピーして抽出します。

2 抽出されたアーカイブから、"webapps¥ROOT"フォルダのすべてのファイルを"Quark Publishing Platform Server¥webapps¥ROOT"フォルダへ上書きコピーします。
3 "webapps¥clientinstallers"フォルダの内容を、"Quark Publishing Platform Server¥webapps¥clientinstallers"フォルダへコピーします。

その後で、以下のセクションの手順に従い、各コンポーネントをインストールします。

Microsoft Office用のQuark Publishing Platform Adapterの設定 - Word

Microsoft Word用のPlatform Adapterをインストールするには、下記の手順に従ってください。

1 Platformサーバーマシンのwebapps¥clientinstallersフォルダから、アーカイブの"Word Adapter.zip"ファイルの内容を、webapps¥clientinstallers¥Word Adapterフォルダへ抽出します。
2 clientinstallers/Word Adapterフォルダにある"ConfigureAdapter.bat"ファイルを開きます。
3 HostName/IPとPortを更新してから、バッチファイルを実行します。

Microsoft Office用のQuark Publishing Platform Adapterの設定 - Excel

Microsoft Excel用のPlatform Adapterをインストールするには、下記の手順に従ってください。

1 Platformサーバーマシンのwebapps¥clientinstallersフォルダから、アーカイブの"Excel Adapter.zip"ファイルの内容を、webapps¥clientinstallers¥Excel Adapterフォルダへ抽出します。
2 clientinstallers/Excel Adapterフォルダにある"ConfigureAdapter.bat"ファイルを開きます。
3 HostName/IPとPortを更新してから、バッチファイルを実行します。

Microsoft Office用のQuark Publishing Platform Adapterの設定 - PowerPoint

Microsoft Excel用のPlatform Adapterをインストールするには、下記の手順に従ってください。

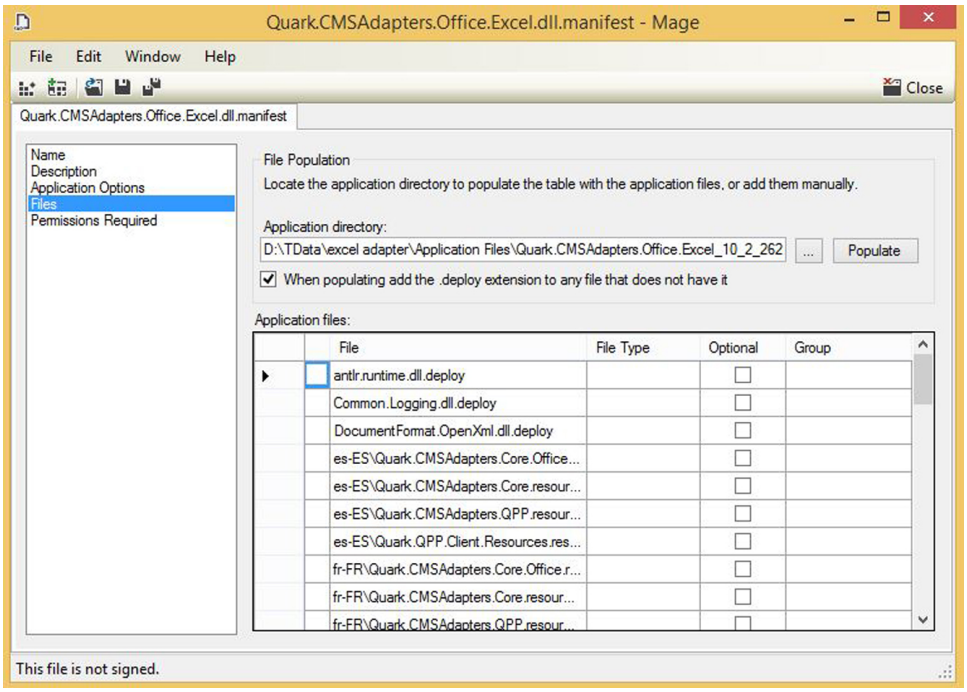
1 Platformサーバーマシンの"webapps¥clientinstallers"フォルダから、アーカイブの"PowerPoint Adapter.zip"ファイルの内容を、"webapps¥clientinstallers¥PowerPoint Adapter¥"フォルダへ抽出します。
2 "clientinstallers/PowerPoint Adapter"の下にあるバッチファイル"ConfigureAdapter.bat"のHostName/IPおよびPortを更新してから、バッチファイルを実行します。

発行済みのClickOnce展開の更新

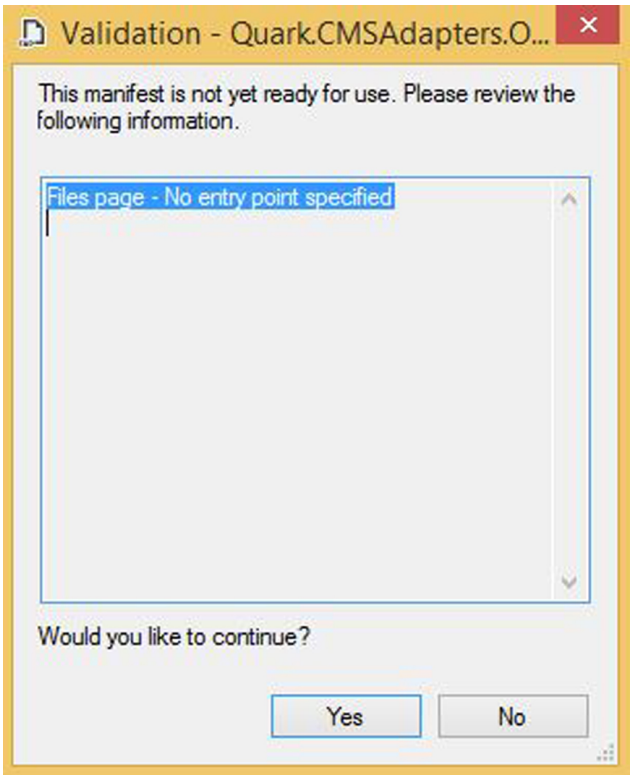
発行済みのClickOnce展開パッケージを更新するには、Microsoft Visual Studioとともにインストールされるマニフェスト生成および編集ツール（mage.exe）を使用します。

パッケージとアプリケーションマニフェストの更新

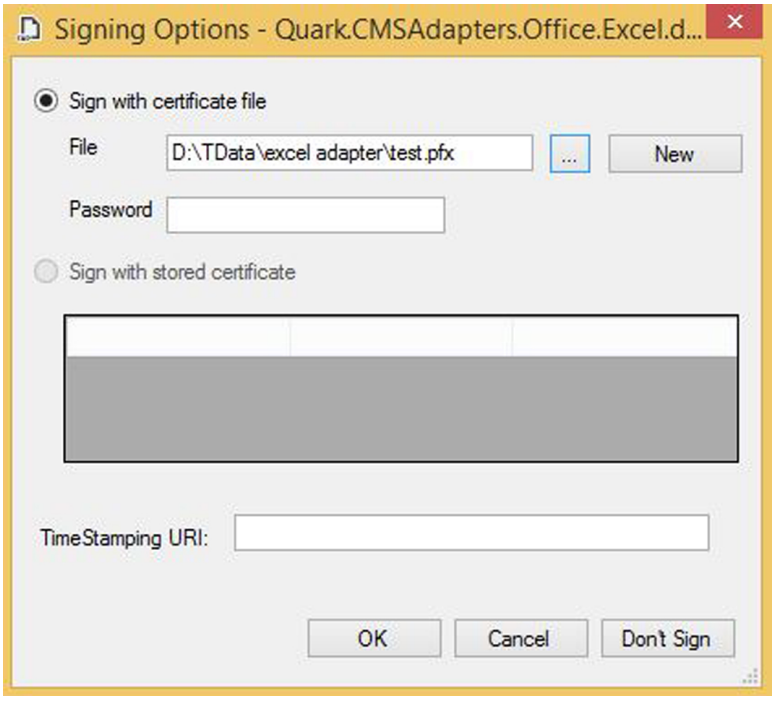
1. Application Files¥Quark.CMSAdapters.Office.<component>_%HighestVersion%へ移動し、"Quark.CMSAdapters.Office.<component>.vsto"ファイルを削除します。ここで、コンポーネントはWord、Excel、PowePointのいずれかです。
2. 必要に応じて、設定ファイルに変更を加えます。
3. Mageツールを実行します。 Visual Studio ToolsからVisual Studioコマンドプロンプトを使用し、mage.exeと入力します。
4. Mageで、¥Application Files¥App_%HighestVersion%¥フォルダにあるアプリケーションマニフェスト"Quark.CMSAdapters.Office.<component>.dll.manifest"を開きます。



5. **ファイル**を選択し、**書き込みのとき、.deploy拡張子がないファイルへ自動的に拡張子を付加する**オプションをチェックします。 次に**書き込み**ボタンをクリックし、**保存**または**ファイル > 保存**をクリックしてファイルを保存します。



6 **検証**ダイアログで**はい**をクリックします。



7 **証明書ファイルで署名する**オプションを選択し、証明書ファイルのある場所のパスとパスワード（使用している場合）を入力します。**OK**をクリックします。

VSTOの更新

1. パッケージフォルダに存在する"Quark.CMSAdapters.Office.<component>.vsto"を更新するには、下記のコマンドを使用してください。

   mage -Update "**<path>**¥Quark.CMSAdapters.Office.<component>.vsto" -AppManifest "**<path>**¥Quark.CMSAdapters.Office.<component>.dll.manifest" -CertFile "**<certificate Path>**¥<certificate>.pfx" -Password "<password>"
2. "Quark.CMSAdapters.Office.Word.vsto"ファイルを、パッケージフォルダから¥Application Files¥App_%HighestVersion%¥へコピーします。

QuarkXPressおよびQuarkCopyDesk XTensionsの手作業での設定

QuarkXPressおよびQuarkCopyDesk XTensionsにはxmlベースの環境設定があり、アプリケーション環境設定フォルダのQPPXPressXT.xmlおよびQPPCopyDeskXT.xmlファイルに格納されています。

➡ Windowsでは、QuarkXPress環境設定フォルダはC:¥Users¥<ユーザー名>¥AppData¥Local¥Quark¥QuarkXPress 2016に、QuarkCopyDesk環境設定フォルダはC:¥Users¥<ユーザー名>¥AppData¥Local¥Quark¥QuarkCopyDesk 2016にあります。

➡ Mac OS Xでは、QuarkXPress環境設定フォルダは/Volume/users/username/Library/Preferences/Quark/QuarkXPress 2016にあります。QuarkCopyDesk環境設定フォルダは/Volume/users/username/Library/Preferences/Quark/QuarkCopydesk 2016にあります。

ほとんどの環境設定はユーザーインターフェイスを使用して設定されますが、Platformメニューアイテムの場所は手作業で設定します。

Platformメニューアイテムの場所を設定するには、下記の操作を行ってください。

- "QPPXPressXT.xml"および"QPPCopyDeskXT.xml"ファイルの<Basic>セクションの下で、PlatformMenuItemsLocationノードのlocation属性を設定します。次のいずれかの値に設定します。

1. 0：ファイルメニュー（デフォルト値）
2. 1：プラットフォームメニュー
3. 2：ファイルメニューとPlatformメニューの両方が表示されます。

バックアップとファイルストレージの管理

Quark Publishing Platformデータベース、Quark Publishing Platformアセット、およびQuark® Job Jackets®（ジョブジャケット）ファイルやスクリプトファイルなどの必須ファイルのバックアップソフトウェアを選択し、バックアップ間隔を決定します。バックアップを復元しなければならない場合の不一致を避けるため、データベース、アセット、および必須ファイルのバックアップを同期することをお勧めします。また、必要に応じてバックアップが正常に復元できることを確認するために定期的にテストすることもお勧めします。

アセットリポジトリを移動する場合には、「*Quark Publishing Platform Rendererアセットリポジトリの移行*」の手順に従ってください。

Quark Publishing Platform Serverのバックアップ

バックアップを実行する前にQuark Publishing Platform Serverを停止することをお勧めしますが、Quark Publishing Platform Serverの停止は必須ではありません。データベース、アセット、およびFTS索引ファイルなどの必須ファイルを別の保存装置にバックアップします。"Quark Publishing Platform Server"フォルダ全体をバックアップすることもできますが、次のフォルダが特に重要です。

- "conf"フォルダ（インストール後に手動で編集されたものを含め、システム設定ファイルが保存されています）
- "index"フォルダ（全文検索索引ファイルが保存されています）

データベースのバックアップ

データベースには、Quark Publishing Platformアセットのすべてのメタデータが入っています。

Microsoft® SQLデータベースまたはOracleデータベースを使用する場合は、MS-SQLまたはOracleで提供されるバックアップツールと手順を使用します。

Quark Publishing Platform Serverで埋め込みHSQLデータベースを使用する場合には、データベース情報は"Quark Publishing Platform Server"フォルダ内の"database"フォルダに専用に保存されます。メタデータを保護し、ワークフロー設定を保存するために"database"フォルダをバックアップする必要があります。"database"フォルダが復帰できない場合には、復帰させるためにQuark Publishing Platformアセットを手動で識別する必要があります。

アセットのバックアップ

Quark Publishing Platformアセットリポジトリをバックアップするソフトウェアと間隔を指定します。

➡ Quark Publishing Platformアセットリポジトリ内では、アセット名は暗号化されています。

索引ファイルのバックアップ（全文検索）

Quark Publishing Platform Serverでは、データベースにチェックインしたすべてのファイルに索引が付けられるため、ユーザーはQuark Publishing Platformアセットのテキストコンテンツ内で検索を実行できます。全文検索操作の索引情報は"index"フォルダに保存されます。"index"フォルダはQuark Publishing Platform Serverフォルダのルートレベルにあり、全文索引に必要なファイルのデフォルトの場所です。*全文索引の設定*"を編集して全文索引のストレージ場所を変更する方法については、「全文索引の設定」を参照してください。場所を変更する場合は、新しい場所をバックアップします。

Quark Publishing Platform Serverの復元

Quark Publishing Platformアセットリポジトリを復元する必要がない場合には、アセットリポジトリへのストレージパスはQuark Publishing Platformデータベースを復元した後にも有効です。データベースだけでなくQuark Publishing Platformアセットリポジトリを復元する必要がある場合には、「*Quark Publishing Platform Rendererアセットリポジトリの移行*」の手順に従って、更新されたストレージ場所を指定します。

たとえば、Quark Publishing Platform Serverを実行しているコンピュータ上のハードドライブが故障した場合、最新のバックアップは別の場所に保存されている必要があります。『**Quark Publishing Platform ReadMe**』の手順に従ってQuark Publishing Platform Serverを再インストールします。Quark Publishing Platform Serverを再インストールした後、データベース、アセット、その他のファイルを復元する前にQuark Publishing Platform Serverを実行しないでください。

アセットの修復

なるべく、古いアセットリポジトリに使用したパスと同じものを使用してください。たとえば、Quark Publishing Platform Serverを実行しているコンピュータ上のハードドライブを交換する場合、アセットリポジトリのバックアップを同じ場所にコピーできます。新しいコンピュータが必要な場合でも、なるべく同じパス（たとえばC:¥QPP_Assetsなど）を使用してください。

Quark Publishing Platformアセットを検索することができ、暗号化されたファイル名を参照する必要がある場合には、検索が完了したときに「ファイルパス」属性を参照してください。マスターファイルとすべてのアセットのリビジョンは同じストレージ場所に保存されます。暗号化されたアセット名は同じパターンに従います。たとえば、34.1.1.1.JPGとなります。暗号化された名前の最初の数字はアセットIDを示します。2番目の数字はバージョン番号、3番目の数字はプレビューのレンダリングタイプ、4番目の数字はプレビューの最初のページを示します。

➡ ハードウェアを復元する必要がある場合には、Quark Publishing Platform Clientを起動して**管理：ストレージ画面**の**リポジトリタブで定義される1つまたは複数のリポジトリ**

へのリンクを回復します。ストレージリポジトリを削除して別のものを作成することはしないでください。

Quark Publishing Platform Serverデータベースの復元

Microsoft SQLデータベースまたはOracleデータベースを使用する場合は、MS-SQLまたはOracleで提供される復元ツールと手順を使用します。Quark Publishing Platform Serverを再インストールする場合、インストール処理中にMS-SQLまたはOracleの正しい情報を入力できます。

全文索引の復元

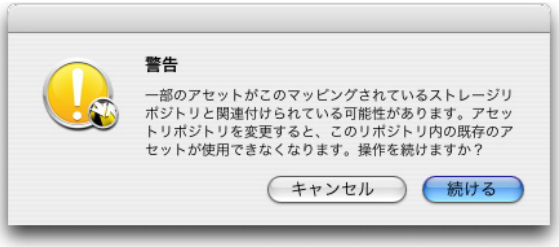
"LuceneTextIndexingConfig.properties"ファイルで指定された場所にある"index"フォルダを復元します。

Quark Publishing Platform Rendererアセットリポジトリの移行

Quark Publishing Platformアセットリポジトリを移動する場合、Quark Publishing Platform Clientでアセットリポジトリパスを更新できます。アセットストレージの指定についての詳細は、『**A Guide to Quark Publishing Platform**』の「ストレージオプションの設定」を参照してください。

アセットリポジトリパスを更新するには、下記の手順に従ってください。

1. Webブラウザでhttp://[サーバー]:[ポート]/adminに移動し、管理者権限でログオンします。
2. **管理**画面を表示し、**ストレージ**をクリックします。**管理：複製**画面が表示されます。
3. **リポジトリ**タブを表示します。**リポジトリ名**列に1つ以上のエントリーが表示されます。
4. リポジトリを選択し、**編集 > リポジトリの編集**を選択します。警告メッセージが表示されます。



この警告はアセットリポジトリを編集すると表示されます

5. **はい**をクリックします。**リポジトリの編集**ダイアログボックスが表示されます。

リポジトリの編集ダイアログボックス

6 **場所**フィールドに新しい場所を入力します。

7 **リポジトリ**タブのリストにあるすべてのリポジトリに対して手順4～6を繰り返します。

法律上の注記

索引