



# Deploying QlikView

---

QlikView®

x.y

Copyright © 1993-2016 QlikTech International AB. All rights reserved.



Copyright © 1993-2016 QlikTech International AB. All rights reserved.

Qlik®, QlikTech®, Qlik Sense®, QlikView®, Sense® and the Qlik logo are trademarks which have been registered in multiple countries or otherwise used as trademarks by QlikTech International AB. Other trademarks referenced herein are the trademarks of their respective owners.

---

|  |          |
|--|----------|
| <b>1 Introduction</b>                              | <b>8</b> |
| <b>2 Planning QlikView Deployments</b>             | <b>9</b> |
| 2.1 Architecture                                   | 9        |
| Front End  | 9        |
| Back End   | 10       |
| QlikView Server                                    | 10       |
| QlikView Server – Client Communication             | 10       |
| QlikView Server access to User Document            | 12       |
| Web Server   | 12       |
| QlikView Server Tunnel                             | 13       |
| Directory Service Connector                        | 13       |
| Management Service                                 | 13       |
| Distribution Service                               | 13       |
| QlikView architecture without Distribution Service | 14       |
| Reload Engine                                      | 14       |
| Documents, Data, and Tasks                         | 14       |
| User Documents                                     | 14       |
| Shared Files                                       | 15       |
| Source Data  | 15       |
| Source Documents                                   | 16       |
| Tasks  | 16       |
| Transforming Source Document into User Document    | 16       |
| Ports  | 17       |
| Service by Service                                 | 19       |
| QlikView Server Load Sharing (Clustering)          | 19       |
| Overview   | 19       |
| Files  | 20       |
| Load Sharing (Clustering)                          | 21       |
| QlikView Distribution Service                      | 23       |
| Overview   | 23       |
| Files  | 23       |
| Changing the storage time of log files             | 25       |
| QlikView Batch                                     | 26       |
| Settings and Configuration                         | 26       |
| Logs   | 26       |
| QlikView Publisher Repository                      | 26       |
| Overview   | 26       |
| Files  | 26       |
| Security Groups                                    | 27       |
| Configuration Files                                | 27       |
| QlikView Web Server                                | 29       |
| Overview   | 29       |
| Files  | 30       |
| Directory Service Connector                        | 40       |

---

## Contents

---

|  |    |
|--|----|
| Overview .....                                       | 40 |
| Files .....  | 41 |
| DSP Interface .....                                  | 41 |
| QlikView Management Service .....                    | 42 |
| Overview .....                                       | 42 |
| Files .....  | 43 |
| SNMP .....   | 43 |
| MIB File .....                                       | 45 |
| 2.2 Deployment .....                                 | 46 |
| Building a Farm .....                                | 46 |
| Planning .....                                       | 46 |
| Trust Mechanism .....                                | 46 |
| Web Server .....                                     | 47 |
| Redundancy Level .....                               | 47 |
| Account to Run the Services Under .....              | 47 |
| QVPR Format .....                                    | 47 |
| User Directory .....                                 | 47 |
| User Authentication .....                            | 47 |
| Firewalls .....                                      | 47 |
| Root/First Install .....                             | 47 |
| Adding Services on Other Machines .....              | 48 |
| Clustering .....                                     | 48 |
| QlikView Server .....                                | 48 |
| QlikView Distribution Service .....                  | 48 |
| Directory Service Connector .....                    | 48 |
| QlikView Web Server .....                            | 48 |
| Clustering QlikView Servers .....                    | 49 |
| Why Cluster QlikView Servers? .....                  | 50 |
| Horizontal User Scalability .....                    | 50 |
| Resilience .....                                     | 51 |
| Load balancing .....                                 | 51 |
| Requirements for Clustered QlikView Deployment ..... | 51 |
| Clustered QlikView Server License Key .....          | 51 |
| Shared Network Storage .....                         | 52 |
| QVS Load Balancing Options .....                     | 52 |
| Load Balancing the Web Server .....                  | 53 |
| Building and Installing a QlikView Cluster .....     | 54 |
| Unbalanced QVS Clustering .....                      | 58 |
| Clustering QlikView Publisher .....                  | 59 |
| Introduction .....                                   | 59 |
| Source Documents .....                               | 60 |
| User Documents .....                                 | 60 |
| Tasks .....  | 60 |
| Why Cluster QlikView Publisher? .....                | 61 |

---

## Contents

---

|   |    |
|---|----|
| Horizontal Scalability .....  | 61 |
| Resilience .....  | 61 |
| Requirements for a Clustered QlikView Publisher Deployment .....              | 62 |
| Clustered QlikView Publisher License Key .....                                | 62 |
| Shared Network Storage .....  | 62 |
| Load Balancing Strategies .....   | 62 |
| Security .....  | 64 |
| Directory Services .....  | 65 |
| QlikView Server Authorization Modes .....                                     | 65 |
| Static Data Reduction .....   | 66 |
| Configuring QlikView Publisher Clustering .....                               | 66 |
| Requirements .....  | 66 |
| Step-by-step Instructions .....   | 66 |
| Clustering QlikView Distribution Service .....                                | 70 |
| What is a QDS Publisher Group? .....  | 70 |
| QDS publisher group configuration .....                                       | 71 |
| Task Configuration .....  | 72 |
| QlikView Server Extensions .....  | 72 |
| Adding Extensions to QlikView Server .....                                    | 72 |
| Configuring IIS for Custom Users .....  | 73 |
| QlikView Triggering EDX Enabled Tasks .....                                   | 75 |
| Cleaning and converting the shared files .....                                | 77 |
| Verify mode .....   | 77 |
| Purge mode .....  | 78 |
| Converting the shared files .....   | 78 |
| Setting and changing ownership of shared file content .....                   | 78 |
| Cleaning tool command format .....  | 78 |
| Using the shared file cleaning tool .....                                     | 80 |
| Examples .....  | 81 |
| 2.3 Security Overview .....   | 83 |
| Protection of the Platform .....  | 83 |
| Functionality .....   | 83 |
| Special Accounts .....  | 83 |
| Supervision Account .....   | 83 |
| Anonymous User Account .....  | 83 |
| QlikView Administrators .....   | 84 |
| Communication .....   | 84 |
| Protection of AJAX Client .....   | 84 |
| Protection of Plugin .....  | 84 |
| Server Communication .....  | 84 |
| Services Communication .....  | 84 |
| SSL and TLS support .....   | 84 |
| Authentication .....  | 85 |
| Authentication when Using QlikView Server in a Windows User Environment ..... | 85 |

---

|   |            |
|---|------------|
| Authentication with a QlikView Server Using an Existing Single Sign-on Software |            |
| Package .....   | 87         |
| Authentication Using neither IWA nor Single Sign-on Software .....              | 88         |
| QlikView Server Authentication Using Custom Users .....                         | 89         |
| Authorization .....   | 90         |
| Document Level Authorization .....  | 91         |
| NTFS vs. DMS .....  | 91         |
| Data Level Authorization .....  | 91         |
| Dynamic Data Reduction .....  | 92         |
| Static Data Reduction .....   | 92         |
| Certificate Trust .....   | 93         |
| Architecture .....  | 93         |
| Requirements .....  | 95         |
| General .....   | 95         |
| Expiration .....  | 95         |
| Undecryptable Data .....  | 95         |
| Communication Ports .....   | 96         |
| Access .....  | 97         |
| Installation .....  | 97         |
| Enabling SSL .....  | 97         |
| Adding Services to Issue the Certificates .....                                 | 98         |
| Updating Certificates .....   | 100        |
| Backup .....  | 101        |
| Restoring Certificates .....  | 101        |
| Configuration Files .....   | 101        |
| Using Microsoft Management Console .....  | 102        |
| 2.4 Logs and error codes .....  | 103        |
| Logging from QlikView Server .....  | 103        |
| Session Log .....   | 104        |
| Performance Log .....   | 106        |
| Event Log .....   | 109        |
| End-user Audit Log .....  | 109        |
| Manager Audit Log .....   | 113        |
| Task Performance Summary .....  | 114        |
| 2.5 Licensing .....   | 114        |
| OEM .....   | 115        |
| General .....   | 115        |
| Detailed Function Description .....   | 115        |
| <b>3 Installing QlikView .....</b>  | <b>117</b> |
| 3.1 Installing QlikView Server .....  | 117        |
| Before Installing QlikView Server .....   | 117        |
| Setup Procedure .....   | 117        |
| Logging the Installation .....  | 119        |
| Obtaining the MSI package .....   | 119        |

---

|  |            |
|--|------------|
| Completing the Installation .....                      | 119        |
| Running Microsoft IIS .....                            | 119        |
| Handling Timeouts .....                                | 119        |
| Enabling ASP.NET .....                                 | 120        |
| Optimizing the Performance .....                       | 120        |
| Licensing .....  | 120        |
| 3.2 Silent Installation .....                          | 122        |
| Settings .....   | 123        |
| Dialogs .....  | 123        |
| Region .....   | 124        |
| License Agreement .....                                | 124        |
| Customer Information .....                             | 125        |
| Destination Folder .....                               | 126        |
| Profiles .....   | 127        |
| Logon Information .....                                | 128        |
| Service Authentication .....                           | 129        |
| Ready to Install .....                                 | 130        |
| Additional Dialogs .....                               | 131        |
| Custom Setup .....                                     | 131        |
| Website .....  | 131        |
| MST .....  | 132        |
| 3.3 Deploying MSI Packages with Group Policies .....   | 132        |
| General .....  | 133        |
| Deploying the MSI Package .....                        | 133        |
| Advertising .....                                      | 134        |
| Step-by-step Guide .....                               | 134        |
| <b>3 Upgrading QlikView .....</b>                      | <b>142</b> |
| 3.4 Maintenance contract on upgrade .....              | 142        |
| Restoring an older QlikView Desktop installation ..... | 142        |
| Restoring an older QlikView Server installation .....  | 143        |
| 3.5 Upgrade and Migration .....                        | 143        |
| Best practices .....                                   | 143        |
| QlikView Desktop .....                                 | 143        |
| QlikView Server .....                                  | 143        |
| Upgrade QlikView Desktop .....                         | 144        |
| Upgrade QlikView Server .....                          | 144        |
| Multi-machine Preparation .....                        | 144        |
| Simple Upgrade .....                                   | 144        |
| Update certificates .....                              | 144        |
| Maximize Uptime .....                                  | 145        |
| Migration to a New Machine .....                       | 145        |

# 1 Introduction

In this guide you will find information on how to plan and deploy QlikView, including installation advice.

QlikView Server is a platform for hosting and sharing QlikView information over an intranet or the Internet. QlikView Server connects users, client types, documents, and objects within a secure environment.

QlikView Publisher manages content, access, and distribution. By reducing data, each user can be presented with tailored information. The QlikView Publisher service and user interface are fully integrated into QlikView Server and QlikView Management Console (QMC).



## 2 Planning QlikView Deployments

This section provides details on the QlikView architecture, deployment, security, logging and licensing.



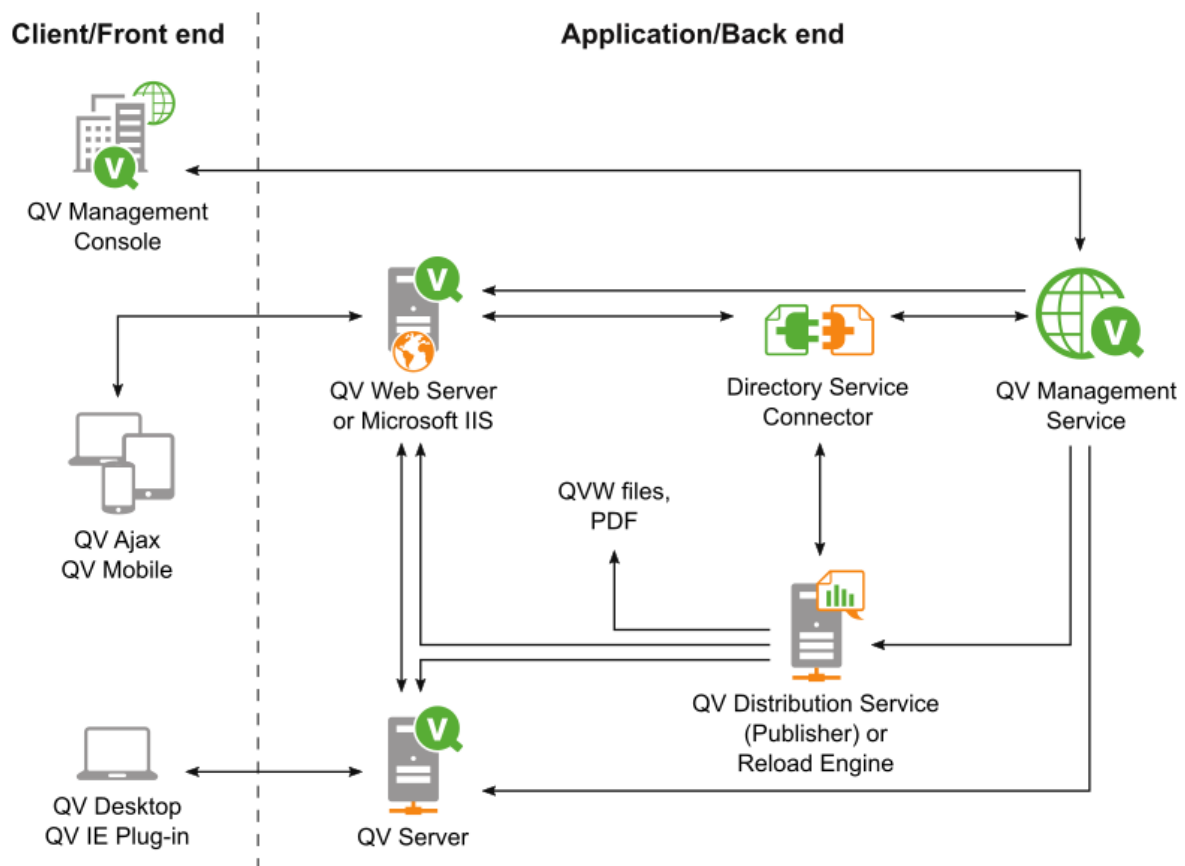
*If Microsoft IIS is to be used as web server, it must be installed prior to QlikView Server.*



*IPv4 is required for installation of QlikView Server. IPv6 is currently unsupported.*

### 2.1 Architecture

The overall architecture of a QlikView installation reflects the separation of roles.



*Example of a QlikView deployment*

#### Front End

The front end is where end users interact with the documents and data that they are authorized to see via QlikView Server. The front end contains the QlikView user documents that typically have been created via QlikView Publisher (QlikView Distribution Service with Publisher license) at the back end. All

---

## 2 Planning QlikView Deployments

---

communications between the client and server take place here and QlikView Server is fully responsible for the client authorization.

The front end relies on infrastructure resources (for example, Windows-based file share for clustering).



*QlikView Server currently only conforms with Windows file sharing. This means that storage must be owned, governed, and shared by a Windows operating system instance (typically accessed using a path like \\<servername>\<share>).*



*QlikView does not support Windows Distributed File System (DFS).*

Authentication of end users is (with exception of the built-in Custom Users) handled outside QlikView.

### Back End

The back end is where the QlikView source documents, created using QlikView Developer, reside. These source files contain scripts to extract data from various data sources (for example, data warehouses, Microsoft Excel® files, SAP®, and Salesforce.com®). This extraction sometimes involves intermediate files (QVD files). The main QlikView component that performs the loading and distribution at the back end is the Distribution Service.

The back end uses the infrastructure resources for clustering (for example, Windows-based file share) and may also use resources like SMTP servers and directory catalogs.



*QlikView Server currently only conforms with Windows file sharing. This means that storage must be owned, governed, and shared by a Windows operating system instance (typically accessed using a path like \\<servername>\<share>).*



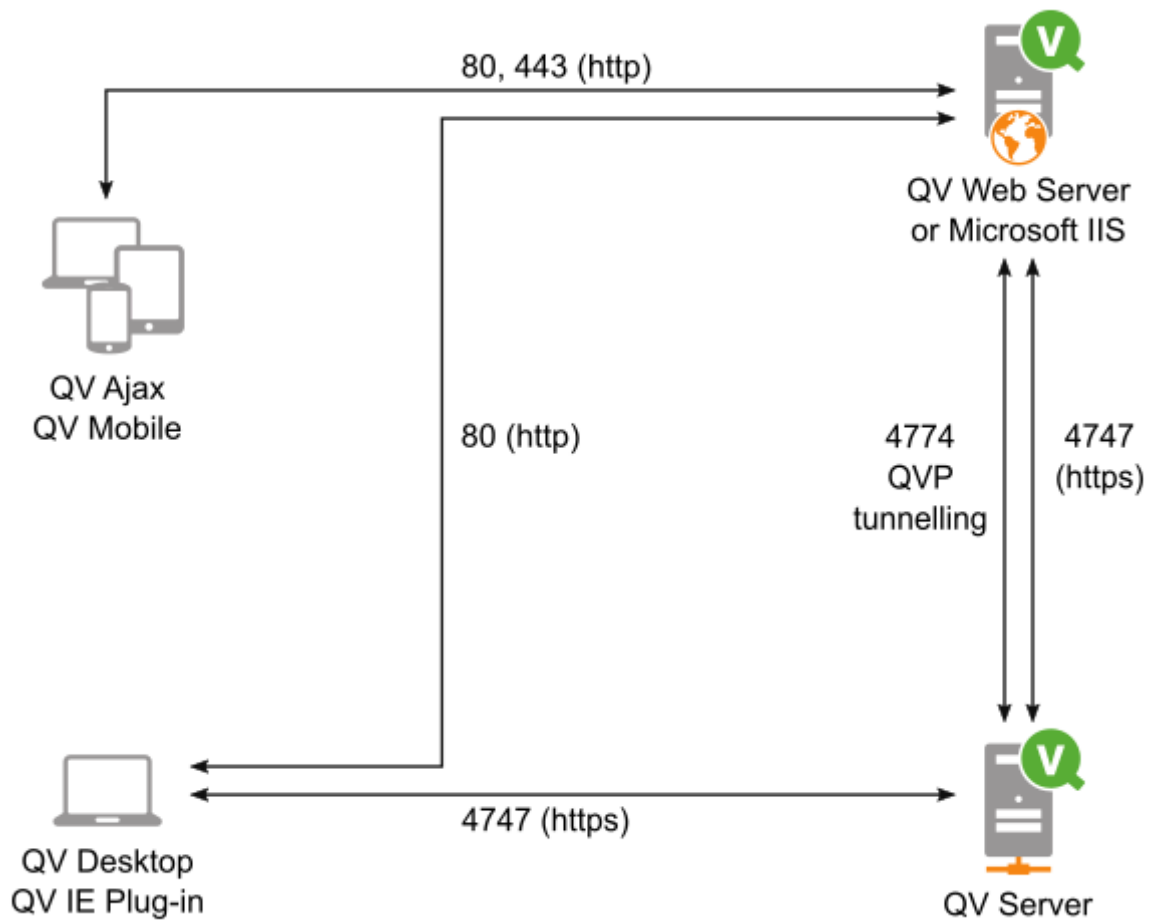
*QlikView does not support Windows Distributed File System (DFS).*

### QlikView Server

The number of servers (clustered or not) within an installation is only limited by the license. It is, however, not feasible to run more than one QlikView Server (QVS) process per server (physical or virtual). QlikView Server is designed to make the most of the resources available to it. Notably the QlikView Server keeps as many calculation results as possible cached in memory to keep the response times to a minimum.

### QlikView Server – Client Communication

The QlikView Server – client communication architecture requires three primary processes, which must be able to communicate with each other in a consistent and secure manner. This interaction can potentially involve multiple machines and multiple network connections, as well as other subordinate processes.



### *QlikView Server – client communication*

The three primary processes are described below.

| Process               | Description  |
|-----------------------|--|
| QlikView Server (QVS) | Provides QlikView functionality to the client. The machine that hosts this service must be running a Microsoft Windows operating system.   |
| Client                | Runs in a web browser or an application shell that provides a container for the client code. The client communicates with QVS either directly or through the web server to provide the QlikView interface and functionality to the end user. |
| Web server            | Runs an http server, which can be used to serve html web pages to the client, assists with authentication of the user, and enables communication between the client and QVS.   |

With the exception of Custom Users, the authentication of client users is done outside QlikView using, for example, Windows authentication.

The protocols defined for client communication with QlikView Server are listed below.

| Protocol                | Description  |
|-------------------------|--|
| QlikView Protocol (QVP) | Encrypted, binary, and TCP-based; communicates directly with QVS on port 4747. |
| QVPX                    | XML-based; communicates with the QVS using http/https through a web server.    |

Windows clients (.exe/.ocx) communicate directly with QlikView Server using QVP on port 4747. These clients do not require a web server to establish and maintain a connection with QlikView Server.

The AJAX client and mobile clients do not communicate directly with QlikView Server. They establish and maintain a connection using the QVPX protocol through a web server, QlikView Web Server (QVWS) or Microsoft IIS. This is normally done using port 80 (http). The web server, in turn, communicates with QVS using the QVPX2 protocol on port 4747.

The default installation settings for QVS use QVWS, not Microsoft IIS. QVWS shares port 80 with IIS on Windows 7 (and later) and Windows Server 2008 (and later).

### QlikView Server access to User Document

For a user to open a document, it is required that:

- There is a Client Access License (CAL) for the user
- The user has access to the document

The user documents are always read by QlikView Server (QVS) and thus technically only need to be readable by the account running QVS. The access rights are either stored in the ACL list of the document (when QVS runs in NTFS mode) or in the .META file (when QVS runs in Document Metadata Service – that is, DMS – mode). These settings are part of the distribution from the back end.

Items (for example, layout, reports, bookmarks, annotations, and input field values) created by end users are stored in shared files. Shared files are not replaced by the distribution from the back end.

### Web Server

QlikView Web Server (QVWS) is included as part of the QlikView Server installation. The web server can act as a standalone service to fulfill the need of many QlikView Server installations.

As an alternative, a Microsoft IIS solution that provides more flexibility, additional authentication schemes, and web services for applications other than QlikView Server can be deployed. When Microsoft IIS is used, a special service, QlikView Settings Service, that handles management calls is installed.

Other web servers can be used in a QVS environment, but at some point the traffic targeting QVS has to go through either QVWS or the dedicated ASPX pages on IIS.

The QlikView Web Server component (either QVWS or IIS-based) performs several tasks:

- Handles the AccessPoint back end
- Transforms/routes traffic between stateless http and to/from the session-based communication with QVS
- Handles load balancing of QVS clusters
- Serves static content (optional)
- Handles authorization of Windows-authenticated users
- Handles authentication of Custom Users (optional)
- Handles group resolution through Windows or Directory Service Connector (DSC) (optional)

### QlikView Server Tunnel

If the QVS communication port (4747) is blocked in the network firewall, Windows clients attempt to re-route their connection through port 80 (http). This connection path must then include the QVWS, or be installed on Microsoft IIS, so that QVS tunnel communication can be established.

### Directory Service Connector

The Directory Service Connector (DSC) is responsible for retrieving user information related to end users from a variety of sources, including (but not limited to) Active Directory, LDAP, ODBC, and Custom Users.

The web server uses DSC for group resolution, the Distribution Service uses it to look up e-mail addresses or UIDs during distribution, and the Management Service uses it to help the administrator find users and groups.

### Management Service

The Management Service is the entry point for all management, both through QlikView Management Console and the QlikView APIs.

The QlikView Management Service (QMS) keeps settings in a database of its own, the QVPR. The QVPR is by default stored as XML files – an alternative is storing the settings in an SQL database.



*All QlikView servers must have the same regional settings. Different regional setting may cause errors when loading QlikView XML reference files.*

An installation can only have a single instance of QMS active. Active/passive failover should be used for redundancy. Note that no other service needs QMS to be running.

### Distribution Service

In a QlikView installation using QlikView Distribution Service, both the back end and the front end are suitable for development, testing, and deployment.

The Distribution Service works with the source documents to produce:

- User documents
- .qvw files for distribution to a folder or via e-mail

- *.pdf* documents for distribution to a folder or via e-mail

The chain of events up to the final distribution involves one or many of the following tasks:

1. Data is loaded from one or more data sources (including QVD) into one or more *.qvw* or *.qvd* files.
2. A document is reduced into one or more smaller documents.
3. Attributes and usage rules are added (applicable only when distributed to a QVS).

The Distribution Service performs the tasks according to defined schedules and/or as responses to events.

### QlikView architecture without Distribution Service

Without Distribution Service, the QlikView architecture becomes more restrictive. All distribution and reduction facilities are removed and replaced by a reload directly on the user documents. Without Distribution Service, developers need to manually deploy the *.qvw* file behind the server.

### Reload Engine

In the absence of a Publisher license connected to the QlikView Distribution Service, the Reload Engine provides a subset of the Publisher distribution services. The Reload Engine only reloads user documents and the settings are defined directly in the user documents.



*All QlikView services must be running on the same machine for the Reload Engine to work. If you install the services on different machines (for example, the QMC, DSC, and QDS on one machine and the QVS and QVWS on another machine), the Reload Engine will not work.*

## Documents, Data, and Tasks

### User Documents

A user document is the document that an end user sees when accessing a document on QlikView Server (QVS). To fully identify a user document, both the QVS server/cluster and the path relative to the server have to be known. Technically, a user document consists of three files:

1. *.qvw* file that contains the data and layout.
2. *.META* file that contains:
  - a. AccessPoint attributes
  - b. Pre-load options
  - c. Authorization (Document Metadata Service – that is, DMS – mode only)
3. Shared file (*.Shared* or *.TShared*, see below)



*If the user document is distributed by the QlikView Distribution Service, both the *.qvw* and the data in the *.META* file are overwritten.*

The access to user documents is controlled by QlikView Server.

### Shared Files

There are multiple objects available for user collaboration and sharing through QlikView Server:

- Bookmarks
- Sheet objects, including charts
- Reports
- Annotations

Each of these objects may be defined as a user object, available to authenticated users, regardless of access method or location, or a shared object, available to all users of the document through QVS.

The objects are configured and managed using QlikView Management Console (QMC).

Once QVS is enabled for server objects, any of the QVS object settings are checked, and the document is opened in QVS, a special database file is created and maintained in the same location as the QlikView document. The file has the same name as the QlikView document, but a shared file extension (*.Shared* or *.TShared*).

#### Example:

- QlikView document: *Presidents.qvw*
- QVS share file: *Presidents.qvw.TShared*

If the name of the QlikView document is changed, the shared file has to be manually renamed to match before opening the renamed QlikView document in QVS. This preserves the shared objects attached to the document.

When updating a Server object, report, bookmark, or input field data, the file is exclusively locked. Making a selection or simply activating the object does not lock the file and any number of servers can read the file at the same time. A partial lock is implemented so that different sections of the file may be updated simultaneously by different servers in a cluster.

The file is read once when the server opens the document, but it is not read again unless there are changes. All sessions share the same internal copy of the shared file (that is, opening a session generally does not require the file to be read from disk).

The server objects can be managed (for example, change of ownership or delete) on the **Documents>User Documents>Server>Server Objects** tab in QMC.

### Source Data

Source data is any external data used to populate the data within a *.qvw* file. The source data is loaded to the *.qvw* at reload time, which can be done:

1. Through the QlikView Distribution Service
2. Through the Reload Engine
3. Manually by the developer

Access to source data is not required for end users to use the .qvw document through QVS once the .qvw file is populated.

### Source Documents

Source documents are only applicable when a Publisher license is applied. Most source documents originate from a developer, others are created by the QlikView Distribution Service as part of the distribution process. QlikView Data files (QVD) can also be created as part of the distribution process as an intermediate step. A QVD file is a table of data stored in format that is optimized for speed when read by QlikView.

The access to source documents is governed by NTFS.

### Tasks

Tasks can be used to perform a wide variety of operations and be chained together in any arbitrary pattern. The starting point when describing tasks is the transformation of a source document into a user document.

### Transforming Source Document into User Document

The transformation starts with a source document and ends in one or many user documents.

#### Source

A task is always tied to a source document, so the source is given.

#### Layout

The source document contains the layout, which is copied unchanged all the way to the user documents. The server side layout is associated with the user document and is also unchanged.

#### Reload

The data can be:

- Used as stored in the document (that is, no reload)
- Partly reloaded from the source (that is, require script preparation)
- Fully reloaded from the source, discarding any old data
- Reloaded in parts by use of "Script Parameters" (which require script preparation)

#### Reduce

The document can be reduced after reload. The reduction can either reduce the input into a smaller document (simple reduce) or split it up into several smaller documents (loop and reduce).

The reduction is based on a selection, either done directly in QMC or using bookmarks.

#### Distribution

Distribution requires a QlikView Publisher license.

The destination is defined as:

- A list of users and a folder on a QlikView Server
- A list of users and a folder in the file system
- A list of users (assuming their e-mail addresses are known)





*“Loop and distribute” must be used, if different content is to be distributed to different users. If not, the same document (or documents) is distributed to all.*

### Information

Information can be associated with the document as part of the distribution to a server. The information is not moved with the document, if it is distributed to another location. The information is used in QlikView AccessPoint.

The following information can be associated with the document:

- Description
- Category
- Arbitrary name value pairs

### Server Settings

The settings for the document are distributed to a server. The settings are not moved with the document, if it is distributed to another location. The settings are enforced by QlikView Server.

Authorization enforced by the server (equal to all servers):

- The users authorized to create server objects
- The users authorized to download the document
- The users authorized to print and export the document to Microsoft Excel

Preferences applied by QlikView AccessPoint (equal to all servers):

- Internet Explorer plugin is recommended
- Mobile client is recommended
- AJAX client is recommended

Performance enforced by the server (equal to all servers):

- Audit logging
- Maximum open sessions
- Document timeout
- Session timeout

Availability (per server):






- Never
- On-demand
- Pre-loaded

## Ports

QlikView uses ports to communicate between web browsers (users) and servers, and between different services in single or multi-node deployments.

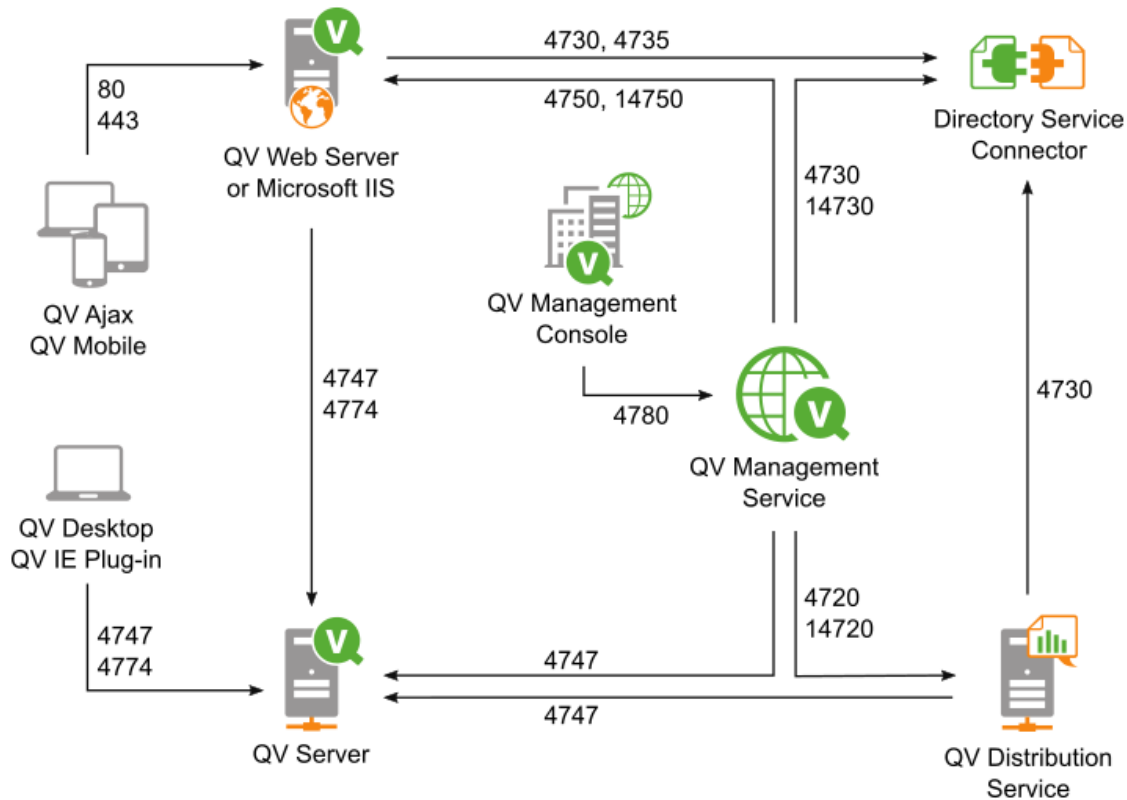
## 2 Planning QlikView Deployments

The following table provides an overview of the ports used in a QlikView deployment.

|   | Component                                  | Inbound  | Outbound  |
|---|--|--|---|
|    | QlikView Server (QVS)                      | 4747 (QVP: QlikView Protocol)<br><br>4774 (QVP Tunnelling)<br><br>14747 (Cluster users SOAP API)<br><br>4749 (SSL) | 14747 (Cluster broadcast)   |
|    | QlikView Web Server (QVWS)                 | 80 (HTTP)<br><br>443 (HTTPS)<br><br>4750 (SOAP API)<br><br>14750 (Certificate)                                     | 4730 (DSC SOAP API)<br><br>4735 (DSC custom users SOAP API)<br><br>4747 (QVS QVP)<br><br>4774 (QVS QVP tunnelling)  |
|   | QlikView Distribution Service (QDS)        | 4720 (SOAP API)<br><br>14720 (Certificate)   | 4730 (DSC SOAP API)<br><br>4747 (QVS QVP)   |
|  | QlikView Directory Service Connector (DSC) | 4730 (SOAP API)<br><br>14730 (Certificate)<br><br>4735 (Custom users SOAP API)                                     |   |
|  | QlikView Management Service (QMS)          | 4780 (HTTP)<br><br>4799 (SOAP API)   | 4747 (QVS QVP)<br><br>4750 (QVWS SOAP API)<br><br>14750 (QVWS Certificate)<br><br>4720 (QDS SOAP API)<br><br>14720 (QDS Certificate)<br><br>4730 (DSC SOAP API)<br><br>14730 (DSC Certificate)<br><br>4735 (DSC custom users)<br><br>4799 (Remote QMS SOAP API) |

## 2 Planning QlikView Deployments

The following example shows the ports used to connect the different QlikView services. In a QlikView deployment, all these services can be installed on the same server (single-node deployment). Alternatively, you can decide to set up a multi-node deployment, and install different services on different servers. For further information about QlikView architecture and deployments, see *Architecture (page 9)* and *Deployment (page 46)* pages.



Example showing the ports used for connecting the different components in a QlikView deployment (QV= QlikView)

### Service by Service

This chapter describes the QlikView Server/Publisher components in detail.



*The account that is used to run the QlikView services must have local administrator privileges.*

### QlikView Server Load Sharing (Clustering)

#### Overview

|                   |   |
|-------------------|---|
| <b>Executable</b> | <code>%ProgramFiles%\Qlik View\Server\QVS.exe</code>  |
| <b>Data</b>       | <code>%ProgramData%\Qlik Tech\Qlik View Server</code> |

## 2 Planning QlikView Deployments

|                      |   |
|----------------------|---|
| <b>Listens to</b>    | QVP: 4747; QVP (tunneling): 4774; Broadcast: 14747; SNMP: 161 |
| <b>Uses/Controls</b> | -   |
| <b>Used by</b>       | QDS, QMS, QVWS, QlikView Desktop/Internet Explorer plugin/OCX |

### Files

#### Settings and Configuration

| File                | Description   |
|---------------------|---|
| <i>Settings.ini</i> | Stores the QlikView Server (QVS) settings. Manual changes in this file require restart of QVS. This file is always stored in the “Data” folder. |

#### Cluster

QVS uses .pgo files to coordinate a cluster. The files are stored in the “Data” folder.

| File                       | Description  |
|----------------------------|--|
| <i>BorrowedCalData.pgo</i> | Keeps track of borrowed Client Access Licenses (CALs). |
| <i>CalData.pgo</i>         | Keeps track of CALs.                                   |
| <i>IniData.pgo</i>         | Coordinated version of <i>Settings.ini</i> .           |
| <i>ServerCounters.pgo</i>  | Keeps track of statistics.                             |
| <i>TicketData.pgo</i>      | Keeps track of tickets.                                |


#### Logs

The logs are kept one per node in the cluster. The log files are stored in the “Data” folder by default.

| File   | Description      |
|--|------------------|
| <i>Events_&lt;computer_name&gt;.log</i>      | Event log.       |
| <i>Performance_&lt;computer_name&gt;.log</i> | Performance log. |
| <i>Sessions_&lt;computer_name&gt;.log</i>    | Session log.     |

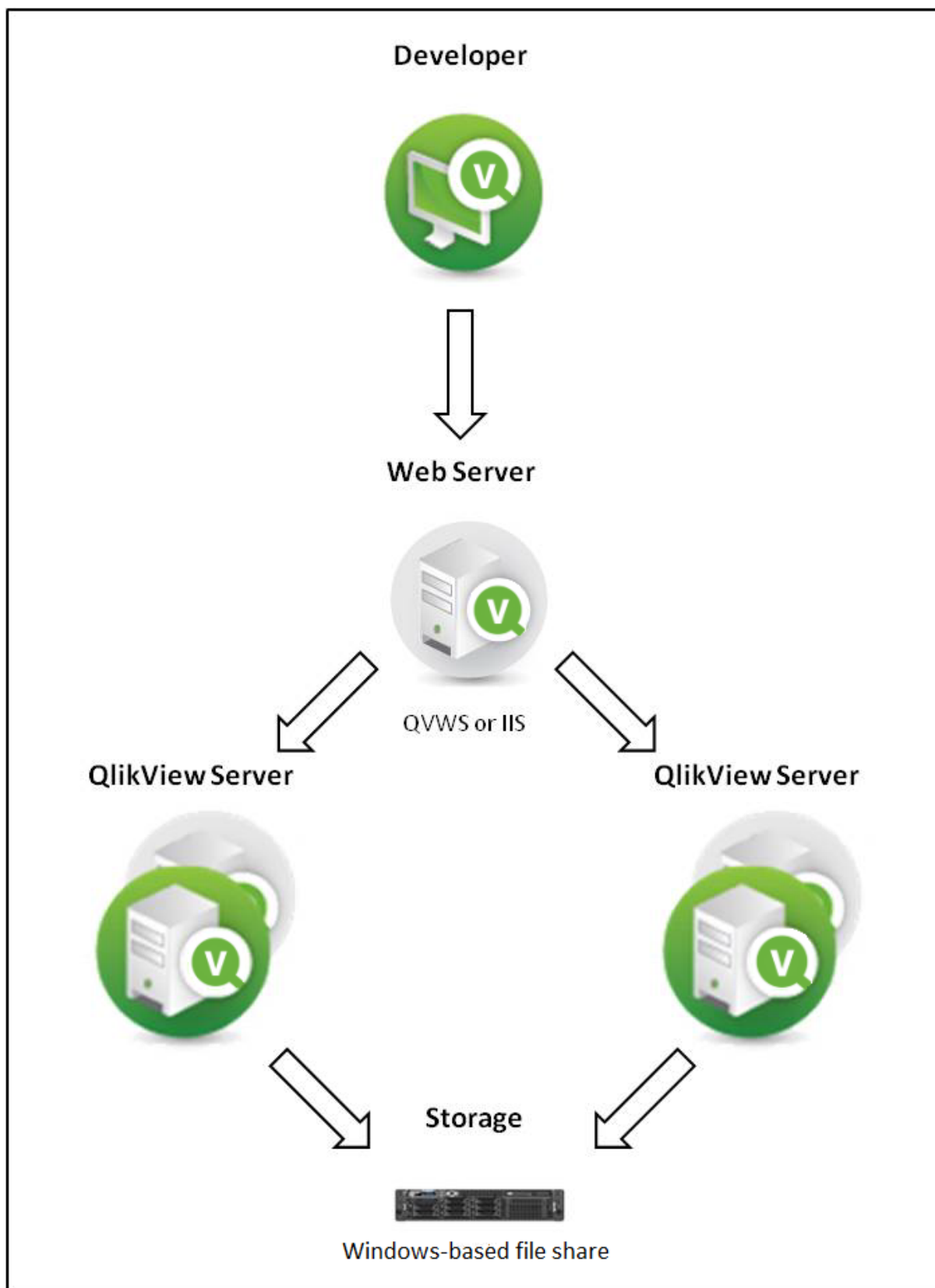
#### Special Folders

The special folders are stored in the “Data” folder.

| Folder            | Description   |
|-------------------|---|
| <i>Extensions</i> | <div> <i>The Extensions folder has to be created manually.</i></div> <p>By default, QVS looks for extensions in this folder. Extension objects are located in <i>Extensions\Objects</i> and document extensions are located in <i>Extensions\Document</i>. Use QlikView Management Console (QMC) to manage all extensions in one place in case of a cluster.</p> |
| <i>Temp</i>       | By default, QVS puts temporary files in this folder (for example, when exporting using the AJAX client, a temporary file is created in the folder).   |

### Load Sharing (Clustering)

All clustering requires a cluster-enabled QlikView Server license. QlikView Server supports load sharing of documents across multiple machines. This sharing includes the ability to share in real time, information about server objects, automated document loading, and user license CALs. Special licensing is available to enable multiple server instances share the same license number.



*Load sharing using QlikView Web Server*

## 2 Planning QlikView Deployments

To use load sharing between multiple QVSs, all document and support files must be shared between the servers. In other words, all servers should point to the same physical location for the files. QVS creates and maintains additional files to store load sharing data. These files have a Persistent Group Object (.pgo) file type extension and are located in the “Data” folder. These files are locked when QVS is running. The different .pgo files contain information on borrowed CALs, CALs in use, server settings, and ticket data.

Operating system load balance or failover configurations are external to the QVS load sharing configuration, and QVS has no control over those systems.

Server configuration settings are shared between all clustered QVSs and can be maintained through QMC connected to any of the clustered QVSs. Performance of a particular QVS system can be monitored through QMC by connecting to that system. The load balancing settings, that is, which QVS the client should be directed to, are stored in QlikView Web Server (QVWS).

Document-related meta data is shared via .meta files (one per document). This data is often referred to as Document Metadata Service (DMS) data. Since DMS data is shared among the QVSs, any automated document load procedures are performed on all servers. DMS authorization is also shared among all clustered QVSs.

### QlikView Distribution Service

#### Overview

|                      |  |
|----------------------|--|
| <b>Executable</b>    | %ProgramFiles%\QlikView\Distribution Service\QVDistributionService.exe |
| <b>Data</b>          | %ProgramData%\QlikTech\DistributionService                             |
| <b>Listens to</b>    | HTTP: 4720; SNMP: 4721   |
| <b>Uses/Controls</b> | DSC, QVS, QVB  |
| <b>Used by</b>       | QMS  |



*After restarting the machine, the Windows event log may contain a message that the QlikView Distribution Service (QDS) failed to start in a timely manner, even though it started successfully. This is because the QDS initialization phase is longer than the Windows timeout period (30 seconds by default). To avoid the event log message, either change the Windows timeout period or configure QDS to depend on another late starting service to make QDS start up during a less busy period.*

#### Files

The QlikView Distribution Service (QDS) files can be divided into three groups based on main purpose. All files are stored in the QDS “Data” folder.

In a clustered setup, all QDSs must share the same program folder. This is solved by the file *config\_<computer\_name>.xml*, which contains the program data path to use.

#### Settings and Configuration

The files listed below are local copies of the information stored in QVPR.

## 2 Planning QlikView Deployments

| File                                       | Description  |
|--|--|
| <i>Configuration.xml</i>                   | Configuration file for the service.  |
| <i>Tasks\Task_&lt;GUID&gt;.xml</i>         | The actual tasks. Note that deleted tasks are not automatically removed (due to support issue analysis).                             |
| <i>Triggers\Triggers_&lt;GUID&gt;.xml</i>  | The actual triggers. Note that deleted triggers are not automatically removed (due to support issue analysis).                       |
| <i>MasterConfigurationNotification.xml</i> | A list of configuration notification files. Used to keep QDS in sync and to notify QDS nodes of configuration changes.               |
| <i>MasterTaskNotification.xml</i>          | A list of task notification files. Used to keep QDS in sync and to notify QDS nodes of task changes.                                 |
| <i>MasterTaskExecutionNotification.xml</i> | A list of task execution notification files. Used to keep QDS in sync and to notify QDS nodes of task execution changes.             |
| <i>MasterTriggerNotification.xml</i>       | A list of trigger notification files. Used to keep QDS in sync and to notify QDS nodes of changes to triggering events.              |
| <i>TaskDetails.xml</i>                     | A list of the available tasks in the <i>Tasks</i> folder. In addition, used to synchronize the files in that folder with QVPR.       |
| <i>TriggerDetails.xml</i>                  | A list of the available triggers in the <i>Triggers</i> folder. In addition, used to synchronize the files in that folder with QVPR. |
| <i>DistributionGroupDefinition.xml</i>     | Configuration file for Distribution Groups   |

### Cluster

| File                       | Description   |
|----------------------------|---|
| <i>LoadBalancer.xml</i>    | Used by QDS to decide which node (in a cluster) should run a task.      |
| <i>NodeInformation.xml</i> | Contains all other QDS node data that is not used by the load balancer. |

### Logs

| File  | Description  |
|---|--|
| <i>TaskResults\TaskResult_&lt;GUID&gt;.xml</i>    | Latest result of the task identified by the GUID.  |
| <i>TaskLogIndex\TaskLogIndex_&lt;GUID&gt;.xml</i> | This is just for lookup (one file per task), pointing to the actual log.   |
| <i>EdxResults\EdxResult_&lt;GUID&gt;.xml</i>      | Until the task is completed, this file contains the current status of the EDX task. When the execution is finished, it contains the result (success/fail) and the task started as a result (if any). |
| <i>&lt;node-nr&gt;\Log\&lt;Date&gt;.txt</i>       | General QDS event and error log.   |



|  |  |
|--|--|
| <code>&lt;node-nr&gt;\Log\Cluster_<br/>&lt;Date&gt;.txt</code>   | Synchronization log.   |
| <code>&lt;node-nr&gt;\Log\LoadBalancer_<br/>&lt;Date&gt;.txt</code>  | Load balancing log.  |
| <code>&lt;node-nr&gt;\Log\Root_<br/>&lt;Date&gt;.txt</code>  | QDS event log.   |
| <code>&lt;node-nr&gt;\Log\WebService_<br/>&lt;Date&gt;.txt</code>  | QDS event log.   |
| <code>&lt;node-nr&gt;\Log\Workorder_<br/>&lt;Date&gt;.txt</code>   | QDS event log.   |
| <code>&lt;node-nr&gt;\Log&lt;date&gt;\&lt;time&gt;<br/>- &lt;task name&gt;\Tasklog.txt</code>                | QDS task event log.  |
| <code>&lt;node-nr&gt;\Log&lt;date&gt;\&lt;time&gt;<br/>- &lt;task<br/>name&gt;\DistributionReport.xml</code> | The distribution related to the task (only exists for distribution tasks). |

### Changing the storage time of log files

By default, log files are stored for 30 days and then are deleted from the Application Data Folder, by default `C:\ProgramData\QlikTech\DistributionService`.

You can change the storage time of log files within the `QVDistributionService.exe.config` file.



*It is recommended that you make a copy of `QVDistributionService.exe.config` as a backup before you edit `QVDistributionService.exe.config`.*

Do the following:

1. Open Windows Services.
2. Stop the QlikView Distribution Service by right-clicking the service and clicking **Stop**.
3. Close Windows Services.
4. Browse to `C:\Program Files\QlikView\DistributionService` and open `QVDistributionService.exe.config` with a text editing program.
5. Locate `<add key="NbrofDaysToKeepQDSLogs" value="30" />` and add the number of days reports are to be stored as the value.
6. Save and close the file.
7. Open Windows Services.
8. Restart the QlikView Distribution Service by right-clicking the service and clicking **Start**.
9. Close Windows Services.

### QlikView Batch

#### Overview

|                      |  |
|----------------------|--|
| <b>Executable</b>    | %ProgramFiles%\QlikView\Distribution Service\qvb.exe |
| <b>Data</b>          | -  |
| <b>Listens to</b>    | COM  |
| <b>Uses/Controls</b> | -  |
| <b>Used by</b>       | QDS  |



*QlikView Batch (QVB) does not support graphical or user input objects. This means that QVB cannot reload documents that, for example, contain scripts that require user input.*

#### Files

### Settings and Configuration

| File         | Description             |
|--------------|-------------------------|
| Settings.ini | Used to store settings. |

### Logs

| File                | Description  |
|---------------------|--|
| <document_name>.log | Reload log that is placed together with the reloaded document. |

### QlikView Publisher Repository

#### Overview

|                      |   |
|----------------------|---|
| <b>Executable</b>    | -   |
| <b>Data</b>          | %ProgramData%\QlikTech\ManagementService\QVPR |
| <b>Listens to</b>    | -   |
| <b>Uses/Controls</b> | -   |
| <b>Used by</b>       | QMS   |

#### Files

By default, QlikView Publisher Repository (QVPR) is a set of XML files. These files are backed up as .zip files in %ProgramData%\QlikTech\ManagementService\QVPR\Backups.

### Security Groups

When installing QlikView Server/Publisher, a couple of security groups are created.

The QlikView Server/Publisher services must run under an account that is member of the security group QlikView Administrators. Users connecting to QMC must be part of this group. Anyone connecting to a remote service must also be member of QlikView Administrators.

The users connecting through the API must be members of the QlikView Management API security group. The group is not created during the installation and has to be added (and populated, for example, with the members of the QlikView Administrators group) manually. A membership in this group is required to import tasks from another QlikView Server/Publisher.

The QlikView EDX security group is not created during the installation and has to be added (and populated) manually in order for users to run EDX tasks.

### Document Administrators

To delegate the responsibility of creating tasks to people not part of the QlikView Administrators group, users can be appointed document administrators. The document administrators are only allowed to access the tabs in QMC that are related to either user documents or source documents.



*The use of document administrators requires a QlikView Publisher license.*

For more information on how to appoint document administrators, see the QMC online help.

### Configuration Files



*Use QMC to set the parameters described in this section, since modifying the configuration files directly may cause problems.*

### Management Service – QVManagementService.exe.config

In a default installation, this file is located in `%ProgramFiles%\QlikView\Management Service`. The file has a number of automatically generated tags that should not be modified, but the settings listed below can be modified.

| Setting               | Description   |
|-----------------------|---|
| ApplicationDataFolder | Folder where the log folder and all other files/folders are created. The default value is <code>%ProgramData%\Qlik Tech\ManagementService</code> . This folder is where the XML version of QVPR and the LEF information are stored. |
| UseHTTPS              | True = Communication runs over https. A certificate for the web site is needed to enable this setting.  |
| Trace                 | Used for debug logging.   |

## 2 Planning QlikView Deployments

| Setting                   | Description   |
|---------------------------|---|
| QMSBackendListenPort      | Port that the back end management service listens to. The default value is 4799.  |
| QMSFrontendWebServicePort | Port that the front end management service listens to. The default value is 4780.   |
| MaxLogRecords             | Maximum number of log records that should be retrieved for a task.  |
| EnableAuditLogging        | True = Track a) changes on tasks and settings made in the system, b) who made the changes, and c) when the changes were made. |
| AuditLogFolder            | Path to the folder where the audit logs are saved.  |
| AuditLogKeepMaxDays       | Maximum number of days each log is saved.   |

### Distribution Service – QVDistributionService.exe.config

In a default installation, this file is located in *%ProgramFiles%\QlikView\Distribution Service*. The app settings tag is the part that can be modified. Some of the settings in the configuration file are described below.

| Setting                          | Description   |
|----------------------------------|---|
| ApplicationDataFolder            | Folder where the log folder and all other files/folders are created. The default value is <i>%ProgramData%\Qlik Tech\DistributionService</i> . This folder is where the XML version of QVPR and the LEF information are stored.         |
| WebservicePort                   | Port that the QlikView Distribution Service uses to communicate with. The default value is 4720.  |
| UseHTTPS                         | True = Communication runs over https.   |
| DSCAddress                       | Port that the Directory Service Connector service uses to communicate with. The default value is 4730. If the value is modified, the tag "DSCAddress" in the <i>QVDirectoryServiceConnector.exe.config</i> file has to be modified too. |
| DSCTimeoutSeconds                | Timeout for calls to the Directory Service Connector.   |
| DSCCacheSeconds                  | How long the service caches the responses from the Directory Service Connector.   |
| QlikViewEngineQuarantineTimeInms | How often a QlikView engine is allowed to start (in milliseconds).  |
| OpenDocumentAttempts             | How many tries that can be made to open a document before it is logged as an error during distribution.   |
| DebugLog                         | True = Enable logging of memory usage and stack trace on "Error" logging.   |

## 2 Planning QlikView Deployments

| Setting                       | Description   |
|-------------------------------|---|
| Trace                         | True = Enable debug logging.  |
| EnableBatchMode               | Enable this setting to make batch calls to the QlikView Distribution Service.   |
| ServiceStopGracetimeInSeconds | The time in seconds allowed for tasks running in the QlikView Distribution Service (QDS) to complete, when a request is made from the QMC to shut down the QDS.<br><br>The default value is 1800. |

### Directory Service Connector – QVDirectoryServiceConnector.exe.config

This file is by default located in *%ProgramFiles%\QlikView\Directory Service Connector\QVDirectoryServiceConnector.exe.config*. The settings most commonly modified are listed below.

| Setting               | Description   |
|-----------------------|---|
| ApplicationDataFolder | Folder where the log folder and all other files/folders are created. The default value is <i>%ProgramData%\QlikTech\DirectoryServiceConnector</i> .   |
| WebservicePort        | Port that the Directory Service Connector service uses to communicate with. The default value is 4730. If the value is modified, the tag "DSCAddress" in the <i>QVDistributionService.exe.config</i> file has to be modified too. |
| UseHTTPS              | True = Communication runs over SSL instead of http. A certificate for the web site is needed to enable this setting.  |
| PluginPath            | Path where the Directory Service Connector looks for available DSP plugins. The default value is <i>%ProgramFiles%\QlikView\Directory Service Connector\DSPlugins</i> .   |
| Trace                 | True = Enable debug logging.  |
| DisableCompress       | Enable this setting to disable compression of the http communication.   |

### QlikView Web Server

The web server can be the built-in QlikView Web Server (QVWS) or Microsoft IIS. QVWS is installed as a Windows service during a default, complete installation of QlikView Server. When IIS is used, the same functionality is provided by a set of ASPX pages and a special support service, QlikView Settings Service (QSS). QSS acts as the management interface for settings used by the ASPX pages.

#### Overview

##### QlikView Web Server

|            |  |
|------------|--|
| Executable | <i>%ProgramFiles%\QlikView\Server\Web Server\QVWebServer.exe</i> |
|------------|--|

## 2 Planning QlikView Deployments

|                      |  |
|----------------------|--|
| <b>Data</b>          | %ProgramData%\QlikTech\WebServer       |
| <b>Listens to</b>    | HTTP: 80; HTTP: 4750; SNMP: 4751       |
| <b>Uses/Controls</b> | DSC                                    |
| <b>Used by</b>       | Web browser clients and mobile clients |

### QlikView Settings Service

|                   |   |
|-------------------|---|
| <b>Executable</b> | %ProgramFiles%\QlikView\Server\Web Server Settings\QVWebServerSettingsService.exe |
| <b>Data</b>       | %ProgramData%\QlikTech\WebServer  |
| <b>Listens to</b> | HTTP: 4750  |
| <b>Used by</b>    | QMS   |

### Files

#### Settings and Configuration

| File              | Description                         |
|-------------------|-------------------------------------|
| <i>Config.xml</i> | Configuration file for the service. |

### Logs

| File                        | Description          |
|-----------------------------|----------------------|
| <i>Log\&lt;date&gt;.txt</i> | Event and error log. |

### Configuring the QlikView Web Service

You may configure the web server either through the Qlik Management Console. Additional configuration can be done by editing the *config.xml* file, found in the following location:

*C:\ProgramData\QlikTech\WebServer*

The *config.xml* file contains the following section that is commented out to simplify the usage of common but non-default options.

```
<Config>
<DefaultUrl>http://_/</DefaultUrl>
<DefaultQvs>local</DefaultQvs>
<ConfigUrl>http://_:4750/qws.asmx</ConfigUrl>
<TunnelUrl>/scripts/QVSTunnel.dll</TunnelUrl>
<QvsStatusUrl>/QvAjaxZfc/QvsStatus.aspx</QvsStatusUrl>
<LogLevel>Information</LogLevel>
<UseCompression>True</UseCompression>
<InstallationPath>C:\Program Files\Qlikview\Server\Web Server</InstallationPath>
<QvsTimeout>60</QvsTimeout>
<QvsAuthenticationProt>Negotiate</QvsAuthenticationProt>
<QvpPort>-1</QvpPort>
<AddCluster>
```

```
<Name>local</Name>
<LoadBalancing>Random</LoadBalancing>
<AlwaysTunnel>False</AlwaysTunnel>
<AddQvs>
<Machine>localhost</Machine>
<Port>4747</Port>
<LinkMachineName>RD-CENTEST1</LinkMachineName>
<Weight>1</Weight>
</AddQvs>
</AddCluster>
<AddDSCCluster>
<CustomUserPort>-1</CustomUserPort>
<DirectoryServiceConnectorSettings>
<ID>17da91ee-c4a6-4cdb-a2fb-ab472ece659f</ID>
<Url>http://rd-centest1:4730/qtds.asmx</Url>
<Name>Default DSC</Name>
<Username>DxdCGMwfOWU=</Username>
<Password>DxdCGMwfOWU=</Password>
<LogLevel>Normal</LogLevel>
</DirectoryServiceConnectorSettings>
</AddDSCCluster>
<Authentication>
<AuthenticationLevel>Always</AuthenticationLevel>
<LoginAddress>/qlikview/login.htm</LoginAddress>
<LogoutAddress>logout.htm</LogoutAddress>
<GetTicket url="/QvAjaxZfc/GetTicket.aspx" />
<HttpAuthentication url="https://_/scripts/GetTicket.asp" scheme="Basic" />
<HttpAuthentication url="/QvAJAXZfc/Authenticate.aspx" scheme="Ntlm" />
</Authentication>
<AccessPoint>
<Path>/QvAJAXZfc/AccessPoint.aspx</Path>
<AjaxClientPath>/QvAJAXZfc/opendoc.htm</AjaxClientPath>
<PluginClientPath>/QvPlugin/opendoc.htm</PluginClientPath>
<DefaultPreferredClient>Ajax</DefaultPreferredClient>
<DefaultView>Thumbnails</DefaultView>
<DefaultPagesizeDetails>40</DefaultPagesizeDetails>
<DefaultPagesizeThumbnails>4</DefaultPagesizeThumbnails>
<HighlightNotExecutedJobs>False</HighlightNotExecutedJobs>
<HighlightThresholdMinutes>60</HighlightThresholdMinutes>
<AllowCmdUrl>False</AllowCmdUrl>
<Target />
<RespectBrowsable>True</RespectBrowsable>
</AccessPoint>
<Ajax>
<Path>/QvAJAXZfc/QvsViewClient.aspx</Path>
<Path>/QvAJAXZfc/QvsViewClient.asp</Path>
<NoCrypto>False</NoCrypto>
<ProhibitMachineId>False</ProhibitMachineId>
<Recording>False</Recording>
<AllowCmdUrl>True</AllowCmdUrl>
</Ajax>
<web>
<Folders>
<Folder>
<Name>QlikView</Name>
<Path>C:\Program Files\QlikView\Web</Path>
</Folder>
<Folder>
<Name>QvClients</Name>
```

```
<Path>C:\ProgramFiles\QlikView\Server\QvClients</Path>
</Folder>
<Folder>
<Name>QvAJAXZfc</Name>
<Path>C:\ProgramFiles\QlikView\Server\QvClients\QvAjaxZfc</Path>
</Folder>
<Folder>
<Name>QvDesktop</Name>
<Path>C:\Program Files\QlikView\Server\QlikviewClients\QlikviewDesktop</Path>
</Folder>
<Folder>
<Name>QvPlugin</Name>
<Path>C:\Program Files\QlikView\Server\QvClients\QvPlugin</Path>
</Folder>
</Folders>
<Types>
<Type>
<Extension>.css</Extension>
<Content>text/css</Content>
</Type>
<Type>
<Extension>.htm</Extension>
<Content>text/html</Content>
</Type>
<Type>
<Extension>.html</Extension>
<Content>text/html</Content>
</Type>
<Type>
<Extension>.jpg</Extension>
<Content>image/jpg</Content>
</Type>
<Type>
<Extension>.gif</Extension>
<Content>image/gif</Content>
</Type>
<Type>
<Extension>.jar</Extension>
<Content>application/octet-stream</Content>
</Type>
<Type>
<Extension>.png</Extension>
<Content>image/png</Content>
</Type>
<Type>
<Extension>.exe</Extension>
<Content>application/octet-stream</Content>
</Type>
<Type>
<Extension>.msi</Extension>
</Type>
<Type>
<Extension>.htc</Extension>
<Content>text/xml</Content>
</Type>
<Type>
<Extension>.js</Extension>
<Content>text/javascript</Content>
</Type>
```



```
<Type>
<Extension>.xslt</Extension>
<Content>text/xml</Content>
</Type>
<Type>
<Extension>.xml</Extension>
<Content>text/xml</Content>
</Type>
<Type>
<Extension>.xls</Extension>
<Content>application/vnd.ms-excel</Content>
</Type>
<Type>
<Extension>.csv</Extension>
<Content>application/octet-stream</Content>
</Type>
<Type>
<Extension>.pdf</Extension>
<Content>application/pdf</Content>
</Type>
</Types>
</Web>
</Config>
```

The following table describes the tags listed in the example.

| Tag                          | Description  |
|------------------------------|--|
| <b>DefaultURL</b>            | The URL of the QlikView Server.  |
| <b>ConfigURL</b>             | This is the URL the QMC uses to communicate with the QlikView Web Server.                              |
| <b>TunnelURL</b>             | The URL used for tunneling.  |
| <b>QvsStatusURL</b>          | The URL to the status page for the QlikView Server.  |
| <b>LogLevel</b>              | Sets the level of logging. Possible settings are Information (High), warning (Medium) and Error (Low). |
| <b>UseCompression</b>        | Set whether the information sent should be compressed.   |
| <b>InstallationPath</b>      | The installation path of the QlikView Web Server.  |
| <b>QvsTimeout</b>            | The timeout in seconds of the QlikView Server.   |
| <b>QvsAuthenticationProt</b> | How the QlikView Server Authenticates. Set to Negotiate, Kerberos or NTLM.                             |

## 2 Planning QlikView Deployments

---

| Tag   | Description  |
|---|--|
| <b>AddCluster - Name</b>  | The name of the cluster.   |
| <b>AddCluster - LoadBalancing</b>                                   | How the load balance should be calculated.<br>Possible values are <code>Random</code> , where the client is directed to a QVS at random, <code>CpuUsage</code> where the QVS reporting the least average CPU will be selected or <code>LoadedDocument</code> , where the client is directed to the QVS where the document the client requests is already loaded. |
| <b>AddCluster - AddQvs - AlwaysTunnel</b>                           | Set to <code>true</code> to always tunnel the communication to the QlikView Server.  |
| <b>AddCluster - AddQvs - Machine</b>                                | The name of the computer where the QlikView Server is running.   |
| <b>AddCluster - AddQvs - Port</b>                                   | The port the QlikView Server listens to.   |
| <b>AddCluster - AddQvs - LinkMachineName</b>                        | The external name of the QlikView Server, used by the QlikView Plug-in clients.  |
| <b>AddCluster - AddQvs - weight</b>                                 | Set a higher value if you wish that the QlikView Server be selected more frequently when using random load balancing.  |
| <b>AddDSCCluster - CustomUserPort</b>                               | The port for the custom user Directory Service Connector.  |
| <b>AddDSCCluster - DirectoryServiceConnectorSettings - Url</b>      | The location of the Directory Service Connector.   |
| <b>AddDSCCluster - DirectoryServiceConnectorSettings - Name</b>     | The cluster name.  |
| <b>AddDSCCluster - DirectoryServiceConnectorSettings - Username</b> | Enter a user name if needed to connect to the Directory Service Connector.   |

## 2 Planning QlikView Deployments

---

| Tag   | Description  |
|---|--|
| <b>AddDSCCluster - DirectoryServiceConnectorSettings - Password</b> | Enter a password if needed to connect to the Directory Service Connector.  |
| <b>Authentication - AuthenticationLevel</b>                         | Sets how the client should access the AccessPoint. Possible values are Always, Login and Never.                  |
| <b>Authentication - LoginAddress</b>                                | The path to an alternative login page used for custom users.   |
| <b>Authentication - LogoutAddress</b>                               | The path to an alternative logout page used for custom users.  |
| <b>Authentication - GetTicket</b>                                   | The URL and authentication used to get a ticket from the Server for a client.                                    |
| <b>Authentication - HttpAuthentication</b>                          | The URL and authentication used to get a ticket from the Server for a client if using SSL.                       |
| <b>AccessPoint - Path</b>   | The path where the Access Point is installed.  |
| <b>AccessPoint - AjaxClientPath</b>                                 | The relative path to the Ajax client.  |
| <b>AccessPoint - PluginClientPath</b>                               | The relative path to the IE plug-in client.  |
| <b>AccessPoint - DefaultPreferredClient</b>                         | Sets which client should be set as the preferred client for a user's first visit to the AccessPoint for clients. |
| <b>AccessPoint - DefaultView</b>                                    | The default view of documents on the AccessPoint, details or thumbnails.   |
| <b>AccessPoint - DefaultPagesizeDetails</b>                         | The number of rows on the AccessPoint when using the Details view.   |

## 2 Planning QlikView Deployments

---

| Tag  | Description  |
|--|--|
| <b>AccessPoint - DefaultPagesizeThumbnails</b> | The number of rows on the AccessPoint when using the Thumbnails view.  |
| <b>AccessPoint - RespectBrowsable</b>          | When set to <code>True</code> only mounts that are set as <code>Browsable</code> in the QVS are displayed on the AccessPoint.  |
| <b>Ajax - Path</b>                             | The path to <i>QvsViewClient.aspx</i> . The path may be changed, but the file name must remain unchanged for the installation to work.   |
| <b>Ajax - NoCrypto</b>                         | Prohibit the use of encryption between the QlikView Web Server and the QlikView Server.  |
| <b>Ajax - ProhibitMachineID</b>                | Prohibit sending the machine ID. This will effectively exclude the usage of anonymous bookmarks.   |
| <b>Ajax - Recording</b>                        | When set to <code>True</code> , the qvpx calls for the AJAX zero footprint client are logged.  |
| <b>SafeForwardList</b>                         | When a redirect is requested through <i>Authenticate.aspx</i> , a DNS lookup is done to retrieve the IP addresses of the path provided in this tag. If the IP addresses matches that of the redirect request, the redirect is allowed. |

### Tag

**StrictSafeForwardList**

### Description

When a redirect is requested through *Authenticate.aspx*, the host name of the path provided in this tag is compared with the host name of the incoming redirect path. If they match (not case sensitive), the redirect is allowed.

**web - Folders**

The path to the different virtual folders in the QlikView Web Server. Change the name and path if the files are installed to folders other than the default.

**web - Types**

Specify what file extensions the clients are allowed to download from the Access Point/QlikView Web Server.

### Load Balancing

QVWS hosts web pages, prepares the file list for AccessPoint, and manages the load balancing of QlikView Servers (QVSs).

AccessPoint is a web portal for documents hosted on QVWS. The pages for AccessPoint are by default located in the folder *%ProgramFiles%\QlikView\Web*. QVWS also acts as web server for any AJAX pages accessed by the end users.

The load balancing performed by QVWS is different from load balancing a web server, since the additional work and resource consumption is almost similar for each user, so it does not matter on which server the user ends up.

The load balancing schemes are listed below.

| Scheme          | Description  |
|-----------------|--|
| Random          | The default load balancing scheme. The user is sent to a random server, no matter if the document the user is looking for is loaded or not.  |
| Loaded Document | If only one QVS has the particular document loaded, the user is sent to that QVS. If more than one QVS or none of the QVSs has the document loaded, the user is sent to the QVS with the largest amount of free RAM. |

## 2 Planning QlikView Deployments

| Scheme                | Description                             |
|-----------------------|---|
| CPU with RAM Overload | The user is sent to the least busy QVS. |

The settings for load balancing are configured in QMC.

### QlikView AccessPoint

QlikView AccessPoint is a web portal that lists the documents each user has access to. AccessPoint only links to each document – it does not host the documents. The hosting is done by QlikView Server.

The documents can be displayed as thumbnails or in a detailed list.

The screenshot displays the QlikView AccessPoint web interface. At the top, there is a header with the QlikView logo and a user welcome message. Below the header, the main content area is titled 'AccessPoint'. It features a navigation bar with filters for 'Category' (set to 'All') and 'Attribute' (set to 'No Attributes Available'). There is also a 'View as' dropdown and a search bar. The main area shows a grid of seven document thumbnails, each with a title, a star icon, a last update timestamp, and a 'view details' link. The documents are: 'Data Visualization.qvw' (last update: 2015-08-03 22:45), 'Getting Started.qvw' (last update: 2015-07-21 22:38), 'Movies Database.qvw' (last update: 2012-04-03 03:14), 'Prescription Tracker.qvw' (last update: 2015-09-02 18:23), 'Qlik DataMarket.qvw' (last update: 2015-07-29 13:44), 'QlikView Developer Toolkit.qvw' (last update: 2013-12-02 22:26), and 'Retail Store Performance.qvw' (last update: 2015-07-20 22:52). At the bottom right, there is a pagination control showing 'Showing 1-7 of 7' and '12 items per page'.

*Thumbnails view in AccessPoint*

## 2 Planning QlikView Deployments

The screenshot shows the QlikView AccessPoint interface. At the top, there's a header with the QlikView logo and a welcome message. Below the header, there's a navigation bar with the title 'AccessPoint' and a search bar. The main content area displays a list of documents in a 'Detailed' view. The list has columns for 'Name', 'Category', and 'Last Update'. The first document is 'Data Visualization.qvw' with a category of 'Default' and a last update of '2015-08-03 22:45'. Below the list, there's a detailed view of the selected document, showing its file size (6 MB), available clients (Full Browser Version and Small Device Version), and a thumbnail of the document's content. The thumbnail shows a QlikView dashboard with various charts and tables.



### Detailed view in AccessPoint

The settings available in AccessPoint are listed below.

| Setting   | Description   |
|-----------|---|
| Category  | Category grouping for the document. Categories are managed in QMC under <b>Documents&gt;User Documents&gt;Document Information</b> .                                |
| Attribute | Attribute grouping for the document. Attributes are managed in QMC under <b>Documents&gt;User Documents&gt;Document Information</b> .                               |
| View as   | Document display type, <b>Detailed</b> view or <b>Thumbnails</b> view.<br><br>In the Detailed view, the documents can be sorted by Name, Category, and Last Update. |

Click a **view details** link in the Thumbnails view or a plus sign (+) to the left of a document name in the Detailed view to display additional information on a document (see below).

## 2 Planning QlikView Deployments

| Field/Button               | Description   |
|----------------------------|---|
| Last Update                | When the document was last updated.<br><div> <i>This is only displayed in the Thumbnails view.</i></div>                                       |
| Next Update                | When the document will be updated next time.<br><div> <i>This is only displayed if the document is part of a task that has a schema.</i></div> |
| File Size                  | Size of the document.   |
| Available Clients          | Click a client to open the document with that client.   |
| Remove last document state | Click this button to remove the last document state.  |

Click a star icon next to a document name in the Thumbnails or Detailed view to set the preferences for the document.

| Setting          | Description   |
|------------------|---|
| Open with        | Select a client to make it the default client to open the document with.  |
| Add to favorites | Click this link to add the document to the favorite documents. Select <b>Category&gt;Favorites</b> in AccessPoint to display the favorites. |

### Modifying the modal dialogs in the Ajax client

The modal dialogs, such as **Print**, **Export**, and **Server Connection Lost**, can be modified in the file *customTranslations*.

Navigate to *C:\Program Files\QlikView\Server\QlikViewClients\QlikViewAjax\htc\customFiles*. The files *customConfig* and *customTranslations* are empty, but the files *customConfigExample* and *customTranslationsExample* present examples on how to edit.

In the file *customConfig*, it is a prerequisite that *TranslationEvents* is set to *true* in order for the edits in *customTranslations* to be valid.

For the changes to take effect, the server has to be stopped and restarted.

## Directory Service Connector

### Overview

|                   |   |
|-------------------|---|
| <b>Executable</b> | %ProgramFiles%\QlikView\Directory Service Connector\QVDirectoryServiceConnector.exe |
| <b>Data</b>       | %ProgramData%\QlikTech\DirectoryServiceConnector                                    |



|                      |                        |
|----------------------|------------------------|
| <b>Listens to</b>    | HTTP: 4730; SNMP: 4731 |
| <b>Uses/Controls</b> | -                      |
| <b>Used by</b>       | QDS, QMS, QVWS         |

### Files

#### Settings and Configuration

These settings originate from QVPR.

| File                            | Description                         |
|---------------------------------|-------------------------------------|
| <i>Config.xml</i>               | Configuration file for the service. |
| <i>Resources/&lt;id&gt;.xml</i> | DSP configurations.                 |

### Logs

| File                        | Description          |
|-----------------------------|----------------------|
| <i>Log\&lt;date&gt;.txt</i> | Event and error log. |

### DSP Interface

The reason for developing a proprietary Directory Service Provider (DSP) is to have QlikView distribute documents to users in a directory service not supported by default, and to provide group resolution to the web server.

#### DirectoryServiceProvider

DirectoryServiceProvider is the interface of the class that plugs into the framework. The members of the interface are listed below.

| Member  | Description  |
|---|--|
| <code>LogMessage LogMessageEvent { set; get; }</code>                           | Directly after construction, this field is instantiated with a delegate that provides crude logging facilities.  |
| <code>string ProviderName { get; }</code>                                       | A free-form, preferably descriptive, name of the component that is suitable for the end user.  |
| <code>string ProviderType { get; }</code>                                       | An installation-unique identifier used internally by the framework and related components. The identifiers used by the supplied providers are AD, NT, Local, and Custom. |
| <code>void SetupPath (string _path, string _username, string _password);</code> | Creates a node that represents the corresponding directory service node on the specified path. Upon failure, an exception is thrown.                                     |
| <code>IList&lt;string&gt;GetKnownRootPaths ();</code>                           | The returned list should contain one or more viable paths for the methods listed here.   |
| <code>void ClearCache ();</code>  | Clears the cache (if any).   |

## 2 Planning QlikView Deployments

| Member  | Description   |
|---|---|
| <code>string DomainName { get; }</code>   | A “domain name” associated with the path that is set up. It is used as a qualifier to separate nodes from different providers (for example, the shipped Active Directory provider uses <code>NetBIOSName</code> as domain name).  |
| <code>IDictionary&lt;string, string&gt; GetSettings ();</code>  | The dictionary of supported settings has the name of the setting as key and the name of the type as value.  |
| <code>void SetSetting (string _name, string _value);</code>   | The parsing responsibility is obviously put on the provider.  |
| <code>IList&lt;IDSObject&gt; Search (string [] _pattern, eSearchType _type, string _otherattribute);</code> | Searches for nodes with attributes matching any of the patterns provided. The attributes are specified with the type parameter, which can be one or more values from the enumeration. If type is “other”, the last parameter specifies the name of the attribute. The search type “legacyid” is used for backwards compatibility. Search should support patterns containing the wildcard sign “*”, which matches zero or more characters of any kind. |
| <code>void Dispose ();</code>   | Called whenever a provider object is released.  |
| <code>IDSObject</code>  | A simple interface for any type of node within the directory service.   |
| <code>string ID { get; }</code>   | Node ID, unique within the instantiated path and consistent over all executions.  |
| <code>string DisplayName { get; }</code>  | Common name of the node in the directory service.   |
| <code>string AccountName { get; }</code>  | Account name associated with the node (if present).   |
| <code>eDSObjectType ObjectType { get; }</code>  | Basic type of the object.   |
| <code>IList&lt;IContainer&gt; MemberOf ();</code>   | A list of all groups that the node is member of.  |
| <code>string GetCustomProperty (string _name);</code>   | Any other property not natively supported by the interface. If not present, null is returned.   |
| <code>string Email { get; }</code>  | The primary e-mail address associated with the node (if any).   |

### QlikView Management Service

#### Overview

|                   |   |
|-------------------|---|
| <b>Executable</b> | <i>%ProgramFiles%\QlikView\Management Service\QVManagementService.exe</i> |
| <b>Data</b>       | <i>%ProgramData%\QlikTech\ManagementService</i>                           |
| <b>Listens to</b> | HTTP: 4780 (Web); HTTP: 4799 (API); SNMP: 4781                            |

## 2 Planning QlikView Deployments

|                      |                        |
|----------------------|------------------------|
| <b>Uses/Controls</b> | DSC, QDS, QVS, QVWS    |
| <b>Used by</b>       | Web browser/API client |

### Files

#### Settings and Configuration

QlikView Management Service (QMS) keeps a global view of the settings in QVPR.

| File              | Description                         |
|-------------------|-------------------------------------|
| <i>Config.xml</i> | Configuration file for the service. |

### Logs

| File                        | Description          |
|-----------------------------|----------------------|
| <i>Log\&lt;date&gt;.txt</i> | Event and error log. |

### SNMP

QlikView provides SNMP agents for all services.



*QlikView supports the iReasoning MIB browser for pulling data from the SNMP agents.*

The SNMP setting is off by default, since the implementation is in its initial stages and subject to change. At the time of writing, reading operations from the agents are enabled. The following messages are supported:

- GetRequest
- GetResponse
- GetNextRequest

All services answer the standard SNMP queries (see below).

| Identifier      | Query       | Description  |
|-----------------|-------------|--|
| 1.3.6.1.2.1.1.1 | sysDescr    | Description of service/product.<br><br>Example:<br><br>sysDescr.0:Qlikview Publisher Commandcenterservice version 8.50.600               |
| 1.3.6.1.2.1.1.2 | sysObjectID | Unit type.<br><br>Example:<br><br>sysobjectID.0:iso.org.dod.internet.private.enterprises.qliktech.products.publisher.Distributionservice |

## 2 Planning QlikView Deployments

| Identifier      | Query       | Description  |
|-----------------|-------------|--|
| 1.3.6.1.2.1.1.3 | sysUpTime   | System uptime.<br><br>Example:<br><br>sysUpTime.0:0 hours, 12 minutes, 15 seconds                    |
| 1.3.6.1.2.1.1.4 | sysContact  | Can be set in the configuration file.<br><br>Example:<br><br>sysContact.0:Unspecified system contact |
| 1.3.6.1.2.1.1.5 | sysName     | Can be set in the configuration file.<br><br>Example:<br><br>sysName.0:Unspecified name              |
| 1.3.6.1.2.1.1.6 | sysLocation | Can be set in the configuration file.<br><br>Example:<br><br>sysLocation.0:Unspecified location      |
| 1.3.6.1.2.1.1.7 | sysService  | Constant, 72 means application server.<br><br>Example:<br><br>sysServices.0:72                       |

The QlikView Distribution Service can answer additional queries. These are specified in the MIB file.

Each service has a configuration file, which is stored in the subfolder for the service in the installation folder. For example, the configuration file for the QlikView Distribution Service is

*QlikViewdistributionService.exe.config*.

The SNMP settings can be adjusted in the `SNMP SETTINGS` part of the configuration file. SNMP has to be enabled for all services (the default is off).

| Setting        | Description   |
|----------------|---|
| EnableSNMP     | Enables the SNMP listener. The default value is <code>false</code> .  |
| SNMPPort       | Sets the port to use for the particular Publisher service. See the default settings for each service below.   |
| SNMPsysContact | Contact information for the person responsible for the managed node. The default value is <code>unspecified system contact</code> .   |
| SNMPsysName    | An administratively assigned name for the managed node. By convention, this is the fully qualified domain name of the node. If the name is unknown, the value is a zero-length string. If left empty, it defaults to the current machine name. The default value is <code>unspecified name</code> . |

## 2 Planning QlikView Deployments

| Setting         | Description  |
|-----------------|--|
| SNMPsysLocation | Physical location of the node (for example, “telephone closet, third floor”). The default value is unspecified location. |
| DebugSNMP       | Enables the extended debug log for the SNMP listener. The default value is false.  |

The default port settings for the services are listed below.

| Service                     | Default Port Setting     |
|-----------------------------|--------------------------|
| Management Service          | 4781                     |
| Directory Service Connector | 4731                     |
| Distribution Service        | 4721 (default SNMP port) |
| QlikView Server             | 161                      |
| QlikView Web Server         | 4751                     |

All ports can be configured. If the services are installed on different machines, they can all run on the same port. The ports change as the implementation moves away from the experimental SNMP range and into the range allotted by Qlik.

### MIB File

A MIB file is included in the QlikView delivery, so that all SNMP managers can interpret the additional responses from the QlikView Distribution Service. Note, however, that the MIB file is subject to change. The file is installed in *\\QlikView\Support Tools*. The support tools require a customized installation.

The QlikView Distribution Service can answer the queries listed below, in addition to the ones previously mentioned.

| Identifier                    | Query   |
|-------------------------------|---|
| 1.3.6.1.4.1.30764.1.2.2.1     | QDSTaskExecuteStatusTable   |
| 1.3.6.1.4.1.30764.1.2.2.1.1   | QDSTaskExecuteStatusEntry   |
| 1.3.6.1.4.1.30764.1.2.2.1.1.1 | QDSTaskID (task ID number)  |
| 1.3.6.1.4.1.30764.1.2.2.1.1.2 | QDSTaskName (task name)   |
| 1.3.6.1.4.1.30764.1.2.2.1.1.3 | QDSTaskExecuteStatus (task status): <ul style="list-style-type: none"><li>• Waiting</li><li>• Running</li><li>• Aborting</li><li>• Failed</li><li>• Warning</li></ul> |
| 1.3.6.1.4.1.30764.1.2.2.1.1.4 | QDSTaskNextExecutionAt (when the task will be executed next)  |

| Identifier                    | Query   |
|-------------------------------|---|
| 1.3.6.1.4.1.30764.1.2.2.1.1.5 | QDSTaskLastExecutedAt (when the task was executed last) |
| 1.3.6.1.4.1.30764.1.2.2.1.1.6 | QDSTaskCurrentWork (what the task is currently doing)   |
| 1.3.6.1.4.1.30764.1.2.2.1.1.7 |   |

---

**See also:**

-  <http://www.ietf.org/rfc/rfc1157.txt>
-  [http://en.wikipedia.org/wiki/Simple\\_Network\\_Management\\_Protocol](http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol)

## 2.2 Deployment

The QlikView architecture is based on the concept of sites. A QlikView site is a collection of one or more nodes (that is, server machines) connected to a common logical repository or central node.

QlikView can be deployed in many ways. This section describes different deployment scenarios.

### Building a Farm

Server farms can be used to provide additional performance, redundancy, and security in place of a single server solution.

### Planning

Before starting the actual installation, planning is needed. The following items have to be considered:

- Trust mechanism
- Web server (QlikView Web Server or Microsoft IIS)
- Redundancy level
- Account to run the services under
- QVPR format (XML or SQL)
- User directory
- User authentication
- Firewalls

### Trust Mechanism

Trust mechanisms are provided with Windows groups or certificates.

Windows groups can easily be deployed, if all services reside in a single Active Directory (AD). If encrypted communication is needed, it can be added manually.

Certificates provide for trust mechanisms in cross-domain environments and can also provide SSL encryption.

### Web Server

QlikView Web Server is intended for use when the web server is not needed for other purposes. It is lightweight and easy to manage, but at the same time limited to support the tasks needed by a QlikView installation.

A Microsoft IIS-hosted web server is recommended, if:

- More flexibility or more advanced tuning is required
- The web server is to be used for other tasks than QlikView
- An authorization scheme not available out-of-the-box is required

### Redundancy Level

The redundancy level is mainly a question of clustering and/or having multiple machines running the same service. All services except QlikView Management Service (QMS) can be installed on multiple machines. In addition, QlikView Server (QVS), QlikView Distribution Service (QDS), and Directory Service Connector (DSC) can be clustered.

### Account to Run the Services Under

A dedicated account should be created to manage the QlikView services. The account should be assigned with proper privileges during the installation.

It is recommended that the same account is used for all services.

### QVPR Format

The choice of QVPR format is based on reasons outside the QlikView product (for example, backup and availability). The installation always starts in XML mode.

### User Directory

QlikView defaults to Windows users (that is, NTFS mode). If non-Windows users are to be given access (other than anonymously), QlikView Server must run in Document Metadata Service (DMS) mode.

### User Authentication

QlikView supports multiple authentication schemes. Additional schemes may require ASPX development and the possible use of Microsoft IIS for web services.

### Firewalls

Make sure that the services are able to communicate (for example, by opening the appropriate ports in the firewalls).

See: *Service by Service (page 19)*

### Root/First Install

Before starting, make sure that the appropriate service account (or accounts) is set up and available on the machines where the services are to be installed.

In all installations, there must exist exactly one QMS, which must be installed first. Note that the QMS must be able to communicate with all the subsequently installed services.

If more services are to run on the same server, they can be installed at the same time.

### Adding Services on Other Machines

The next step is to install the other services on the other servers. If more services are to run on the same server, they can be installed at the same time. The order in which the services are added is not important.

When the services have been installed, it is time to return to QlikView Management Console (QMC) and configure the services. This is done on the System tab. The first step is to add the services. Make sure to note the differences between building out a cluster and creating a brand new cluster.

### Clustering

This section provides an overview of how create a QlikView Server cluster.

#### QlikView Server

For the QlikView Server cluster to work properly, it is important to set **System>Setup>QVS resource>Folders>Root Folder** to a common shared folder. In addition, **Alternate Temporary Files Folder Path** must be set to a common shared folder (separate from the root folder).

If extensions are used, it simplifies management if **Alternate Extension Path** is set to a common shared folder.

It is also common practice to set **System>Setup>QVS resource>Logging>Log Folder** to a common place, but this is not strictly necessary.



*The root folder must not be used for anything else than cluster files (that is, .pgo files) and user documents.*

#### QlikView Distribution Service

For a cluster of QDSs, **System>Setup>General>Application Data Folder** must be set to a common shared folder. In addition, **Source Folders** must be common shared folders.

#### Directory Service Connector

A cluster of DSCs does not need any specific settings. The difference between clustered and non-clustered DSCs is whether the settings are shared or not.

#### QlikView Web Server

Multiple web servers can be set up, but they are always configured independently (that is, they are never clustered). Note that it is uncommon, but from a technical perspective possible, to have some web servers running QlikView Web Server (QVWS) and some Microsoft IIS.

#### Tunneling Using Microsoft IIS

Tunneling is used by Windows native clients (QlikView Desktop, the OEM OCX, and the Internet Explorer plugin) and needed when the clients cannot communicate with QlikView Server on port 4747 (most likely due to a firewall blocking the traffic):



- QVWS: No extra settings are required.
- Microsoft IIS: The *QVSTunnel.dll* file must be added as an ISAPI filter.

Proceed as follows to set up tunneling for Microsoft IIS 7:

1. Open the Internet Information Services Manager.
2. Select the IIS top node.
3. Open the ISAPI and CGI Restrictions dialog.
4. Select **Add** in the Actions pane and browse to the location of *QVSTunnel.dll*.
5. Provide a description of the instance and check the **Allow extension path to execute** box.
6. Open the site that is to host the QlikView Server and Publisher pages and click **Scripts**.
7. Open the Handler Mappings dialog.
8. Locate ISAPI dll and select **Edit Features Permission** in the Actions pane.
9. Click **Execute** in the dialog that opens.

The following entries are required in the registry when the QVS and Microsoft IIS are located on different machines:

[HKEY\_LOCAL\_MACHINE\SOFTWARE\QlikTech\QlikTunnel]

- "QVSPort"=dword:000012a6
- "QVSServer"="QvsHost"



*If the entries do not already exist in the registry, they have to be added manually.*

Test the QlikView Server tunnel by entering the following URL in a client browser window:

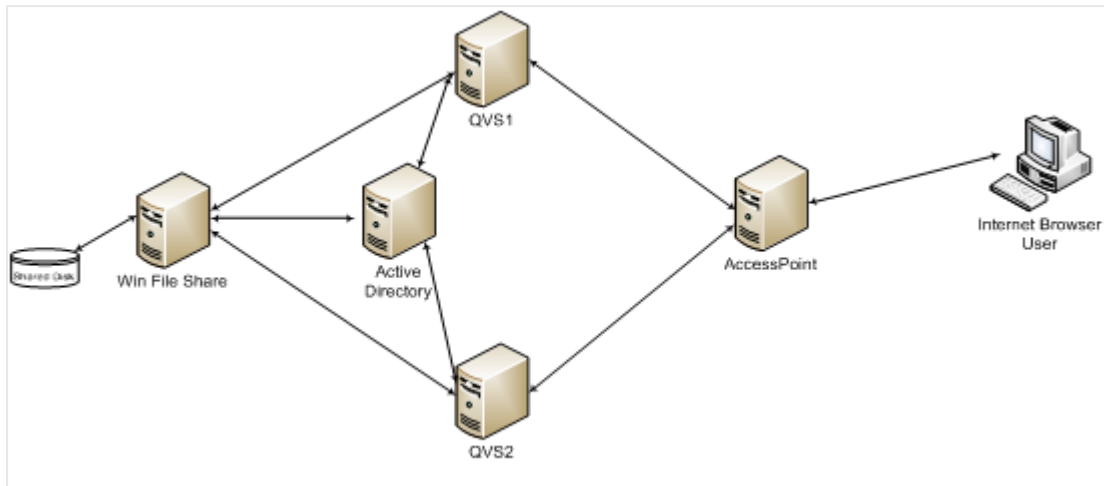
*http://<Servername>/scripts/qvstunnel.dll?test*

*Servername* is the web server. If the tunnel is correctly set up, the web page returns a message (that tunneling is available) and the QlikView Server version number.

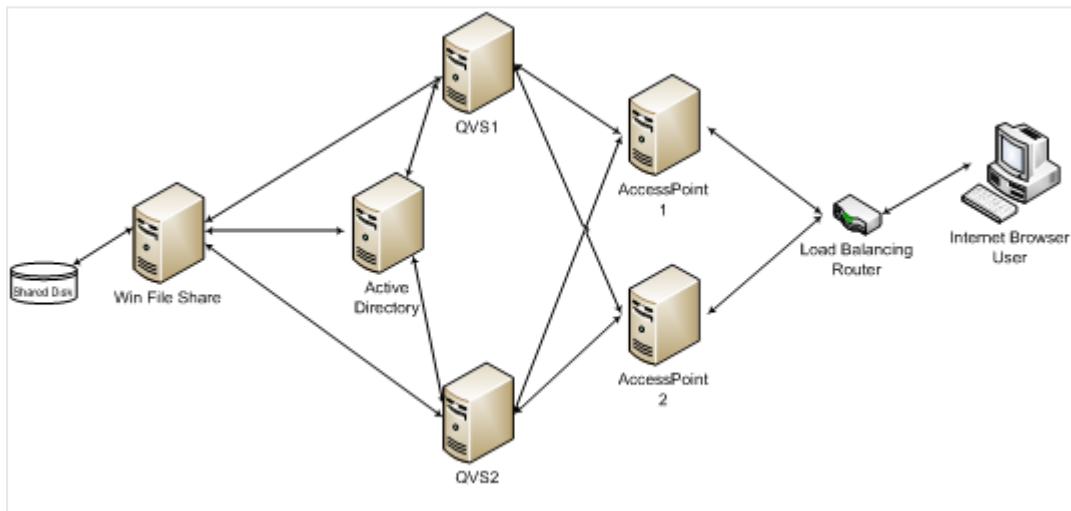
## Clustering QlikView Servers

This chapter discusses the architectural and installation requirements and options for building a clustered and resilient QlikView Server deployment.

The following figure shows a clustered, load balanced QlikView Server deployment.



The following figure shows a resilient, clustered, load balanced QlikView Server deployment that uses AccessPoint load balancing.



The QlikView Server load balancing capabilities are included in the QlikView web portal, AccessPoint. This chapter also discusses how to make this component resilient using network load balancing (if needed).

### Why Cluster QlikView Servers?

By clustering QlikView Servers, the objectives described below can be achieved.

#### Horizontal User Scalability

If more resources than can be provided by a single QlikView Server are needed, an additional server can be added. For example, if the server can support 100 concurrent users, but 200 concurrent users have to be supported, an additional server can be added. In this scenario, the first 100 users could be allocated to server A and the second 100 users to server B. Alternatively, the servers could be clustered so that you set resilience.

### Resilience

When the number of users increases, so does the users' reliance on QlikView. By clustering the QlikView Servers, resilience can be built into the deployment. In the case above, where a single server can support 100 users, three servers could be used to build resilience into the deployment. This would allow one server to be lost (for example, because of hardware failure) with the system still capable of supporting 200 users. Having all three servers as active nodes helps reduce the response times by not running all servers at 100% of their capacity. This also limits the number of users affected if a node is lost.

QlikView does not provide any session recovery. In practice, this means that if a node in the QlikView cluster is lost, the users lose the analysis they are currently performing. They will have to reconnect to the cluster to resume their work. This does not mean that the data within the QlikView application is lost and needs to be reloaded, because the data is stored in the .qvw file on the file system. Only the selections made in the application are lost.

### Load balancing

A QlikView deployment uses a load-balancing algorithm to take advantage of the full capacity of all the machines in a QVS clusters. The web server running the AccessPoint determines which QVS to use. There are three options for how to load balance your QVS. See: *QVS Load Balancing Options (page 52)*.

### Requirements for Clustered QlikView Deployment

There are three high-level requirements for building a clustered QlikView deployment:

1. Clustered QlikView Server license key
2. Shared storage area for Root folder
3. Same build number

#### Clustered QlikView Server License Key

In a clustered environment, the QlikView Server machines are installed with the same license key, which must be enabled for clustering. This can be checked confirmed by examining the following entry in the License Enabler File (LEF):

```
NUMBER_OF_CLUSTER_NODES; 2 (number of nodes in the cluster)
```

Clustered QlikView Servers share configuration and license information between themselves via the shared storage, so that configuration and license management only needs to be performed once from the QlikView Management Console (QMC) for all nodes.

The servers must be installed on the same network subnet and have a shared root document directory; hence the requirement for a shared network storage. The configuration information is stored in Persistent Global Objects (.pgo) files.

If the servers fail to start or reset after ten minutes, check for the LEF entry above. This is usually an indication that QlikView Server is installed on more machines than allowed.

### Shared Network Storage

Shared network storage is required not only for the .pgo files mentioned above, but also for storage of QlikView applications that are required in the cluster. This also enables collaborative objects to be shared across the nodes in the cluster (using shared files).

QlikView requires the storage of documents (.qvw files), .pgo, .meta, and shared files (.Shared or .TShared) to be hosted on a Windows-based file share. Hosting files on any other type of system is unsupported and may create an unstable QVS cluster where CALs disappear and QVSs stall. QlikView supports the use of a SAN shared from a QlikView Server.



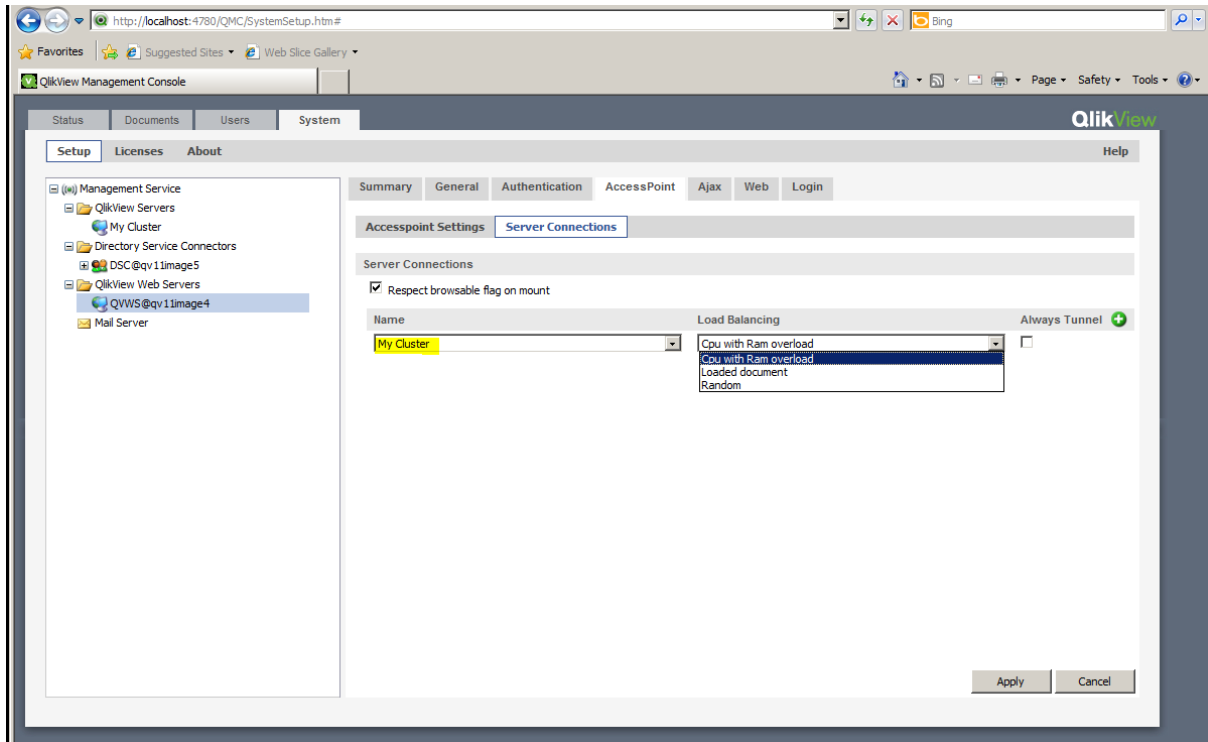
*QlikView does not support Windows Distributed File System (DFS).*

### QVS Load Balancing Options

QVS supports three load balancing strategies:

- Random (default setting): A round robin type strategy ideal for most users, since the session is distributed across all nodes in the cluster.
- Loaded document: Used when sessions for the same document are to be routed to the same server. This strategy is designed for deployments where there are more documents than a single node in the cluster can handle. AccessPoint makes the decision based on if the document is already loaded and on the amount of RAM available on the server.
- CPU with RAM overload allows QlikView Web Server (QVWS) to route traffic based on two factors, (1) RAM and (2) CPU use. The node is chosen using the following criteria:
  - If RAM is readily available (low) on all available nodes, choose the node with the lowest CPU use.
  - If RAM is moderately used on all available nodes, choose the node with the most RAM available.

The QVS load balancing strategy can be set in the QMC under **System>Setup>QlikView Web Servers**. Select the web server on the **AccessPoint** tab:



### Load Balancing the Web Server

The network load balancer provides the resilience for AccessPoint, routing the sessions to an available AccessPoint server. This is done by third-party software and hardware.

There are several requirements on the load balancer:

- Support for session persistence / sticky sessions: This ensures a user's session persists on the same node within the cluster, usually by using a cookie.
- Availability: The load balancer checks the availability of the AccessPoint web server and the QlikView servers.
- Some form of load balancing algorithm to determine which server is the least loaded.

### Session Persistence

The requirement is for the user's session to be routed consistently to the same server. Methods for doing this vary from device to device – refer to the load balancer documentation for information on the options available.

### Availability Checking

A special web page on the AccessPoint provides automated checking of the system status:

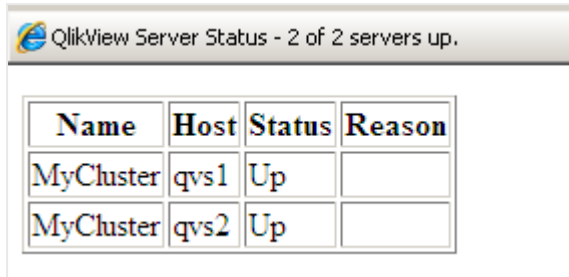
*<http://myAccessPoint/QvAjaxZfc/QvsStatus.aspx>*

This page returns an http status code of 200, if the AccessPoint and at least one QlikView Server in the cluster respond. Any other status code returned by this page should be considered an error. Common errors from this page include:

## 2 Planning QlikView Deployments

- 404: The AccessPoint is unable to respond. Check the web server.
- 503: No QlikView Servers responded to the AccessPoint and therefore it cannot service user requests.

The status of the QlikView Server cluster is also displayed on the web page:

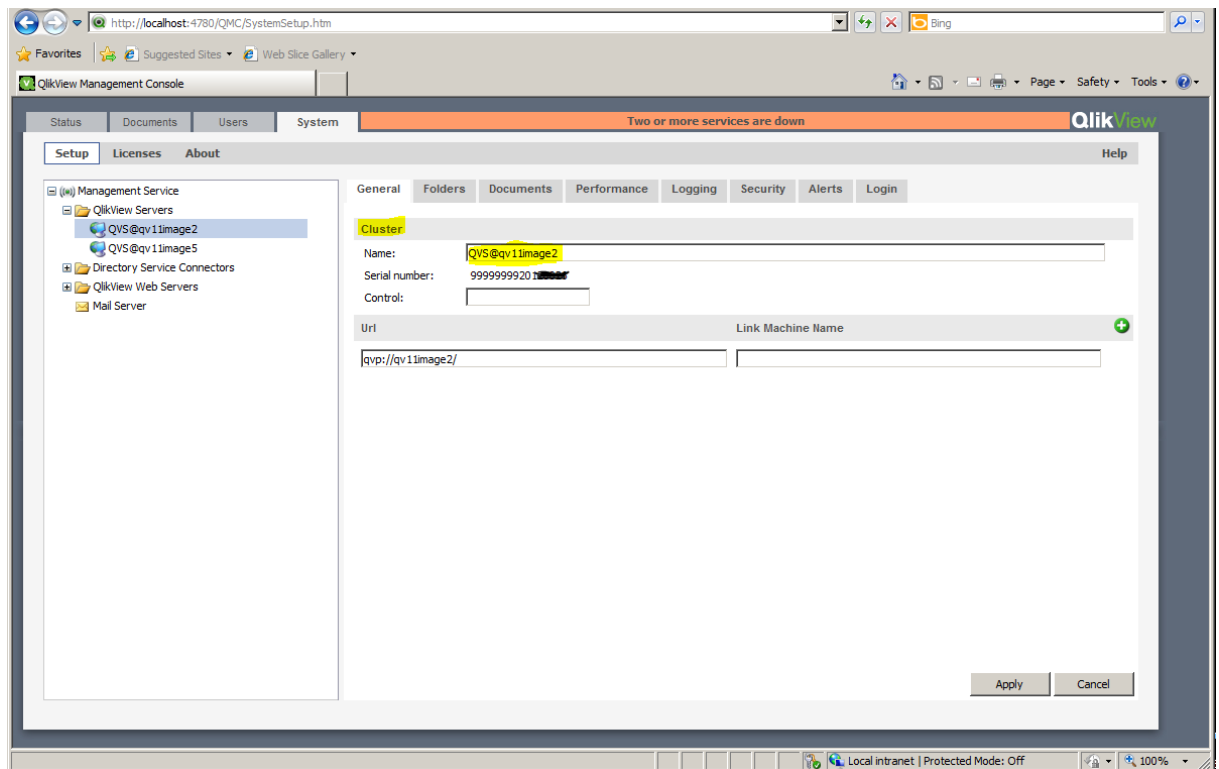


| Name      | Host | Status | Reason |
|-----------|------|--------|--------|
| MyCluster | qvs1 | Up     |        |
| MyCluster | qvs2 | Up     |        |

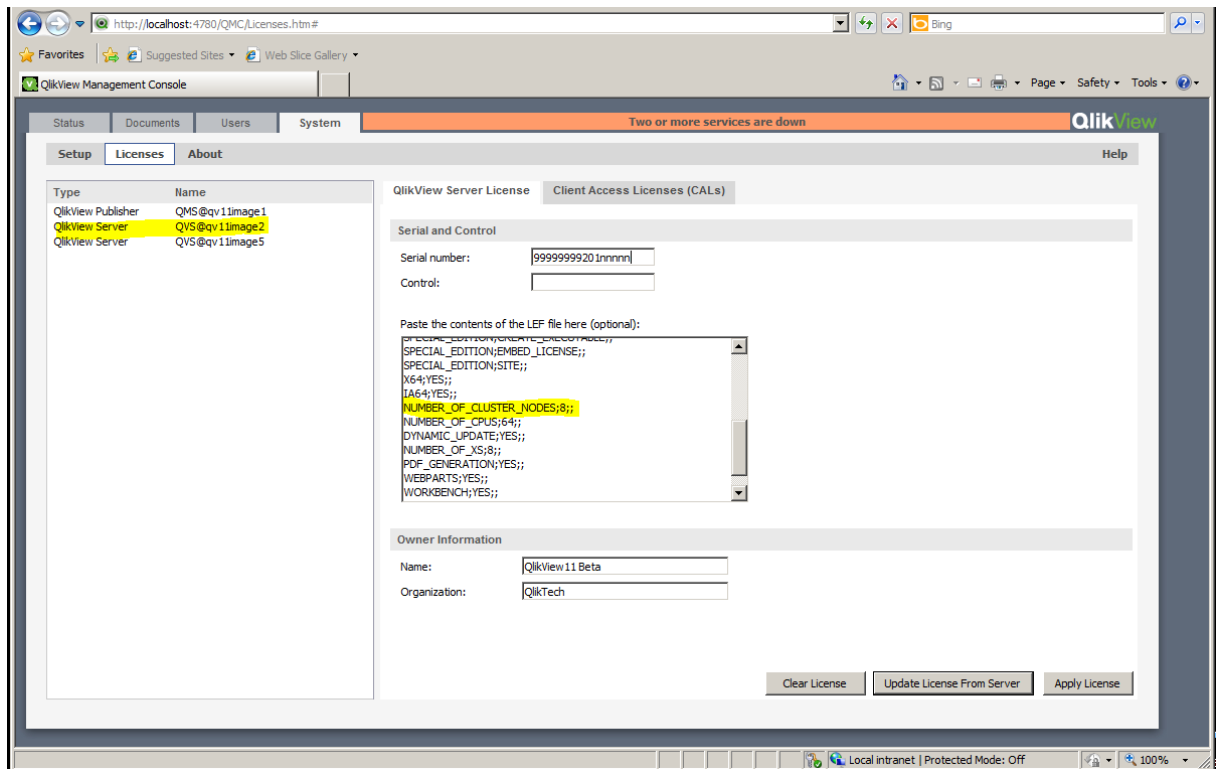
### Building and Installing a QlikView Cluster

Proceed as follows to configure and activate a QlikView Server cluster using the QMC:

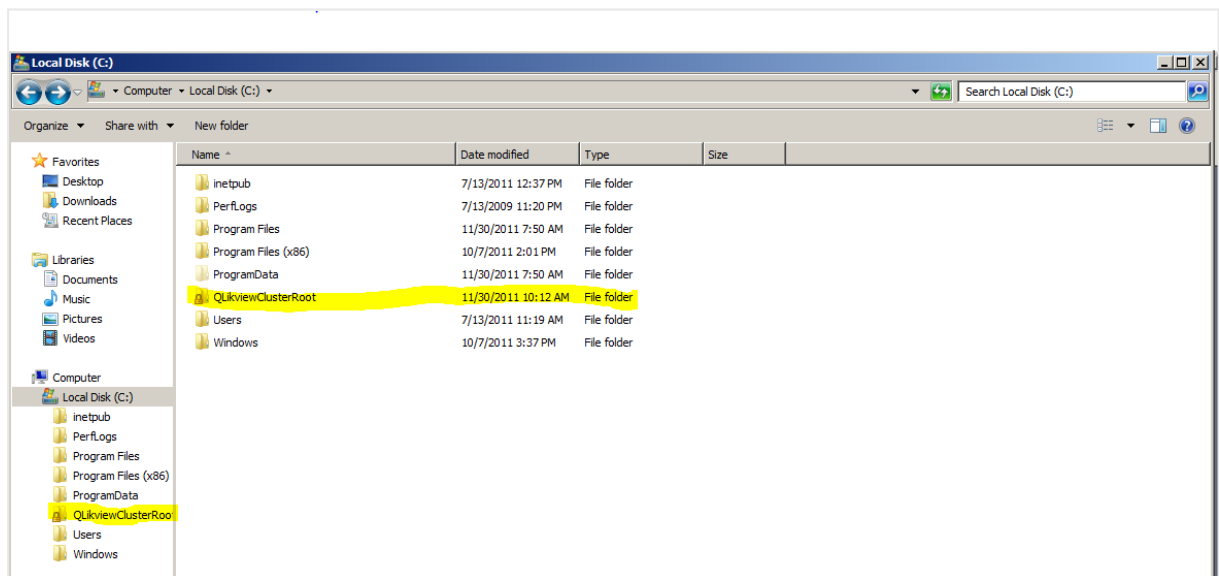
1. Install and license the first QlikView Server in the cluster. This will be the first copy of QlikView Server.



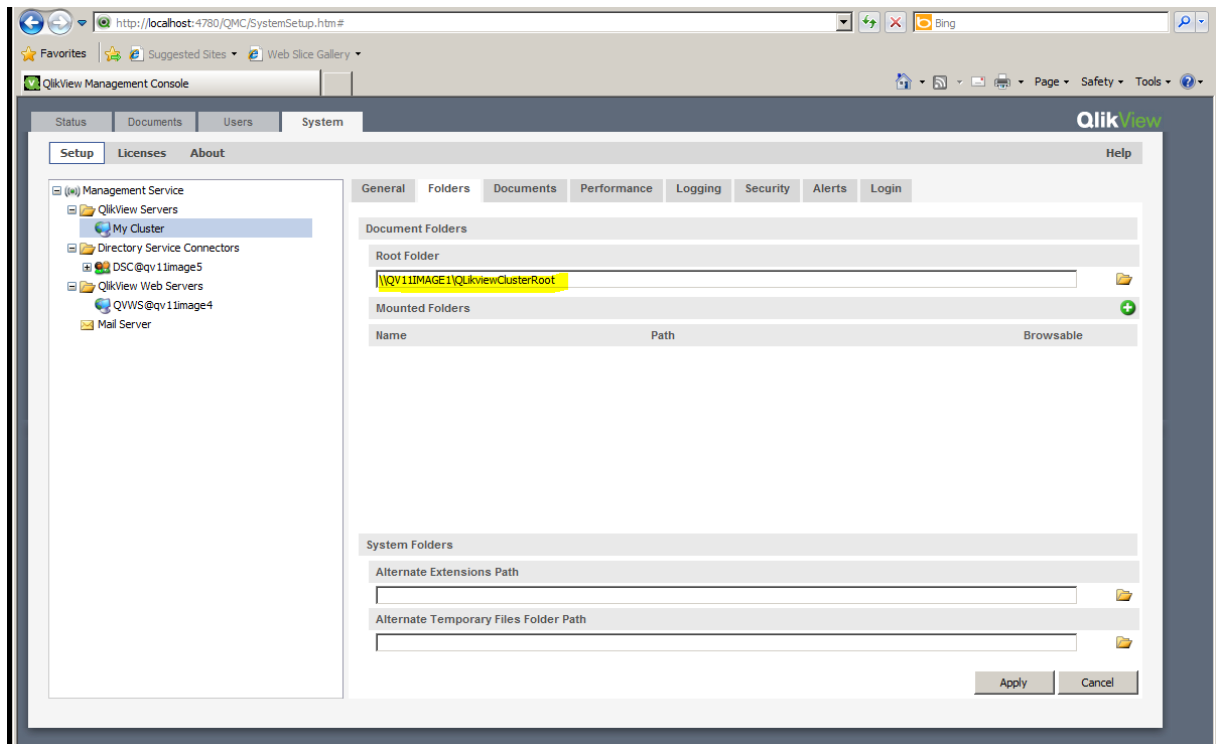
## 2 Planning QlikView Deployments



2. Configure the document folder to point to a folder on the file system that all QlikView Servers in the cluster can access.



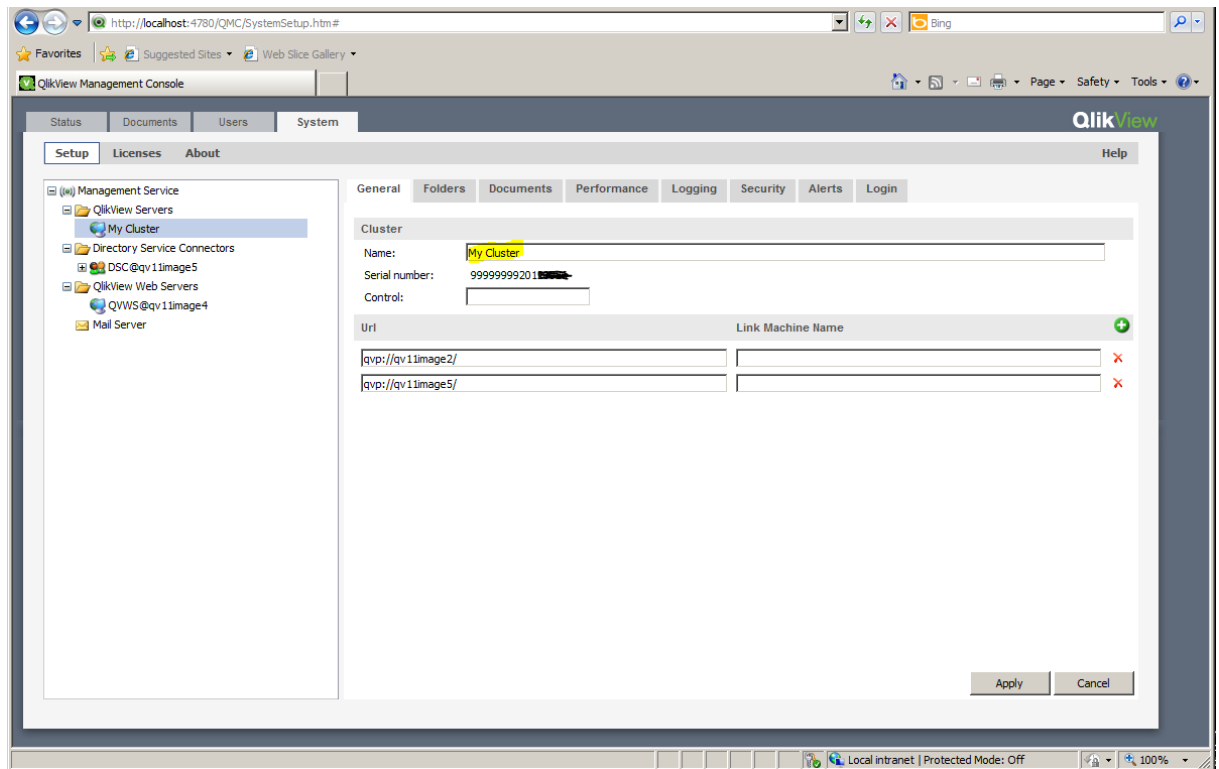
## 2 Planning QlikView Deployments



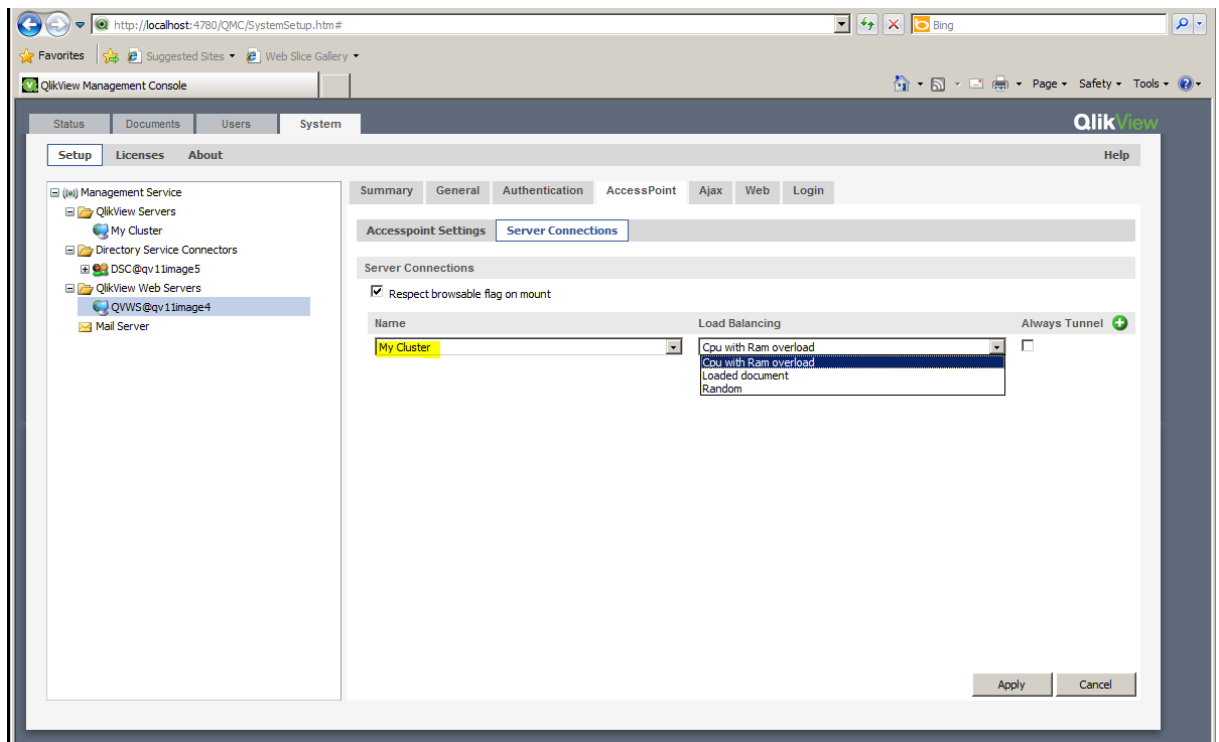
3. Install the next QlikView Server in the cluster.
4. Ensure that all QlikView services are running as local administrators and that they are members of the "QlikView Administrators" local group.
5. Open **System>Setup** in the QMC and select the server. Then go to the **General** tab and enter the control number for your license and the address to the second QlikView Server in the cluster.
6. Rename your cluster to an appropriate name.
7. Repeat steps 3 - 5 for the QlikView Server nodes in the cluster.



## 2 Planning QlikView Deployments



8. Make sure that the cluster is selected in **Server Connections** in the settings for the AccessPoint.



9. The cluster is now configured and ready to use.

### Unbalanced QVS Clustering

By default, a QVS cluster requires that all nodes are equal regarding CPU, cores, and RAM. You might want to use nodes with different hardware specifications, either because of difficulties finding identical machines or a need to handle documents of different sizes.

QlikView Server Unbalanced clustering is enabled with a new setting *ActiveUnbalancedCluster*. This will remove possibilities to set CPU affinity and RAM settings in the QMC. All settings for the individual nodes must now be done in the local node QVS *Settings.ini* files.

Only full affinity is supported when using this feature (100% of cores).

The new load balancing algorithm should be activated when using an unbalanced cluster. This can be configured to specific needs.

Do the following to configure the feature:

1. Set the *ActiveUnbalancedCluster* configuration parameter to *True* in the QMS settings file. By default, the file is located in *C:\Program Files\QlikView\Management Service\*.
2. Set the *UnbalancedClusterLoadBalancer* configuration parameter to *True* in the QVWS settings file. By default, this file is located in *C:\Program Files\QlikView\Server\Web Server\*.
3. In the QMC, navigate to System > Setup > QlikView Web Servers > AccessPoint > Server Connections and select the **CPU with RAM Overload** from the **Load Balancing** field.  
If there is a need to customize the weights of the algorithm (QVWS settings file):
4. Set the *UnbalancedClusterLoadBalancerCpuWeight* to a value between 0 and 10. A higher value indicates the processing power should be given more weight when the load-balancing algorithm determines which QVS cluster is used to open documents.
5. Set the *UnbalancedClusterLoadBalancerRamWeight* to a value between 0 and 10. A higher value indicates the RAM performance should be given more weight when the load-balancing algorithm determines which QVS cluster is used to open documents.
6. Set the *UnbalancedClusterLoadBalancerLoadedDocWeight* to a value between 0 and 10. A higher value indicates the number of previous loaded documents on a QVS cluster should be given more weight when the load-balancing algorithm determines which QVS cluster is used to open documents. Make sure that no CPU affinity settings are present in the local node QVS *Settings.ini* files.  
Remove the following if present:  
*MaxCoreMask*  
*MaxCoreMaskHi*  
*MaxCoreMaskGrp1*  
*MaxCoreMaskGrp1Hi*  
*MaxCoreMaskGrp2*  
*MaxCoreMaskGrp2Hi*  
*MaxCoreMaskGrp3*  
*MaxCoreMaskGrp3Hi*
7. Working set limit Low and High will by default be set to 70 resp 90 (usage of RAM in percent). Remove old settings if necessary or change to customized levels in the local node QVS *Settings.ini* files:  
*WorkingSetSizeLoPct=nn*

WorkingSetSizeHiPct=nn

8. Restart all systems involved.

### Clustering QlikView Publisher

This chapter provides an overview of QlikView Publisher and how to use it in a clustered deployment for scalability, resilience, or both. This chapter also addresses the architectural and installation requirements and the options for building a clustered and resilient QlikView Publisher deployment.

#### Introduction

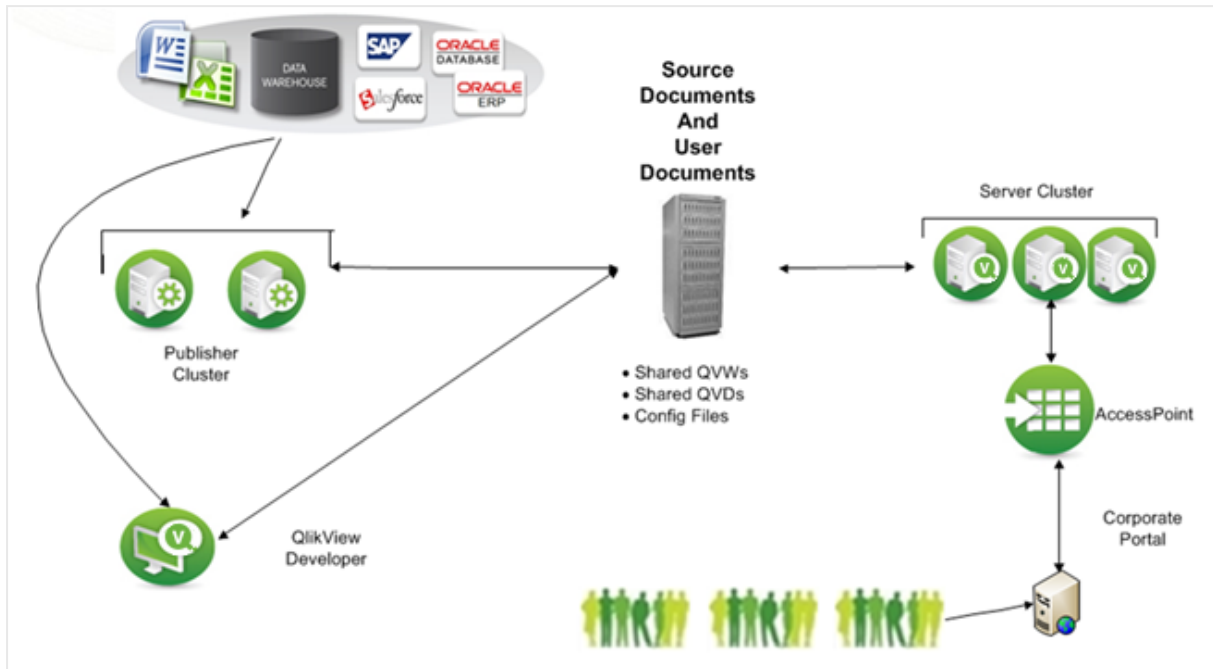
QlikView Publisher is an optional module for QlikView Server that enables scheduling, administration, and management tools that provide a single point of control for QlikView analytics applications and reports. Administrators can schedule, distribute, and manage security and access for QlikView applications and reports across the enterprise.

QlikView Publisher performs the following main functions:

- It loads data directly from data sources defined in connection strings in the source .qvw files.
- It is used as a distribution service to “reduce” data and applications from source .qvw files based on various rules (for example, user authorization or data access) and distribute these newly-created documents to the appropriate QlikView Servers or as static reports via email.
- When using QlikView Publisher, only Publisher has access to the source documents folder and the data sources for data load and distribution. The source documents and data are not accessible by QlikView users.

By deploying a clustered architecture, QlikView Publisher achieves scalability and/or resilience using web services technology. Administrators can cluster services together to provide load balancing. Native support for SNMP enables integration with enterprise system monitoring tools. External enterprise scheduling tools can trigger Publisher tasks using web service calls. Tasks can also be scheduled and executed on demand by QlikView administrators.

The figure below shows a two-server, clustered QlikView Publisher where each server is configured for processing different tasks and load balancing. The figure also includes a three-server, clustered QlikView Server that uses QlikView AccessPoint for load balancing. Documents created by QlikView Developer are stored in the source documents folder. QlikView Publisher tasks are used to retrieve data and store the result in the user documents folder.



To see how to set up an unbalanced distribution service cluster, see *Clustering QlikView Distribution Service* (page 70)

### Source Documents

The source documents contain a) scripts within .qvw files to extract data from various data sources (for example, data warehouses, Microsoft Excel files, SAP, and Salesforce.com), b) the actual binary data extracts themselves within .qvd files, or c) a binary load from another .qvw file, inheriting its data model in one line of code.

The QlikView source documents, created using QlikView Developer, reside in the following folder:

- Windows Server 2008 and later: `\ProgramData\QlikTech\SourceDocuments`. This is the default QlikView location for Windows Server 2008 and later.

### User Documents

The user documents folder is the repository used by QlikView Server. The folder is located at:

- Windows Server 2008 and later: `\ProgramData\QlikTech\Documents`. This is the default QlikView location for Windows Server 2008 and later.

### Tasks

Tasks are created by administrators for data distribution and data reloads. Tasks are stored in the QlikView Publisher repository as a collection of XML files or in an SQL Server database. When a task is executed, QlikView Publisher invokes QlikView Batch (QVB), which is comparable to QlikView Desktop without the user interface.



*QlikView Batch (QVB) does not support graphical or user input objects. This means that QVB cannot reload documents that, for example, contain scripts that require user input.*

QVB reloads the documents, which are stored in the source documents folder(s) and creates an associative QlikView database, which is stored within each document. The QVB performs the reload by retrieving the data described by the load script from the data sources. QlikView Publisher distributes the documents to the user documents folder for QlikView Server using the encrypted QVP protocol, to a mail server, and/or a file folder. QlikView Publisher can use the Directory Service Connector (DSC) to determine where and to whom the documents are to be distributed.

### Why Cluster QlikView Publisher?

The role of Publisher in the QlikView solution is to distribute and refresh data by criteria set by the QlikView administrator. To accomplish this, Publisher executes many tasks, either scheduled or on demand. A Publisher task is the smallest entity that can be distributed in a cluster; a single task cannot be divided and executed in parallel on multiple cluster nodes. Clustering the Publisher service on more than one server enables the administrator to distribute multiple tasks to multiple servers operating in parallel using the Publisher load balancing algorithm. This means Publisher clusters can be used to increase the scalability, availability, and serviceability of data distribution and reloading.

In addition, a Publisher cluster license enables the configuration of Publisher services in clusters and standalone Publisher services. For example, a Publisher cluster can be used in a corporate office to handle large volumes of data and tasks, whereas a single Publisher service can be used in an associated manufacturing plant where the Publisher only needs to distribute documents using the manufacturing data source.

By clustering QlikView Publisher, the following objectives can be met:

- Horizontal scalability
- Resilience

### Horizontal Scalability

Horizontal scaling of hardware provides the ability to increase the resources of the QlikView deployment. By adding additional hardware servers, the workload of QlikView Publisher can be increased. The clustered Publisher servers can then be configured to load balance the QlikView tasks.

For example, on a certain hardware server, QlikView Publisher can process eight concurrent tasks. When the resource needs increase, the QlikView Publisher service can grow as needed. By adding an additional QlikView Publisher service on a new hardware server, the deployment can handle up to sixteen concurrent tasks by configuring the additional server in a Publisher cluster deployment. In this scenario, the first eight tasks are allocated to Server A and the second eight tasks to Server B. Alternatively, if the servers are clustered, the tasks can be load balanced over the two servers.

### Resilience

When the number of tasks in the deployment increases, the window for completing the tasks in time becomes increasingly important. Clustering the QlikView distribution services provides for resilience in the deployment. In the case above, where a single server can support 100 concurrent tasks, an additional server can be

deployed (for a total of three servers) in order to build resilience into the deployment. If a server is lost (for example, due to a hardware failure or network connection issues), the resilient cluster still supports up to 200 tasks. Having all three servers as active nodes helps reduce response times by not running all servers at 100% of their capacity. It also limits the number of tasks and task chains affected if a node is lost.

### Requirements for a Clustered QlikView Publisher Deployment

The following high-level requirements must be fulfilled for a clustered QlikView Publisher deployment:

- Clustered QlikView Publisher license key
- Shared network storage
- Load balancing strategies

#### Clustered QlikView Publisher License Key

In a clustered environment, the QlikView Publisher servers are installed with the same license key. This can be verified by examining the following entry in the License Enabler File (LEF):

```
PRODUCTLEVEL;30;; (where 30 is the code for QlikView Publisher)
```

```
NUMBER_OF_XS;N;; (where N is the number of allowed QlikView Distribution Services)
```

The servers in a clustered QlikView Publisher deployment share configuration and license information among themselves via the shared storage, so configuration and license management only needs to be performed once in the QMC for all nodes.

#### Shared Network Storage

Shared network storage is required for storage of QlikView applications that are needed in the cluster. It is recommended to host the storage of documents (.qvw files) and .meta data on a Windows-based file share. QlikView Publisher supports a SAN (NetApp, EMC, etc.) that is mounted to a Windows Server 2008 (or later) and then shared from that server. Storage presented to a server via a SAN must appear as locally attached storage. If SAN storage is used for Publisher, any distributed data that is accessed by QlikView Server should not reside on the SAN storage.



*QlikView does not support Windows Distributed File System (DFS).*

The QlikView Distribution Services (QDSs) must have a shared application data directory and possibly a shared source document directory as well (hence the requirement for a shared network storage). All configured Publisher services must have reliable network access to the shared storage.

#### Load Balancing Strategies

##### Load Balancing

The load balancing is determined by an internal ranking system based on the amount of memory available and the CPU use. Qlik recommends using the default settings, since they have been extensively tested.

To change the default settings, edit the configuration file, *QlikViewDistributionService.exe.config*. The key is written in JavaScript:

```
<add key="LoadBalancingFormule" value="(AverageCPULoad*400) + ((MemoryUsage / TotalMemory) * 300) + ((NumberOfQlikViewEngines / MaxQlikViewEngines)*200) + (NumberOfRunningTasks*100)"/>
```

where:

- `AverageCPULoad`: Average CPU load for all running QVBs.
- `MemoryUsage`: Total memory use for the entire application.
- `TotalMemory`: Total amount of memory on the server.
- `NumberOfQlikViewEngines`: Number of QlikView engines currently used.
- `MaxQlikViewEngines`: Configured value for the maximum number of QlikView engines.
- `NumberOfRunningTasks`: Number of tasks currently running.

### Simultaneous Tasks

By default, four QlikView tasks can execute simultaneously on a node. The recommended maximum is eight simultaneous tasks per node. If more than ten tasks have to be executed simultaneously on a node, modifications are necessary in the Windows registry to change the desktop heap size to allow for more simultaneous tasks.

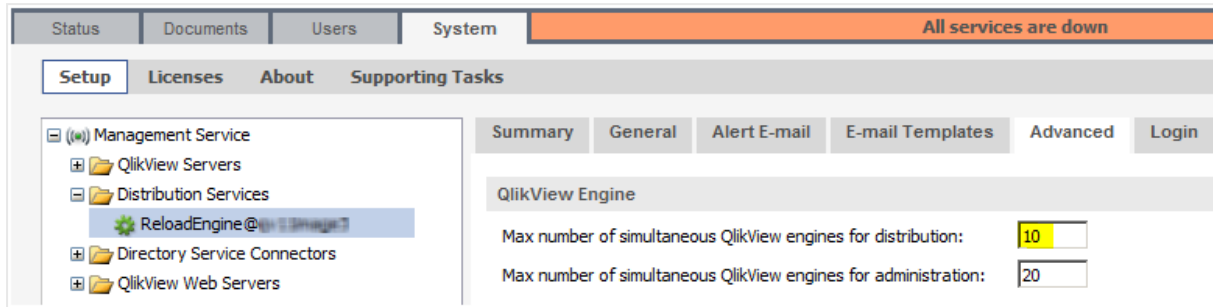


*A large-scale server is required for executing ten or more simultaneous tasks. Alternatively, add additional servers for Publisher tasks.*

Proceed as follows to change the number of tasks allowed to execute simultaneously:

1. Backup the Windows Server registry.
2. Locate the following Windows Server registry setting:  
`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session\Manager\SubSystems\windows`  
`%SystemRoot%\system32\csrss.exe ObjectDirectory=\windows`  
`SharedSection=1024,3072,512 windows=On SubSystemType=windows`  
`ServerDll=baserv,1 ServerDll=winsrv:UserServerDllInitialization,3`  
`ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=off`  
`MaxRequestThreads=16`  
The default value for `sharedsection` is 1024,20480,768 for 64-bit (x64).
3. Change the desktop heap size by setting `sharedsection` to 1024,20480,2048:  
`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session\Manager\SubSystems\windows`  
`%SystemRoot%\system32\csrss.exe ObjectDirectory=\windows`  
`SharedSection=1024,20480,2048 windows=On SubSystemType=windows`  
`ServerDll=baserv,1 ServerDll=winsrv:UserServerDllInitialization,3`  
`ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=off`  
`MaxRequestThreads=16`
4. Save the registry changes and restart the machine.
5. Change the **Max number of simultaneous QlikView engines for distribution** setting

in QMC to the number of engines needed.



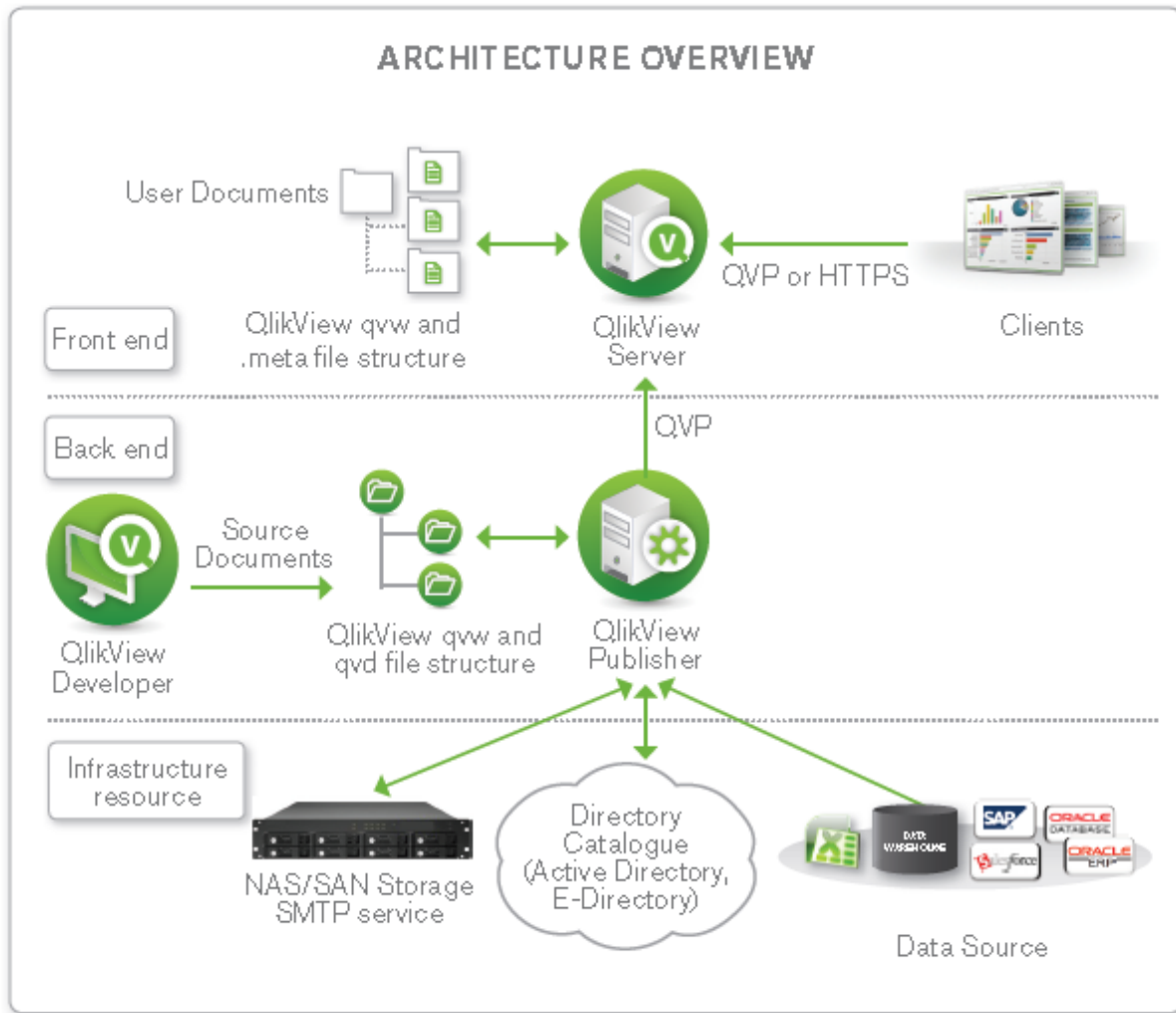
### Security

QlikView Publisher provides access to QlikView applications and data. It is therefore important to integrate QlikView Publisher with the enterprise security solutions in addition to the standard security features of QlikView Server.

QlikView Publisher is viewed as a backend process within the QlikView solution. From a security perspective, it is important to understand that the frontend does not have any open ports to the backend. The frontend does not send any queries to data sources on the backend, nor do any of the user documents (.qvw files) contain any connection strings to data sources located on the backend. End users can only access QlikView documents that exist on the frontend. Within the backend, the Windows file system is always in charge of authorization; QlikView is not responsible for access privileges.

The figure below shows a simplified view of a standard QlikView deployment containing the location of the QlikView products and the data and applications.





### Directory Services

To provide security for QlikView documents, QlikView Publisher can connect to an external directory service (for example, Active Directory, LDAP, a database, or other sign-on solutions). The external directory service is an authentication source with which QlikView has a trust relationship.

QlikView provides a built-in Directory Service Provider (DSP) for Active Directory that allows QlikView administrators to assign Active Directory user privileges to QlikView documents or portions thereof. QlikView Publisher leverages this built-in provider to provide direct integration with, and support for, Active Directory.

QlikView also provides a means of creating a Configurable LDAP for other directory services. A Configurable LDAP enables QlikView administrators to grant privileges to users authenticated by any authentication system other than Active Directory.

### QlikView Server Authorization Modes

QlikView Server provides two mutually exclusive options for authorizing access to QlikView documents. Depending on the authorization mode of QlikView Server (NTFS or DMS), Publisher populates the

appropriate Access Control List (ACL) when assigning rights to a document. In case of NTFS authorization, Publisher populates a standard NTFS ACL when sending documents to QlikView Server. In case of DMS authorization, Publisher populates an ACL contained within a *.meta* file associated with the application.

### Static Data Reduction

Data reduction is a security mechanism that allows application data to be purged from a QlikView application in accordance with row-level security settings. QlikView Publisher can automate data reduction independently of the applicable security scenario. However, Publisher allows an administrator to configure data reduction based on users or groups defined within any external authentication source available through a custom or Active Directory DSP. Publisher performs the data reduction using the “loop and reduce” functionality in QlikView. The Publisher data reduction should not be confused with the dynamic data reduction associated with Section Access.

### Configuring QlikView Publisher Clustering



*The instructions in this section are valid for Windows Server 2008 R2 and later.*

### Requirements

The following requirements must be fulfilled before starting the QDS cluster configuration:

- A QlikView Publisher license that supports more than one QDS. The Publisher LEF must contain the entry `NUMBER_OF_XS;N;;`, where N is 2 or higher.
- QlikView AccessPoint (based on QlikView Web Server or Microsoft IIS), QlikView Management Service (QMS), QlikView Server (QVS), and DSC are already installed in the QlikView system in the network.
- A domain user to run the QlikView services on every machine is available.
- A shared storage device; Qlik recommends a shared device mounted as a Windows-based file share.

All QDS cluster nodes need read and write access to the following, centrally stored data:

- QlikView Publisher status, configuration, and log files
- QlikView source documents

### Step-by-step Instructions

#### Prepare the Shared Storage Device

Create folders for the files accessed by every Publisher cluster node:

- `\\<server1>\ProgramData\QlikTech\DistributionService` (application folder)
- `\\<server1>\ProgramData\QlikTech\SourceDocuments` (source documents folder)

#### Prepare the Cluster Nodes

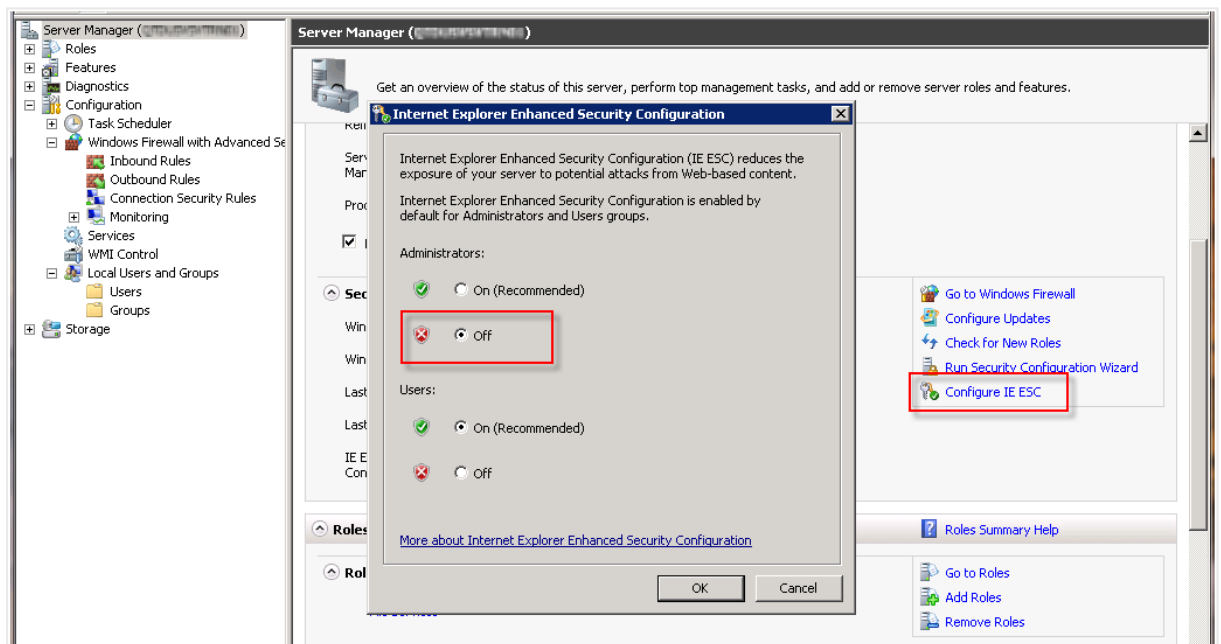
Proceed as follows on each planned QDS cluster node:

1. Login as administrator.
2. Configure the firewall to secure the QlikView solution. The QlikView services require the ports listed in the table below to be “opened”.

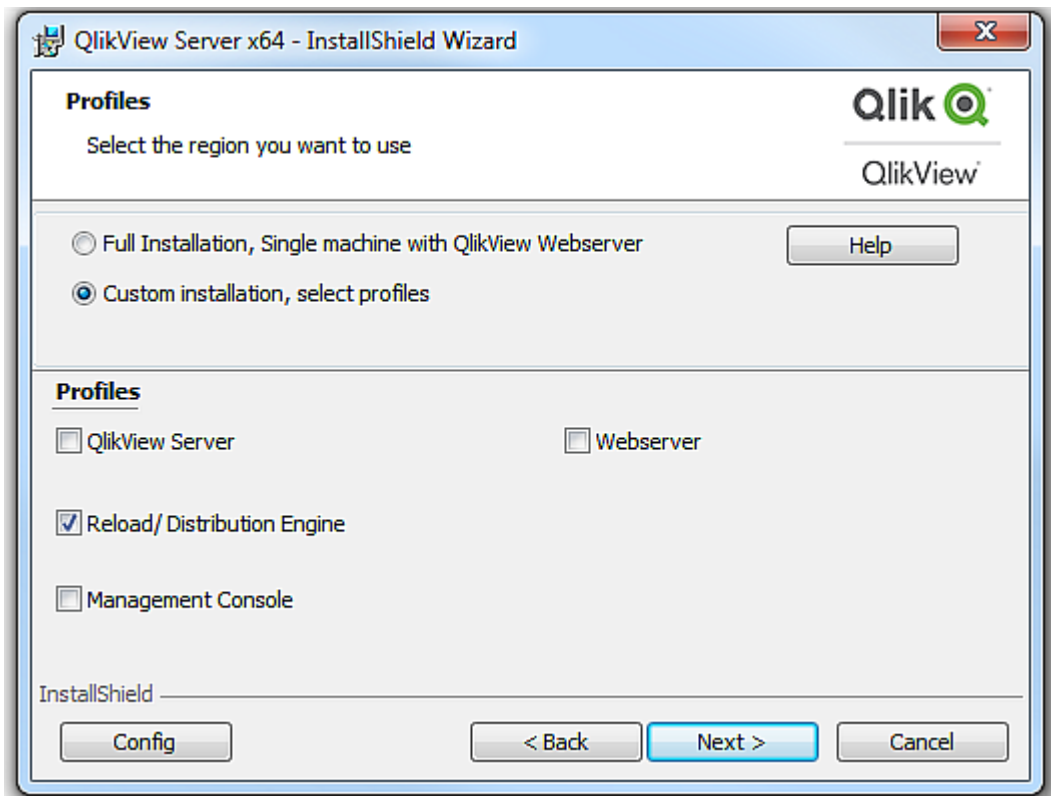
## 2 Planning QlikView Deployments

| Service   | Port     |
|---|----------|
| QDS (Publisher) (required for Publisher)        | 4720/TCP |
| DSC (required for Publisher)                    | 4730/TCP |
| QMS (required for Publisher)                    | 4780/TCP |
| QlikView Web Server/Microsoft IIS configuration | 4750/TCP |
| QVS configuration                               | 4749/TCP |
| QVP communication                               | 4747/TCP |
| QMS (EDX calls) (required for Publisher)        | 4799/TCP |

- Deactivate the Internet Explorer Enhanced Security Configuration for administrators. By default, Windows Server 2008 and later ship with this configuration enabled, which is basically a locked down version that adds a bit of extra security to the servers for web browsing. When the configuration is enabled, it may cause problems in viewing the QMC and service content. The Internet Explorer Enhanced Security Configuration can be left turned on, but if any issues arise, turn off the feature for the Administrators group.



- Add the domain user that is used to run the QlikView services to the Local Administrators Group.
- Start the QlikView 64-bit (x64) server setup and select **Custom installation, select profiles**. Then select the **Reload/Distribution Engine** feature and install it on each node where Publisher is to reside.



6. Enter the QlikView service account credentials.
7. Finish the setup and restart the system immediately.

### Configuring QDS Cluster in the QMC

Proceed as follows to configure a QDS cluster in the QMC:

## 2 Planning QlikView Deployments

1. Open QMC and register the QlikView Publisher license with the activated cluster nodes.

The screenshot shows the 'Licenses' tab in the QlikView Management Console (QMC). The 'Type' column lists 'QlikView Publisher' and 'QlikView Server'. The 'Name' column shows 'QMS@q113hangant' and 'QVS@q113hangant' respectively. The 'QlikView Publisher License' section is active, showing the 'Serial and Control' fields. The 'Serial number' field contains 'NUMBER\_OF\_CLUSTER' and the 'Control' field is empty. Below these fields is a text area for 'Paste the contents of LEF file here (optional):' containing the following text: 

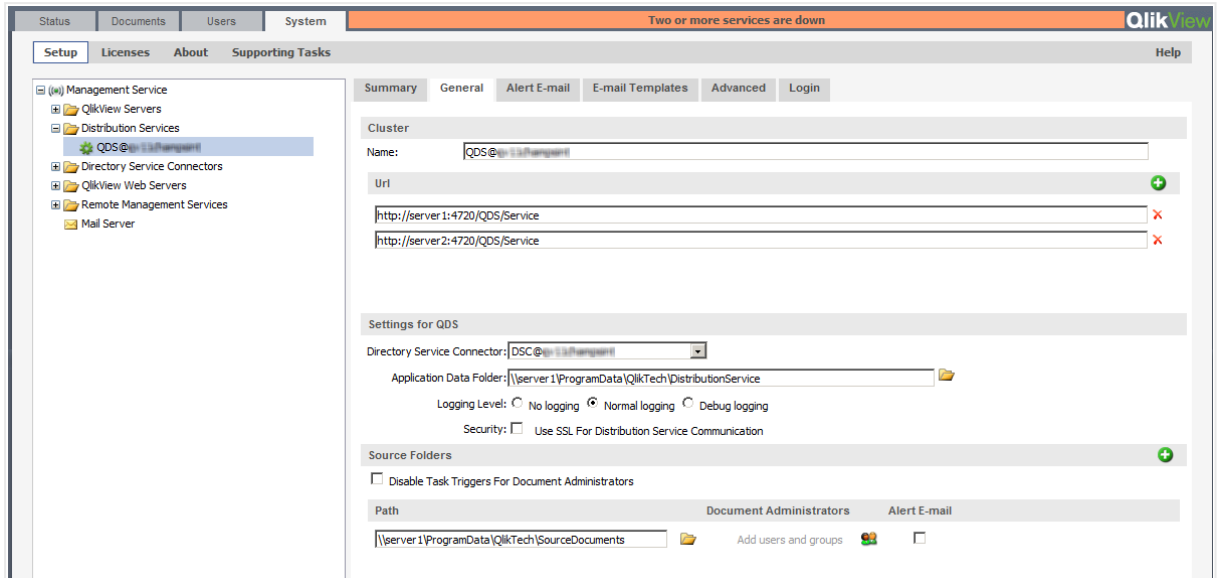
```
SPECIAL_EDITION;CREATE_EXTRACTABLE;;  
SPECIAL_EDITION;EMBED_LICENSE;;  
SPECIAL_EDITION;SITE;;  
X64;YES;;  
IA64;YES;;  
NUMBER_OF_CLUSTER_NODES;8;;  
NUMBER_OF_CPUS;64;;  
DYNAMIC_UPDATE;YES;;  
NUMBER_OF_XS;8;;  
PDF_GENERATION;YES;;  
WEBPARTS;YES;;  
WORKBENCH;YES;;
```

 The 'Owner Information' section shows the 'Name' field empty and the 'Organization' field containing 'QlikTech'.

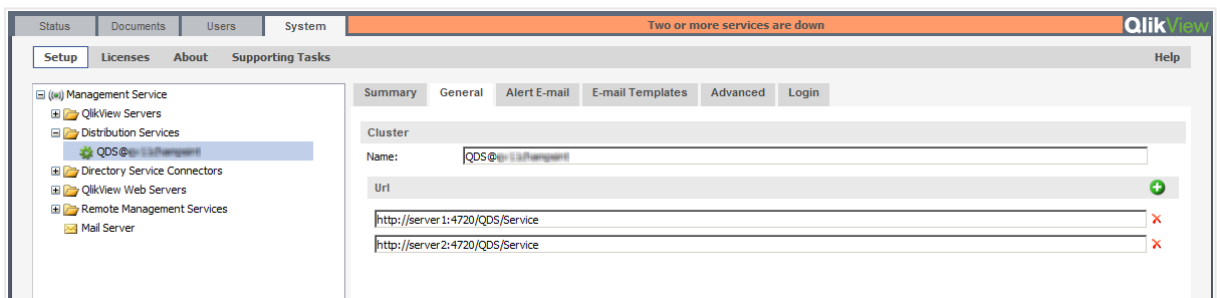
2. On the **System>Setup** tab, add the first QDS cluster node under **Distribution Services**.

The screenshot shows the 'Setup' tab in the QlikView Management Console (QMC). The 'System' tab is selected, and the 'Cluster' configuration window is open. The 'Cluster' section shows the 'Name' field containing 'QDS@q113hangant' and the 'Url' field containing 'http://q113hangant:4720/QDS/Service'. The 'Cluster' section also includes a 'Summary' tab and a 'General' tab. The 'General' tab is selected, showing the 'Cluster' section with the 'Name' and 'Url' fields. The 'Cluster' section also includes a 'Summary' tab and a 'General' tab. The 'General' tab is selected, showing the 'Cluster' section with the 'Name' and 'Url' fields.

3. Switch the **Application Data Folder** and the **Source Folders** to the shared device folder paths using UNC syntax.



4. Click **Apply** and restart the QDS manually.
5. Add each additional QDS cluster node in URL format.



6. Click **Apply** and restart the QDS on all nodes manually.

### Clustering QlikView Distribution Service

This chapter discusses the requirements and options for building a clustered and unbalanced QlikView Distribution Service (QDS) deployment.

A QlikView Publisher license is necessary in order to set up clusters. For more information on QlikView Publisher, see the *Clustering QlikView Publisher* (page 59) page.

The QlikView load balancing capabilities are included in the QlikView Management Console. This chapter also discusses how to make this component efficient using distribution groups.

### What is a QDS Publisher Group?

A publisher group is a subset of a QDS cluster. Each publisher group is given a unique name, and the set of QDS nodes (one or more) that are included in this group. A node may exist in any number of publisher groups (zero or more).

Each task is assigned to none or one of these publisher groups. A task assigned to a publisher group is called a Dedicated Task and may only be executed by one of the QDS nodes included in this group. A task not assigned to any publisher group is called a Regular Task and may be executed by any of the QDS nodes (but may be prevented to run on a QDS in a publisher group under certain circumstances).



*The QDS cluster must be setup and functional prior to activating this feature.*

To activate this feature, make a copy of *DistributionGroupDefinition.Template* in *C:\ProgramData\QlikTech\ManagementService\DistributionGroups* and name it *DistributionGroupDefinition.xml*. Restart the QMS service manually on the QDS cluster node.

### QDS publisher group configuration

You can configure the distribution group using the following settings in the *DistributionGroupDefinition.xml* file.

```
<DistributionGroupDefinition>
<QDSSettings>
<QDS QDIdentifier = "d033930c-0000-e6ec-1519-f3c628a443ae"?
<MaxSimultaneousQvbs>4</MaxSimultaneousQvbs>
<MaxSimultaneousReaderQvbs>2</MaxSimultaneousReaderQvbs>
<DedicatedQvbs>1</DedicatedQvbs>
<RunDedicatedTaskAlone>True</RunDedicatedTaskAlone>
<GraceTimeMinutes>30</GraceTimeMinutes>
<DistributionGroups>
<Group>Group A</Group>
<Group>Group B</Group>
</DistributionGroups>
</QDS>
```

For each QDS in a publisher group, the following should be configured:

- *MaxSimultaneousQvbs* - The maximum number of simultaneous QlikView Batch instances (default 4).
- *MaxSimultaneousReaderQvbs* - The maximum number of simultaneous QlikView Batch readers (default 20).
- *DedicatedDistributionQvbs* - The number of dedicated QlikView Batch instances (default 0).
- *RunDedicatedTaskAlone* - Whether to run dedicated tasks alone or not (default false).
- *GraceTimeMinutes* - If *RunDedicatedTaskAlone* is set to *True* and this setting means that no regular task may be started by this QDS within number of minutes or less until the nearest dedicated task is scheduled (default 0).

The following table provides an example of the number regular and dedicated tasks that may be started based number of dedicated task currently running if *MaxSimultaneousQvbs* is set to 4 and *DedicatedQvbs* is set to 2.

| Number of dedicated | Number of new dedicated task that | Number of regular task that |
|---------------------|-----------------------------------|-----------------------------|
|---------------------|-----------------------------------|-----------------------------|

| task running | may be started | may be started |
|--------------|----------------|----------------|
| 0            | 4              | 2              |
| 1            | 3              | 2              |
| 2            | 2              | 2              |
| 3            | 1              | 1              |
| 4            | 0              | 0              |

A QVB should always be available for dedicated tasks if the *RunDedicatedTaskAlone* option is set to *True*. The following table provides an example of the number regular and dedicated tasks that may be started based number of dedicated task currently running if *MaxSimultaneousQvbs* is set to 4, *DedicatedQvbs* is set to 2 and *RunDedicatedTaskAlone* is set to *True*.

| Number of dedicated task running | Number of new dedicated task that may be started | Number of regular task that may be started |
|----------------------------------|--|--|
| 0                                | 4  | 2  |
| 1                                | 3  | 0  |
| 2                                | 2  | 0  |
| 3                                | 1  | 0  |
| 4                                | 0  | 0  |

### Task Configuration

Once you have created a publisher group, the feature is active and each existing task is considered to be a regular task. When creating a new or editing an existing task, a **Publisher Group** dropdown is available on the Source Document's General tab.

This drop-down contains the names of all publisher groups. If a publisher group is assigned to a document, all task associated with this document dedicated. Select **<any>** from the publisher groups dropdown to make tasks associated with a document regular. A regular task may be executed on any node.

### QlikView Server Extensions

#### Adding Extensions to QlikView Server

To run QlikView Extensions on a QlikView Server, the contents of the *Extensions* folder have to be copied from *%UserProfile%\AppData\Local\QlikTech\QlikView\Extensions\Objects* to the *%ProgramData%\QlikTech\QlikViewServer\Extensions\Objects* folder on the server.

If the path to the extensions is changed (for example, to a common place for all servers in a cluster), that path must be used instead. Note that the path set corresponds to *%UserProfile%\AppData\Local\QlikTech\QlikView\Extensions* (that is, it does not include *\Objects*).



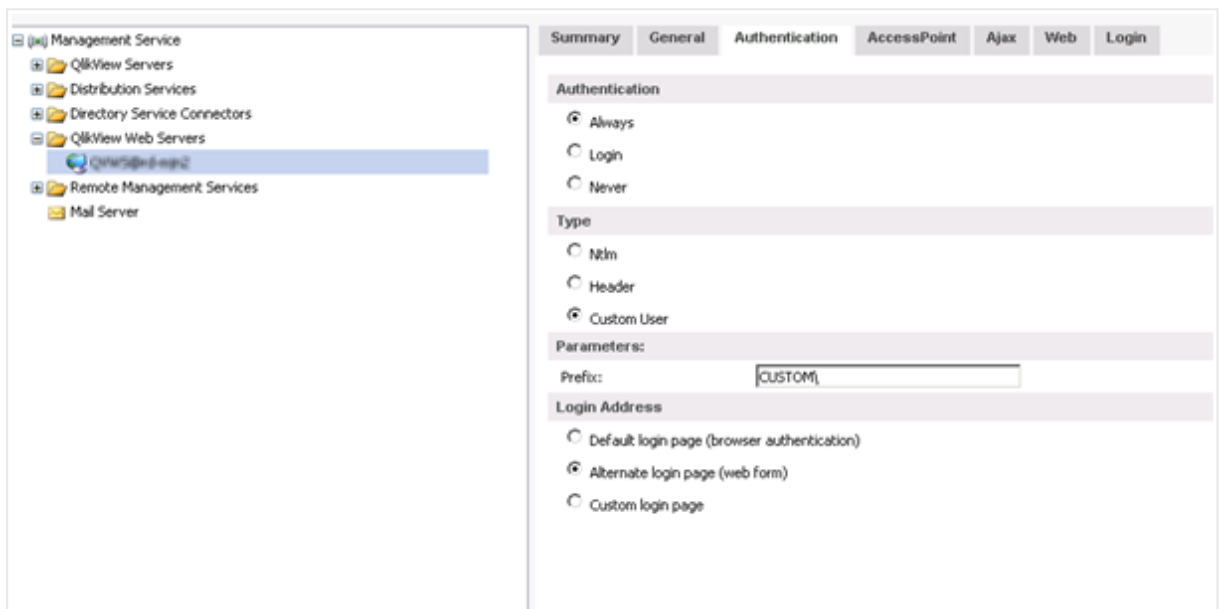
### Configuring IIS for Custom Users

When using Microsoft IIS as web server for Custom Users, configuration is needed.

Proceed as follows to configure IIS for Custom Users:

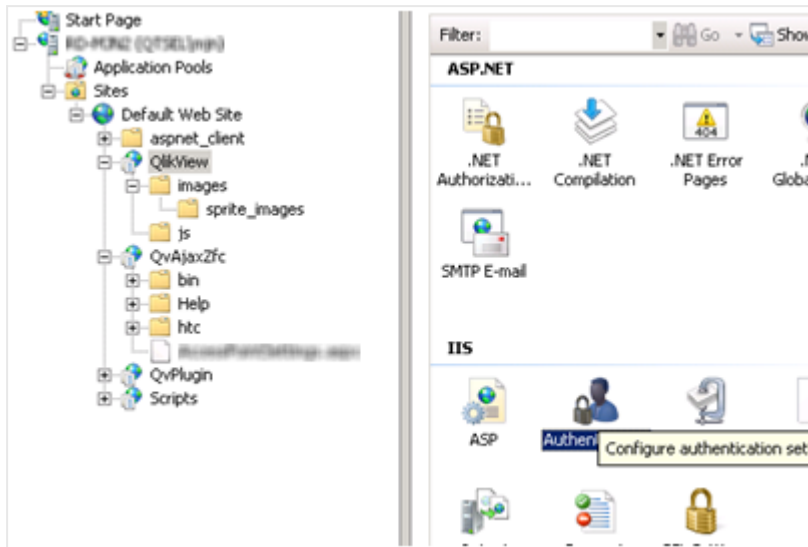
1. In QlikView Management Console, change the parameters on the **System>Setup>Authentication** tab in accordance to the following:

|                       |                                 |
|-----------------------|---------------------------------|
| <b>Authentication</b> | Always                          |
| <b>Type</b>           | Custom User                     |
| <b>Parameters</b>     | CUSTOM\                         |
| <b>Login Address</b>  | Alternate login page (web form) |



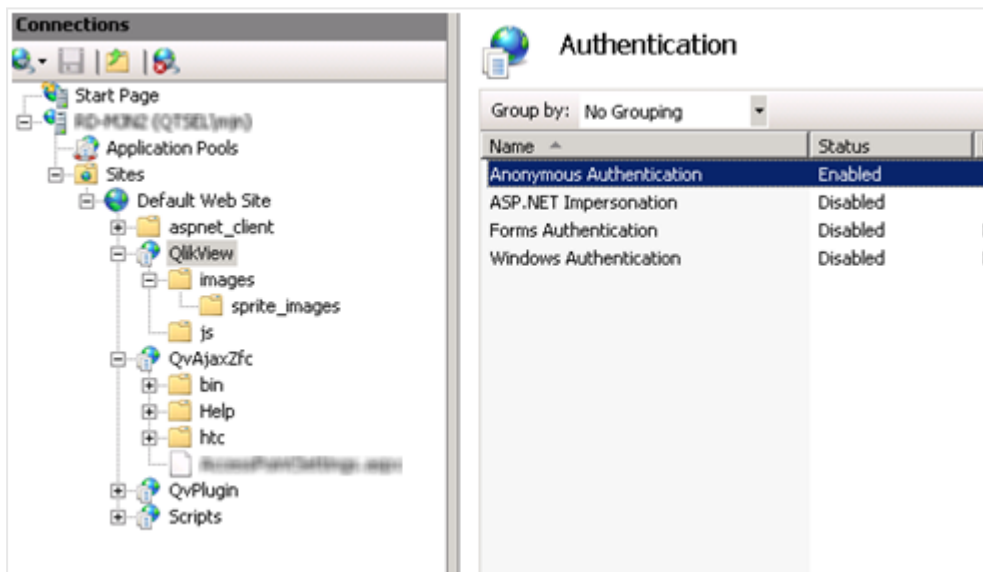
*Authentication tab*

2. Select the qlikview virtual folder and then **Authentication**.



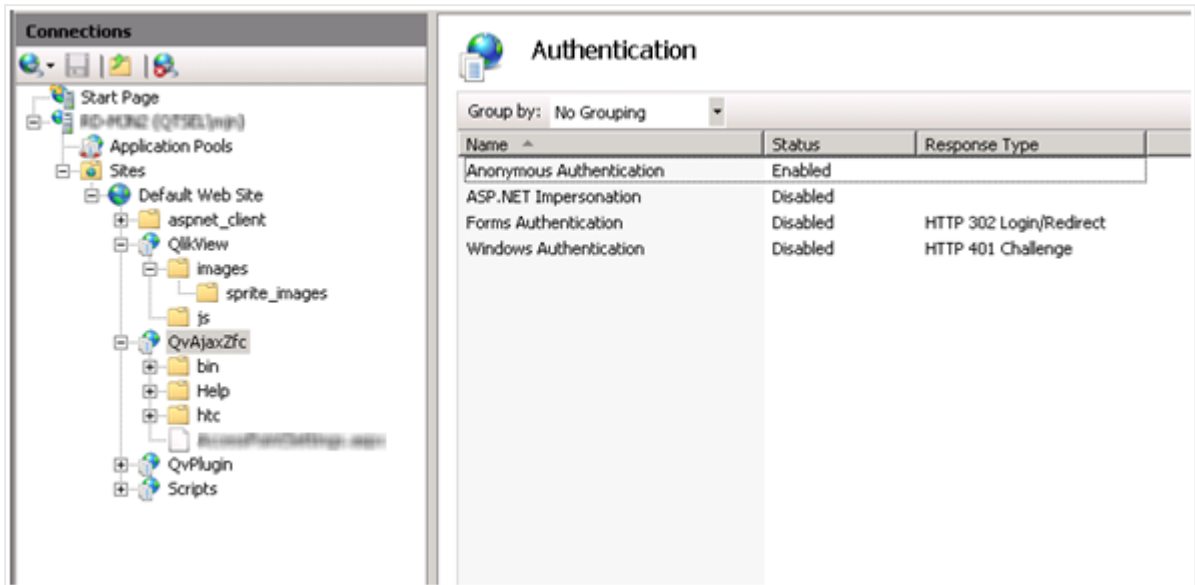
### *Selecting Authentication*

3. Disable **Windows Authentication** and enable **Anonymous Authentication**.



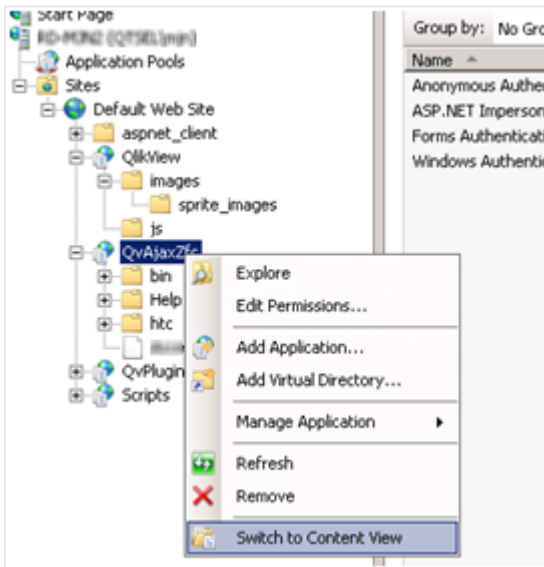
### *Enabling Anonymous Authentication for the QlikView virtual folder*

4. Select the QvAjaxZfc folder and then **Authentication**.
5. Disable **Windows Authentication** and enable **Anonymous Authentication**.



*Enabling Anonymous Authentication for the QvAjaxZfc folder*

6. Right-click QvAjaxZfc and select **Switch to Content View**.



*Selecting Switch to Content View*

7. The configuration of IIS for the Custom User is complete.

## QlikView Triggering EDX Enabled Tasks

To start tasks that have an external event as trigger, the QlikView Management Service API (QMS API) must be used. The user making the request calls must be a member of the QlikView Administrators local group or the QlikView EDX local group. The QlikView Administrators group is set up during the installation of

---

## 2 Planning QlikView Deployments

---

QlikView Server, but the QlikView EDX group must be created manually in **Computer Management**. Members of the QlikView EDX group only have the right to trigger EDX-enabled tasks.

The method to use has the following signature:

```
TriggerEDXTaskResult TriggerEDXTask(Guid guid, string taskNameOrId,  
                                     string password, string variableName,  
                                     List<string> variableValues)
```

| Parameter      | Purpose  |
|----------------|--|
| guid           | ID of the QlikView Distribution Service (QDS) where the task is defined. |
| taskNameOrId   | Task name or ID of the task in string format.                            |
| password       | Password (if required by the task).                                      |
| variableName   | Variable name (if required by the task).                                 |
| variableValues | List of values for the variable.   |

The returned result contains information on whether the task was successfully started or not.

The example below shows how to trigger a task and wait until it has finished or until a certain amount of time has passed.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading;
using QMSAPI;

class Program
{
    static void Main(string[] args)
    {
        try
        {
            // create a QMS API client
            IQMS apiClient = new QMSClient();

            // retrieve a time limited service key
            ServiceKeyClientMessageInspector.ServiceKey =
            apiClient.GetTimeLimitedServiceKey();

            //Get a Distribution Service.
            ServiceInfo qdsService =
            apiClient.GetServices(ServiceTypes.QlikViewDistributionService).FirstOrDefault();

            if (qdsService != null)
            {
                //Trigger the task
                TriggerEDXTaskResult result =
                apiClient.TriggerEDXTask(qdsService.ID, "PauseEDX", "edx", "", new List<string>());

                EDXStatus executionStatus = null;

                //wait until the task is completed or 60 seconds has passed.
                Spinwait.SpinUntil(() =>
                {
                    System.Threading.Thread.Sleep(1000);
                    Console.WriteLine("checking the task...");

                    //Get the current state of the task.
                    executionStatus =
                    apiClient.GetEDXTaskStatus(qdsService.ID, result.Execid);

                    //Return true if the task has completed.
                    return executionStatus !=
                    null && executionStatus.TaskStatus == TaskStatusValue.Completed;
                }, 60 * 1000);

                //write the result
                if (executionStatus != null)
                    Console.WriteLine(executionStatus.TaskStatus);
                else
                    Console.WriteLine("Failed to get execution status.");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("An exception occurred: " + ex.Message);
        }
        // wait for user to press any key
        Console.ReadLine();
    }
}
```

The example comes from the QMS API documentation, which is installed as part of the QlikView Management Console (QMC). It contains detailed information on the available methods and how to get started with the QMS API.

## Cleaning and converting the shared files

The QlikView shared file cleaning tool is a command line tool that allows system administrators to verify (analyze) and purge (repair) shared files. This tool can also be used to convert between different shared file formats, see [Converting the shared files](#). You can invoke it by running the QlikView Server executable (QVS.exe) with special parameters.

There are two modes available with the cleaning tool, each is specified by a different command-line parameter.

### Verify mode

Use the `-v` parameter to analyze the shared file specified in the command-line. During analysis, the cleaning tool detects if there is one or more invalid or corrupted object entries. The QVS then logs as much information as possible about the invalid entries.

### Purge mode

Use the `-p` parameter to verify the shared file and then create a new shared file with the corrupt entries removed. This clean version is placed into the same folder as the original. The new file uses the suffix `_clean` after the format (`.Shared` or `.TShared`), and the original shared file is not overwritten. You can then decide to replace the original shared file with the clean version.

### Converting the shared files

When you create shared files, you can save them in original or transactional format. The original format is recognizable by the `.Shared` ending, while the transactional shared file format ends with `.TShared`. A shared file using the transactional `.TShared` format is more reliable in case of failures, such as network issues, power outages, or insufficient storage space on disk. We recommend to use the `.TShared` format for files larger than 2 GB, because this format can handle file size up to 16 EB (exabyte).

You can use the two different formats, original and transactional, simultaneously for different applications on the same server. However, only one format (either `.Shared` or `.TShared`) should be used in a given application. You can decide which format to use when creating a new shared file by configuring the `Settings.ini` file. For QlikView Server, the `Settings.ini` file is located in `C:\ProgramData\QlikTech\QlikViewServer`.

Set the file format:

```
DefaultBlobDbType=0
```

With this setting, the `.Shared` format is used when creating new shared files.

```
DefaultBlobDbType=1
```

With this setting, the `.TShared` format is used when creating new shared files.

You can also convert the shared files using the QlikView shared files cleaning command as shown in the tab below, and in the example n.4 in the [Examples](#) section at the bottom of the page.

### Setting and changing ownership of shared file content

You can change the owner of server objects with QMC, but for some object types ("DocumentContent", "InputFieldValues" and "ObjectContent") ownership cannot be changed this way. In this case you need to use the cleaning tool to change ownership, using the `-so` (set ownership) or `-ro` (replace ownership) parameters. These parameters should be used in purge mode.

### Cleaning tool command format

The cleaning tool command format is as follows:

```
"<QVS_executable_path>" -x "<Shared_file_path>" <Cleaning_tool_mode> <Output format> <Ownership> <Delete_user_entries> [-l "<Log_folder_path>"] [-rBM <BM_size>] [-o "<Shared_file_save_path>"]
```

The following table describes each command parameter.

## 2 Planning QlikView Deployments

| Parameter           | Description  |
|---------------------|--|
| QVS_executable_path | The full path to the system folder containing the QVS executable (QVS.exe).  |
| -x                  | The -x parameter tells the QVS to only run the cleaning tool.  |
| shared_file_path    | <p>The path to the shared files to clean.<br/>It accepts a path to a directory or a path to a file.</p> <ul style="list-style-type: none"><li>• If invoked with a path to a folder, the operation applies to all shared files in the folder.</li><li>• If a single file is specified, the operation is applied to this item only.</li></ul>  |
| cleaning_tool_mode  | <ul style="list-style-type: none"><li>• -p for purge mode</li><li>• -v for verify mode</li></ul>   |
| Output format       | <p>[Optional] The -f (specify output format) parameter allows to use the cleaning tool to convert between shared file formats.</p> <p>The format can be specified as same, orig or tx (e.g. -f tx).</p> <ul style="list-style-type: none"><li>• same the file format of the input file will be used</li><li>• orig the original <i>.Shared</i> format is used as output format</li><li>• tx the <i>.TShared</i> (transactional file) format is used as output format</li></ul> <p>When the format (-f parameter) is not specified, the default option is same.</p> |
| Ownership           | <ul style="list-style-type: none"><li>• -so user to set ownership</li><li>• -ro from_user to_user to replace ownership</li></ul>   |

| Parameter                | Description  |
|--------------------------|--|
| Delete_user_entries      | <ul style="list-style-type: none"> <li>• -du0 user deletes non-shared entries from the user</li> <li>• -du1 user deletes all entries from the user</li> </ul> <p>This field accepts a path to a file if more than one user needs to be removed</p> <ul style="list-style-type: none"> <li>• -df0 file.txt deletes non-shared entries from the users listed in the file file.txt</li> <li>• -df1 file.txt deletes all entries from the users listed in the file file.txt</li> </ul> <p>To obtain a list of users that have accessed the QlikView servers, the Governance Dashboard application can be used. It is available for free in our <a href="#">download site</a> (see associated documentation <a href="#">here</a>).</p> <p>The list of users can be easily extracted by exporting to 'csv' format the ListBox '<i>Authenticated User</i>' in the Operations/Session sub-tab of the Governance Dashboard. This list can then be edited (keep only the users to be removed from the shared file) and passed on to the Cleaning Tool as an input.</p> |
| -l Log_folder_path       | [Optional] If you want to change the location of the generated log file, use -l and provide a log folder path.   |
| -rBM BM_size             | [Optional] The -rBM parameter is used to remove large bookmarks from the shared file. All bookmarks larger than <BM_size> (in bytes) will be removed.  |
| -o shared_file_save_path | [Optional] The -o parameter is used to change the path to where shared files are saved.  |

### Using the shared file cleaning tool

The share file cleaning tool is run by using the Windows Command Prompt in Administrator mode. Do the following:



*It is recommended to run the cleaning tool with a copy of the QVS.exe and the shared file in a (temporary) folder different from %ProgramData%\Qliktech\Documents. The user running the cleaning tool on the %ProgramData%\Qliktech\Documents folder must have administrator rights over it.*

*The cleaning process completely regenerates the shared file. Issues regarding fragmentation of the file will disappear and file size and access time may be reduced.*





*You can run the cleaning tool for a folder by using the option `-subF`. It is very important to take into account that the list of users to be removed will be common to all shared files within the folder.*



*Backup your shared files before using the cleaning tool.*

1. Create a copy of the QVS executable. By default the QVS.exe is installed in `C:\Program Files\QlikView\Server`.
2. Navigate to the folder where the copy of the QVS.exe is located and run the cleaning tool in verify mode. For example:  
`"C:\<Temporary_path>\QVS.exe -x "C:\ProgramData\QlikTech\Documents\FinanceAnalysis.qvw.Shared" -v`
3. Locate the `CleaningTool_MACHINENAME.log` verify file log. If not specified in your command, the log is stored by default in `C:\ProgramData\QlikTech\QlikViewServer`.  
The log lists each type of corrupted shared file object if there is corruption. If the corrupt entry can be identified, it will list the object ID.
4. If there are corrupt entries, run the cleaning tool again in purge mode.  
The purge process will create a new shared file with the corrupt objects removed or corrected. The new file identified by the suffix `_clean` (for example: `MYFILENAME.QVW.TShared_clean`) is placed in the same folder as the source shared file.



*The new file may be larger than the source file.*

5. Replace the old corrupt shared file with the new file. This must be done when no QlikView Server services are running.

## Examples

### Example 1: Analyzing a shared file

Running the following command in the windows command prompt analyzes the shared file and creates a log file in the `C:\logs` folder:

```
QVS.exe -x "C:\ProgramData\QlikTech\Documents\TESTFILE.QVW.TShared" -v -l "C:\logs"
```

### Example 2: Setting file ownership

Running the following command in the windows command prompt sets ownership of the server objects in the shared file to user UserX:

```
QVS.exe -x "C:\ProgramData\QlikTech\Documents\TESTFILE.QVW.TShared" -p -so UserX
```

### Example 3: Replacing file ownership

Running the following command in the windows command prompt replaces ownership of the server objects in the shared file from UserX to UserY:

```
QVS.exe -x "C:\ProgramData\QlikTech\Documents\TESTFILE.QVW.Shared" -p -ro UserX UserY
```

### Example 4: Changing output format

Running the following command in the windows command prompt allows to convert a file in the original shared file format to the new format:

```
QVS.exe -x "C:\Temp\1.QVW.Shared" -p -f tx
```

### Example 5: Removing non-shared entries from a specific user

Running the following command in the windows command prompt removes all non-shared entries associated to a specified user UserX:

```
QVS.exe -x "C:\ProgramData\QlikTech\Documents\TESTFILE.QVW.TShared" -p -du0 UserX
```

### Example 6: Removing all entries from a set of users specified in a text file

Running the following command in the windows command prompt removes all entries (including the ones that are shared) associated to a list of specified users in the Users.txt column text file.:

```
QVS.exe -x "C:\ProgramData\QlikTech\Documents\TESTFILE.QVW.Shared" -p -df1 "C:\temp\Users.txt"
```

Example of the Users.txt file:

DOMAIN\User1

DOMAIN\User2

DOMAIN\User3

...

DOMAIN\UserX

### Example 7: Removing all entries from a set of users specified in a text file for a whole folder

There is also the possibility of processing a whole set of shared files contained within a folder with a common list of users to be removed from them.

Running the following command in the windows command prompt removes all (including the ones that are shared) entries associated to a list of specified users in the Users.txt column text file. For all shared files within the folder 'Documents':

```
QVS.exe -x "C:\ProgramData\QlikTech\Documents" -p -subF -df1 "C:\temp\Users.txt"
```

Example of the Users.txt file:

DOMAIN\User1

DOMAIN\User2

DOMAIN\User3

...

DOMAIN\UserX

### 2.3 Security Overview

The security of QlikView Server/Publisher consists of the following parts:

- Protection of the platform: How the platform itself is protected and how it needs to communicate and operate.
- Authentication: Who is the user and how can the user prove it? QlikView uses standard authentication protocols, such as Integrated Windows Authentication (IWA), HTTP headers, and ticketing, to authenticate every user requesting access to data.
- Document level authorization: Is the user allowed to access the document or not? QlikView uses server-side capabilities such as Document Metadata Service (DMS) or Windows NTFS to determine access privileges at file level.
- Data level authorization: Is the user allowed to see all of the data or just parts of it? QlikView implements row and field level data security, using a combination of document-level capabilities (Section Access) and server-side data reduction capabilities (QlikView Publisher).

### Protection of the Platform

#### Functionality

The functionality for downloading documents and/or print and export to Microsoft Excel can be restricted at the user level for each document on the server.

#### Special Accounts

##### Supervision Account

The supervision account is granted access to all documents that are created by tasks in QlikView Publisher. The characteristics of the supervision account are as follows:

- Provides access to all files on the QVS
- Does not provide any access to the QlikView Management Console (QMC)
- Respects the types of clients that are allowed for each document (for example, a supervision account cannot open a QlikView document using the AJAX client, if the AJAX client has been blocked by the user that created the task)

##### Anonymous User Account

When QVS is started for the first time on a machine, a Windows account is created for anonymous users. The account name is `IQVS_name`, where `name` is the name of the machine in the local network.

## 2 Planning QlikView Deployments

---

If the machine in question is a domain server, the anonymous account is created as a domain account. If not, it is created as a local machine account.

Each folder and file that is to be available for anonymous clients must be given read privileges for the anonymous account.



*Start QVS and let it create the anonymous account before attempting to grant any privileges. Do not try to create the anonymous account manually.*

### QlikView Administrators

The QlikView Administrators group is used for granting access to the QlikView Management Console (QMC) as well as authorization of communication between services, if Windows Authentication is used.

### Communication

#### Protection of AJAX Client

The AJAX client uses HTTP or HTTPS as the protocol for communication between the client browser and the QlikView Web Server (QVWS) or Microsoft IIS. It is strongly recommended to protect the communication between the browser and the web server using SSL/TSL encryption over the HTTP protocol (that is, HTTPS). If the communication is not encrypted, it is sent as clear text.

The communication between the web server and QVS uses QVP as described below.

#### Protection of Plugin

The QlikView plugin can communicate with QVS in two ways:

- If the plugin has the ability to communicate with QVS using QVP (port 4747), the security is applied as follows:  
See: *Server Communication (page 84)*
- If the communication cannot use QVP or if the client chooses it in the plugin, the communication is tunneled using HTTP to the web server.

If HTTPS is enabled on the web server, the tunnel is encrypted using SSL/TLS.

#### Server Communication

The QVS communication uses the QVP protocol, which is encrypted by default. The QVP protocol can be protected using 1024-bit RSA for key exchange and 128-bit RC4 for data encryption, provided the Microsoft Enhanced Cryptographic Provider is installed. If the Microsoft Base Cryptographic Provider is used, the protection of the communication is 512-bit RSA for key exchange and 40-bit RC4 for data encryption.

#### Services Communication

The services that are part of the QlikView platform (that is, QVS, DSC, QMC, QDS, and QVWS) all communicate using web services. The web services authenticate using Integrated Windows Authentication (IWA).

#### SSL and TLS support

The following table shows QlikView support for SSL and TLS.

|                              | SSL v3.0 | TLS v1.0 | TLS v1.1 | TLS v1.2 |
|------------------------------|----------|----------|----------|----------|
| QlikView 11.20 SR12          | √        | √        |          |          |
| QlikView 11.20 SR16          | √        | √        | √        | √        |
| QlikView 12.00               | √        | √        |          |          |
| QlikView 12.00 SR1 and later | √        | √        | √        | √        |
| QlikView 12.10               | √        | √        | √        | √        |
| QlikView November 2017       | √        | √        | √        | √        |

### Authentication

Although QlikView can be configured to allow anonymous access, the majority of implementations require users to be authenticated. In such environments, QlikView always requires that the user is authenticated when establishing a session via QlikView Server (either through a browser or when downloading and opening a document via the QlikView Desktop client).

In the QlikView context, the authentication of a user is almost always done against an external entity that is then used to pass the externally authenticated user identity to QlikView Server. In such a scenario, QlikView relies on the authentication to be performed prior to accessing QlikView, and that some token of identity is transmitted to, and trusted by, QlikView.

### Authentication when Using QlikView Server in a Windows User Environment

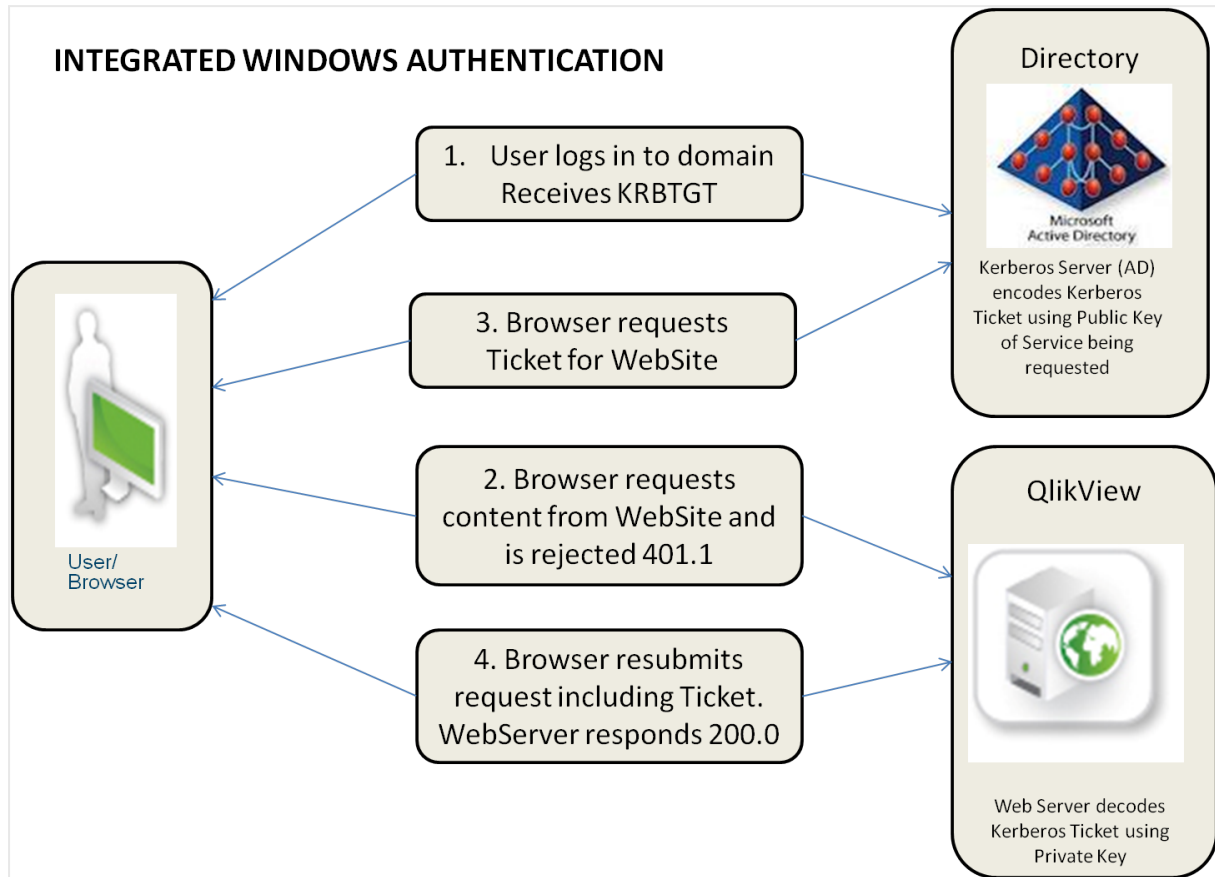
Authentication to a QlikView Server in an environment based on Windows users (for example, incorporating Active Directory) is straightforward. The process is as follows:

1. The user credentials are validated when the user logs in to the Windows operating system on the client machine.
2. Later when the user wants to establish a session with a QlikView Server (QVS) (for example, via a browser on the desktop), QVS can use the built-in Integrated Windows Authentication (IWA).
3. The identity of the logged-in user is communicated to QlikView Server using either the Kerberos or the NTLM security solution. This solution provides single sign-on capabilities right out of the box. In case the authentication exchange fails to identify the user, the browser prompts the user for a Windows user account name and password.



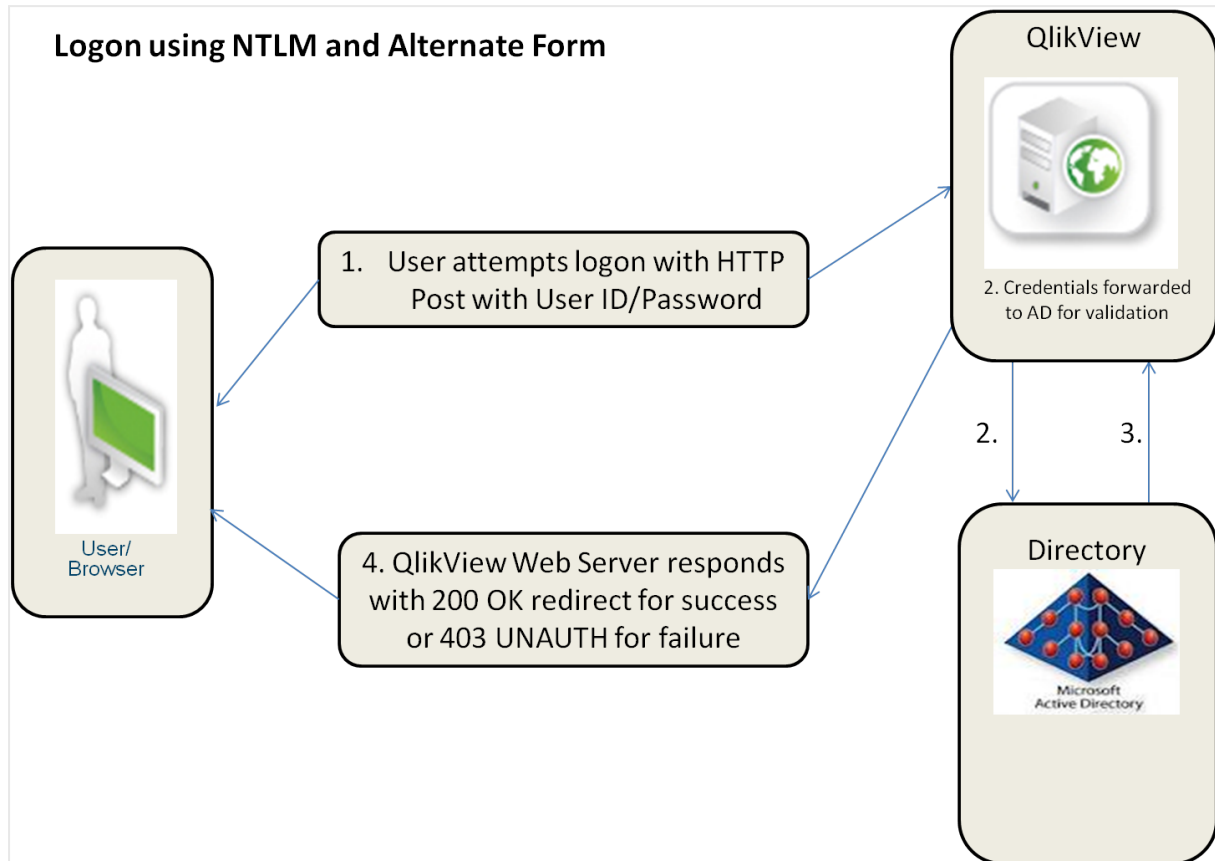
*User groups cannot be transferred to the QlikView system. They have to be resolved by the Directory Service Connector (DSC) or not at all.*

The figure below shows the standard authentication flow for IWA:



*Authentication when using QlikView Server in a Windows user environment*

The figure below shows the authentication flow for the combination of NTLM and alternate login, which differs from the standard flow for IWA:



### *Authentication using NTLM and alternate form*

The authentication process differs based on the environment:

- Local Area Network (LAN): IWA is most common and most suitable for recognizing Windows users on a LAN. The act of authentication is performed when logging in the workstation, and this identity is leveraged by QlikView.
- Multi-domain environment: The internal company network IWA should be avoided in architectures where there is a multi-domain environment with no trust relationship between the domain of the workstation and the domain of the server, or when used across a reverse proxy. In such an environment, configure the QlikView deployment to use either an existing external SSO service or a QlikView custom ticket exchange to expose an authenticated identity to QlikView.

### Authentication with a QlikView Server Using an Existing Single Sign-on Software Package

In environments where an SSO infrastructure already exists (for example, CA SiteMinder®, IBM® WebSeal, or Oracle® Oblix), QlikView can use the HTTP header injection method of single sign-on provided by the SSO infrastructure. This means single sign-on is provided right out of the box. The SSO infrastructure software packages can be configured as follows:

- Repeat user get access: The software packages can be configured to protect a resource. When a user requests access to QlikView, the SSO package grants access, if the user has previously signed in to the SSO authentication page.

- **New user log in:** If the user does not have an existing session with the SSO package, the user is redirected to the SSO package login page. After logging in, the user is redirected to the original URL that the user requested.

In both cases, if the user has properly authenticated to the SSO software, the username is injected into an HTTP header and the value in that header is what the QlikView server accepts as the authenticated identity of the user.



*Unless SSO software is in place, the HTTP header method of authenticating to a QlikView Server must not be used. HTTP headers can easily be spoofed. All of the SSO software packages mentioned above provide protection against this type of spoofing attacks, if the software package is the only path for users to access the content.*

QlikView does not recommend or endorse any specific tool or product for providing identity in HTTP headers. The approach is highly suited to extranet deployments wherein the users may not exist in the internal Active Directory. The act of authentication is performed by the reverse proxy or ISAPI filter that intercepts the attempt of the end user to interact with QlikView content.

### Authentication Using neither IWA nor Single Sign-on Software

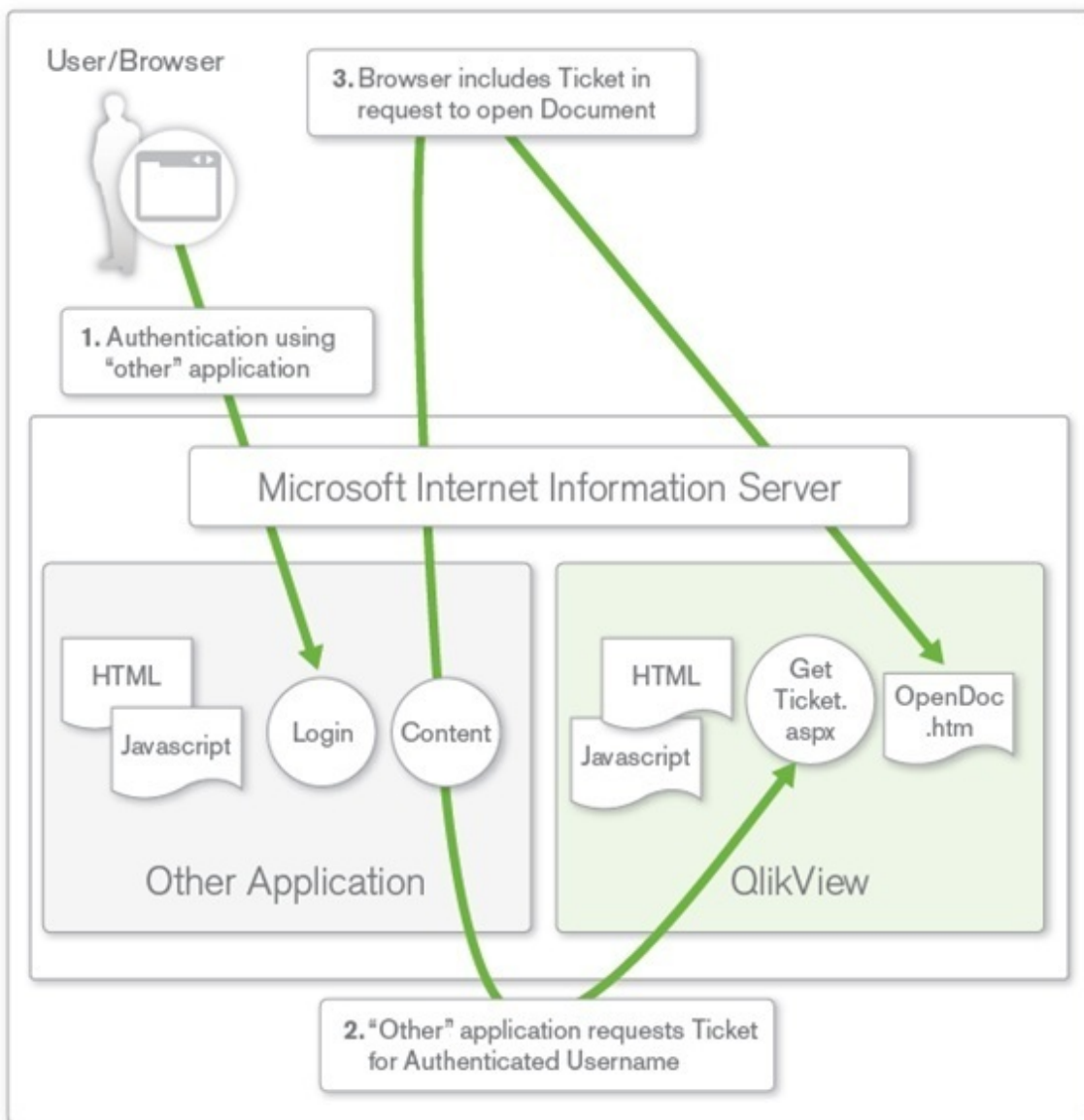
QlikView provides a third method for single sign-on, Custom Ticket Exchange (CTE), when neither of the methods described above is suitable.

CTE relies on the user having authenticated previously to another system:

1. The third-party system is granted the privilege and responsibility to request an authentication token (called a “ticket” in QlikView) from QVS on behalf of the authenticated user of the third-party system. It is the responsibility of the third-party system to only request tickets for users that have been properly authenticated (for example, QVS has no knowledge of the authentication status of the user).
2. The system then passes the authentication token to the user, who uses it in a request to open a session with QVS.
3. QVS checks that the ticket is valid and then opens a session for the authenticated user.

Ticketed authentication is mainly applicable when embedding QlikView content in third-party applications and portals, and is rarely used for providing general access to QlikView. Typically a small amount of custom development is needed to implement the request and passing of the ticket for the CTE method to work.





*Authentication using neither IWA nor single sign-on software*

### QlikView Server Authentication Using Custom Users

The three methods described above all use a single sign-on principle, where the user ID and password are stored externally to QlikView Server and an external entity is responsible for the authentication. Less common, although possible, is the ability to store the user credentials in the QlikView Server environment using the Custom Users functionality in QlikView Publisher. In this case, users and passwords are defined and stored within the QlikView environment and the web tier of the QlikView deployment is responsible for forms authentication. This solution is suitable for smaller, standalone QlikView Server deployments, and must not be used in environments where the user definitions are to be available to multiple systems. In such environments, it is highly recommended to use one of the three single sign-on solutions described above.

## 2 Planning QlikView Deployments

Each coexistent form of authentication may require a distinct web server instance. Several web servers can forward user requests to the same QVS instance(s).



*QlikView Server authentication using Custom Users*

### Authorization

Once a user has been authenticated (that is, the system knows who the user is), the first step in assigning the security privileges has been completed. The second step is to understand the authority or access rights that the user has to applications, data, or both. This step is referred to as Authorization. At a fundamental level, an administrator populates an Access Control List (ACL) with a list of users and/or groups and what they are to have access to. When the time comes for a user to request access, the system looks up the authenticated identity of the user in the ACL and verifies if the administrator has granted the user enough privileges to do so.

Direct access to a QlikView document using QlikView Desktop is always governed by the Windows NTFS file security. Access to the web-based QlikView Management Console (QMC) is restricted to Windows users that are members of a particular local Windows group.

### Document Level Authorization

Once a user has been authenticated, QlikView Server typically handles authorization on its own. QlikView Server provides the choice between storing the ACL information as Windows NTFS privileges (applicable only when the user is authenticated using a Windows user identity) or by storing the ACL information in the internal repository, Document Metadata Service (DMS), in QlikView. The choice of NTFS or DMS affects the access to all documents in QlikView Server.

### NTFS vs. DMS

QlikView Server can use the NTFS privileges of the Windows file system to store authorization information. When in NTFS authorization mode, QlikView Server controls access to a given QlikView document by determining if the authenticated user has NTFS privileges to the underlying QlikView document file (.qvw). This is based on the operating system privileges and Windows NTFS is used for the ACL. The privileges of the authenticated user are configured by a server administrator using standard Windows Explorer functionality via directory properties options.

As an alternative to Windows NTFS, QlikView can use its own ACL, DMS. Unlike NTFS, this allows non-Windows users and groups to be authorized to access applications and data. DMS integrates fully with the existing Directory Service Provider (for example, Active Directory, other LDAP) where Group Membership has been recorded – this is a mechanism by which QlikView Server can re-use existing enterprise accounts and group structures. The permitted users or groups are recorded in a meta file that resides next to the QlikView document, and it is managed using QMC.

NTFS is the default document authorization model, suitable when all users and groups are identified in Active Directory or locally on the QlikView Server host. The NTFS permissions may be inherited from the directory that the QlikView documents are in, or may be assigned using QlikView Publisher distribution tasks.

DMS is required when the authenticated user identity is not a Windows user account. The DMS permissions are explicitly assigned using QMC, or may be assigned using QlikView Publisher distribution tasks.



*When authenticating a user via a web ticket, the user is not a proper Windows user, even if sending in the user name in Active Directory format (for example, QLIKVIEW\jsmith). This means that DMS authorization should be used when using web tickets.*

### Data Level Authorization

Data level authorization allows access to be granted or denied on a document level and even to specific data in a document.

There are two types of data level authorizations:

- **Dynamic data reduction:** Determines if the user is allowed to view the data when the user tries to access it.

- Static data reduction: Performed by QlikView Publisher, determines if the user is allowed to view the data when it is prepared for the user.

Static and dynamic reduction of data can be used on its own, but can also be combined to deliver data level authorization.

### Dynamic Data Reduction

Dynamic data reduction is done in QlikView using the concept of Section Access, which is part of the QlikView document.

Section Access Management is configured in the QlikView Management Console (QMC). For information, see the QMC help.

### Static Data Reduction

For larger deployments and/or those in need of centralized control of authorization capabilities, QlikView Server/Publisher are used. Departments or functions often have a “master” application that contains all relevant data covering all analysis needs, and this master document needs to be separated (“reduced”) according to the needs and access privileges of the intended audience. QlikView Publisher reloads the QlikView document with available data, refreshes the Section Access tables, and splits the large QlikView document into smaller documents based on values in a particular field.

This “reduction and distribution” allows for a file containing many data fields to be broken up by the contents of a field and distributed to authorized users or groups according to their access privileges.

One of the benefits of reducing and distributing source files in this manner is that the documents that are created in this process contain no explicit reference to the source data (for example, a database connection string) in their script environments. Therefore, if a user interacts with the document via QlikView Desktop, the user cannot see the location of the source data. All of the data pertinent to the user needs is contained in the document.

An administrator can use QMC to create tasks on source *.qvw* or *.qvd* files to accomplish this. At a basic level, the steps are as follows:

1. On the source document (either *.qvw* or *.qvd*), apply the data reduction criteria (for example, choose the field name on which to reduce the data).
2. Apply the distribution criteria to the newly created (reduced) files:
  - a. Assign the authorization privileges using either DMS or NTFS ACLs.
  - b. Choose the type of distribution (for example, *.qvw* files or *.pdf* report).
  - c. Choose the location for the newly created files.
3. Apply the notification criteria for the completion of the task (for example, e-mail notification).

The newly created files only contain the data that the user or group is authorized to see, since the data has been “reduced” from the master document in accordance to the reduction criteria. This is why the process is termed “Static Data Reduction”. Hence, there is no risk of an unauthorized person viewing data, since only authorized data exists in each file.

### Certificate Trust

QlikView Server can use certificate trust for authentication and authorization. A certificate provides trust between servers (that is, machines). In addition, dynamic encryption keys are used for sensitive data. The default configuration in QlikView relies on Windows trust (hard-coded cryptographic keys).



*Since the certificates contains encryption keys it is vital to keep a backup of the certificates in a safe place.*

This section describes how to deploy certificates on multiple servers.



*You must reference the QlikView Server by its machine name, and not by the IP address or fully qualified domain name.*

### Architecture

Certificates are used in a QlikView installation to authenticate and authorize communication between services that reside on multiple servers. The certificates include a SecretsKey that handles encryption and decryption of data such as passwords and connection strings.

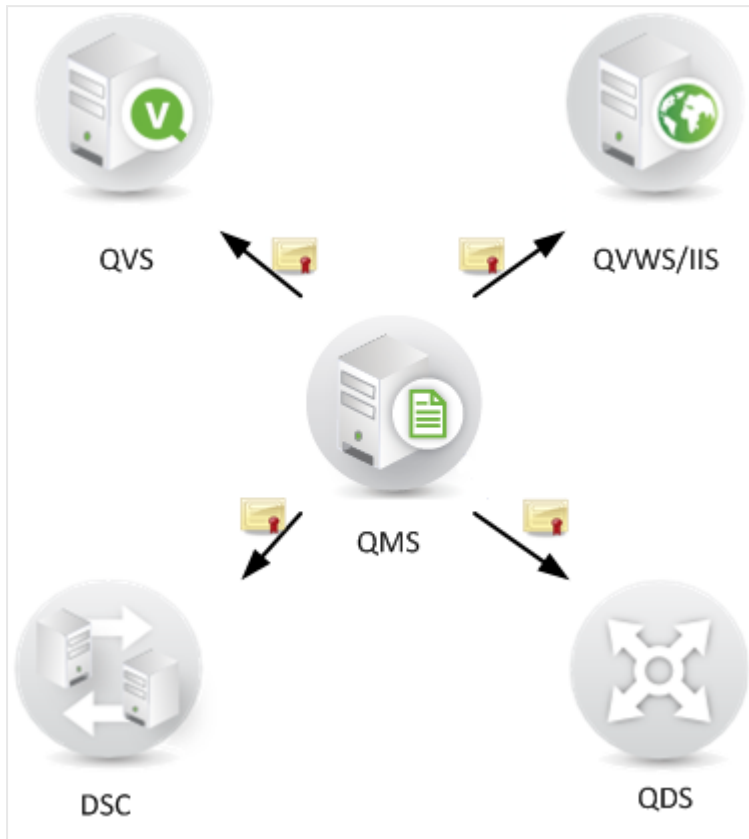
Configuring certificates in a multiple server deployment within QlikView removes the dependency on a QlikView Administration Group for the establishment of trust between the QlikView services. It also allows the use of certificates to build a trust domain between QlikView services that are located in different domains without having to share an Active Directory (AD) or other user directories.



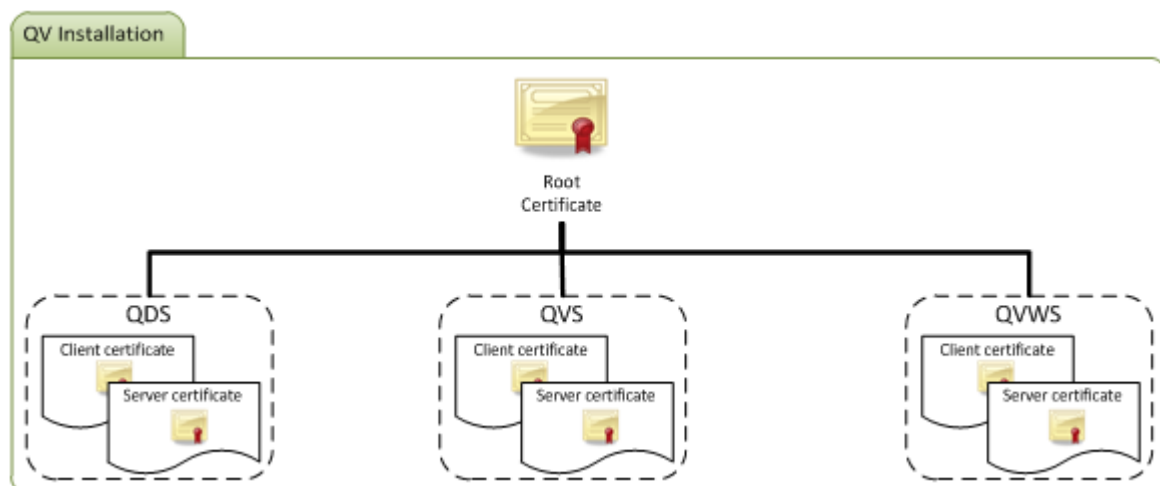
*The configuration steps described in this section only provide a trust domain between the QlikView services. The use of SSL and certificates for securing end-user communication has to be configured separately.*

The architecture is based on the QlikView Management Service (QMS) acting as the certificate manager or Certificate Authority (CA). The QMS can create and distribute certificates to all services in the QlikView installation.

QMS is therefore an important part of the security solution and has to be managed from a secure location to keep the certificate solution secure.



The root certificate for the installation is stored on the QMS server. All servers with QlikView services that are to participate in the installation receive certificates signed using the root certificate when added to the QMS. The QMS (that is, the CA) issues digital certificates that contain keys and the identity of the owner. The private key is not made publicly available – it is kept secret by the QlikView services. The certificate enables the QMS to validate the authenticity of the service. This means that the QMS is responsible for saying “yes, this service deployed on this server is a service in my installation”.



After the servers have received certificates, the communication between the QlikView services is encrypted using HTTPS (SSL encryption). The certificates only secure the communication between the services on the servers. The certificates do not secure the communication with the end user (that is, the certificates are not used for QlikView plugin, client, or web server communication with the QVS).

### Requirements

#### General

The following requirements must be fulfilled for the certificate trust to function properly:

- Certificate trust cannot be partially implemented. It is either used by all services in the QlikView installation or not at all.
- Certificate trust is only supported by Windows Server 2008 and later.
- Make sure that all machines use QlikView 12 Server.
- If it is an initial installation of QlikView Server, install and configure the QlikView services without any modification. Prior to configuring the use of certificates, start and stop the services on the servers (that is, machines) where the QlikView services are deployed.
- Section Access management must not be configured in environments where certificate trust is configured.

In addition, the technical requirements described in the following sections also have to be fulfilled.



*Make sure to back up all three certificates (**Root**, **Client** and **Service**) on each server every time they are updated.*

#### Expiration

The certificates expire after 10 years, but can be updated at any time, if desired. The expiration date of the certificates is displayed in the QMC. When 30 days or less remain before the expiration date, a warning is displayed in the QMC, so that you can create new certificates before they expire. Certificates should not be replaced, but updated. Removing existing certificates may result in undecryptable data.

Besides expiration of certificates, there may be other reasons to update, for example replacing a computer or changing a computer name, since one of the certificates is linked to the computer name.

#### Undecryptable Data

At start-up, each service validates all its encrypted data entries to ensure they are accessible. If the service encounters data that cannot be decrypted, it reports it as an error and stops its own execution. There are two reasons that prevent the service from decrypting its data:

- The certificate containing the required encryption key is missing.
- The encrypted data itself is corrupted.

If you encounter any of these problems, you have two options:

1. If the reason was a missing certificate, the preferred option is to re-install the certificate from a backup. After this you may start the service again.

2. If the missing certificate is truly gone (or the encrypted data was corrupted) you will have to erase the undecryptable data.

### Erasing the Undecryptable Data

Proceed as follows to temporarily activate the hidden configuration flag called `EraseUndecryptableData`:

1. Stop the service.
2. Run Notepad as administrator.
3. Open the configuration file in Notepad.
4. Add the “`EraseUndecryptableData`” entry and set it to “true”.
5. Save the file.
6. Restart the service.

When the service starts, the part of the data that is undecryptable is erased, but all other data is left intact.

7. Stop the service, open the configuration file and remove the “`EraseUndecryptableData`” entry.
8. Save the file and restart the service.

The service starts normally.

Now you need to re-enter the erased data in the QMC. All the undecryptable data entries have already been listed in the service’s log file, and that gives you a hint of what to re-enter in the QMC.

### Communication Ports

This section describes the ports that are needed when using certificate trust.

The ports that are listed in the following table are needed for service to service communication and have to be configured as “open”.



*Firewall configuration changes might be necessary, depending on the location of the QlikView servers within the resulting network and the routing of the QVS communication.*

| Service                       | Ports         | SSL-enabled Ports |
|-------------------------------|---------------|-------------------|
| QlikView Server               | 4747, 4749    | 4749              |
| QlikView Distribution Service | 4720          | 4720              |
| QlikView Web Server           | 4750, 80, 443 | 4750, 443         |
| QlikView Management Service   | 4780, 4799    | 4780, 4799        |
| Directory Service Connector   | 4730          | 4730              |

The ports that are listed in the following table are needed for the certificate installation procedure on the local server.



*The ports are not used for service to service communication.*



## 2 Planning QlikView Deployments

| Service                       | Ports |
|-------------------------------|-------|
| QlikView Distribution Service | 14720 |
| Directory Service Connector   | 14730 |
| QlikView Web Server           | 14750 |

The following table lists the protocols that are used for communication on the ports that are specified in this section.

| Service            | Ports         |
|--------------------|---------------|
| QlikView Server    | QVPX over SSL |
| All other services | SOAP over SSL |

### Access

To install the distributed certificates for the respective services, physical access to the console or remote access to the console (for example, using remote desktop functionality) is needed.

### Installation

Only install the QlikView services (components) needed on each server. Do not perform a full install on all servers – use “custom install” and select only the services that will be active and executing on each server in the QlikView configuration. To simplify the procedure, it is recommended to have the same Windows Administrator on all servers in the QlikView configuration.

### Enabling SSL

Proceed as follows to enable certificate service authentication for DSC, QVWS, QMC, QDS, and QVS:

1. Stop the QMS service.
2. Run Notepad as administrator.
3. Open the QMS configuration file in Notepad.
4. Change the “UseWinAuthentication” entry from “true” to “false”.
5. Save the file.
6. Start the QMS service.

At this point, you can check if the certificates are properly set on the server that executes the QMS service by running the Microsoft Management Console (MMC) from the Start menu.

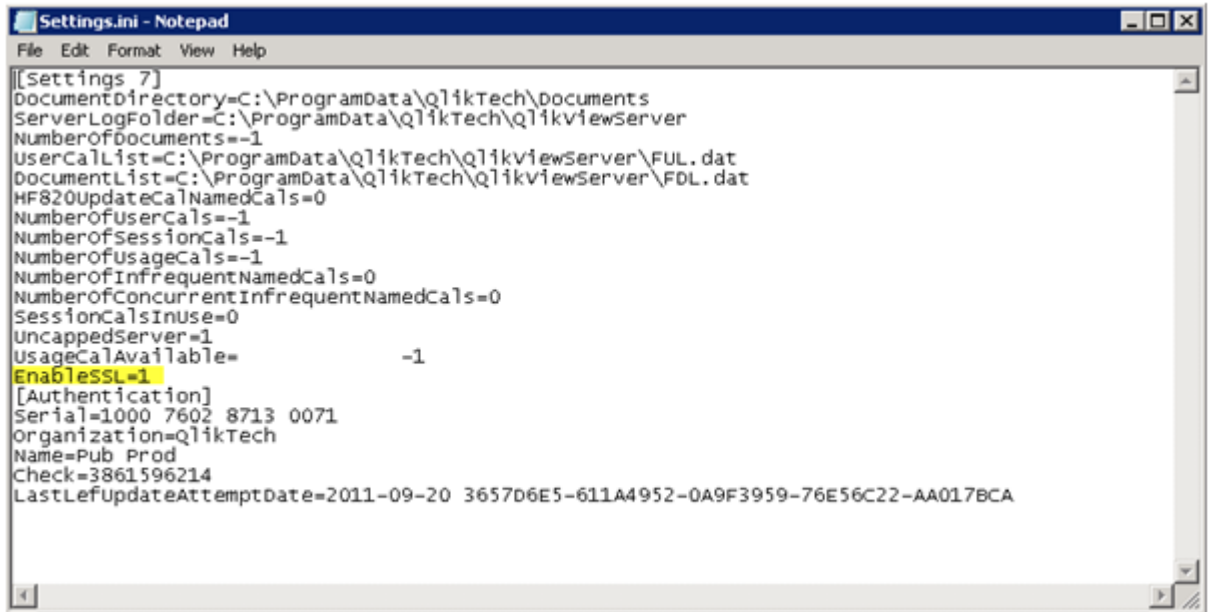
Now repeat the steps above for the DSC, QDS, QVWS and IIS services in your system.

Certificate trust with IIS and QlikView Server is configured using port 4750 (that is, the same port as the QVWS uses). The certificate trust used to enable HTTPS access for users of the web server is also used.

### Enabling SSL for QVS

Proceed as follows to edit the *Settings.ini* file for the QVS service:

1. Stop the QVS service.
2. Run Notepad as administrator.
3. Open the *Settings.ini* file in Notepad.
4. Add `EnableSSL=1` in the [Settings 7] section.

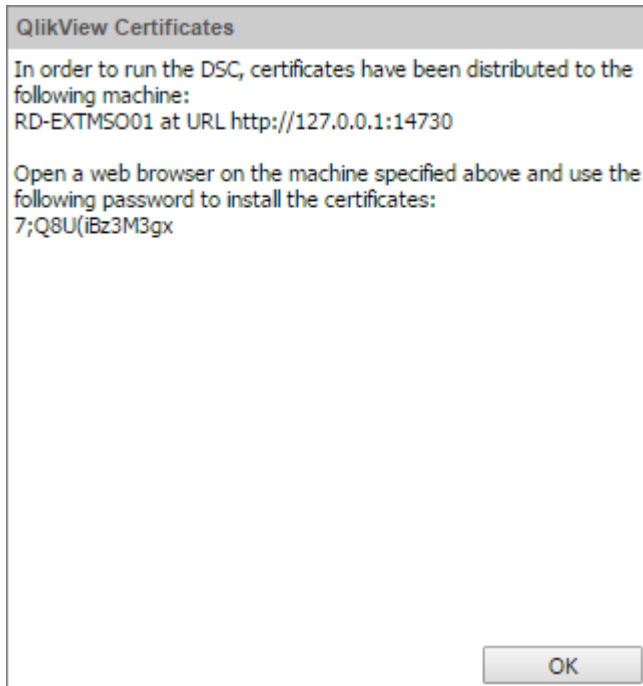


5. Save the file.
6. Start the QVS service.

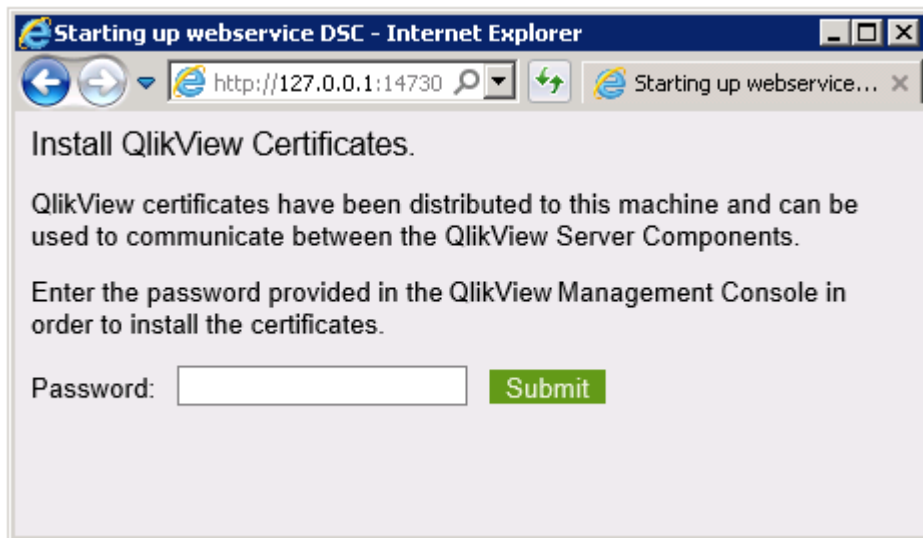
### Adding Services to Issue the Certificates

Proceed as follows to add the services to issue the certificates:

1. Open the QlikView Management Console (QMC).
2. Add each service as a new service and then delete the existing service.
3. When adding a service, a popup window appears.



4. Access the server where the new service resides, either physically or by using a remote desktop connection. Then open a web browser and enter the URL and port provided by the QMC popup window (14720, 14730 or 14750).
5. On the resulting web page, enter the password provided the QMC popup window.



6. If successful, you receive a message that confirms that the password was correct and that the QlikView service at the specific port will unlock.

At this point, you can check to see if the certificates are properly set up on the servers that execute the additional QlikView services by running the MMC from the Start menu.

### Updating Certificates



*Never delete certificates - not even old ones - since they contain encryption keys.*

When the certificates have expired or are about to - or if you want to generate new encryption keys for your sensitive data - a new set of certificates should be generated (and the old ones should be kept).

Perform the following steps on each machine in the cluster:

1. Shut down all QlikView services (in any order).
2. If the machine currently has valid certificates that should be replaced, turn on the configuration flag **InstallingNewCertificatesAndCryptoKey** for all QlikView services.
3. Start up all QlikView services (in any order).
4. In the QMC, click **Apply** for each service (in any order) and perform the instructions that are displayed on the screen.
5. Shut down all QlikView services (in any order).
6. If you turned on the configuration flag **InstallingNewCertificatesAndCryptoKey** in a previous step, now turn it back off for all services.
7. Start up all services (in any order).

At start-up, having new certificates (containing a new encryption key), the services will re-encrypt all their sensitive data with the new encryption key.



*It is strongly recommended not to delete the old certificates (although they now are virtually obsolete), because if you later on need to restore an older backup of your data you will need the previous certificates (with the corresponding encryption key) to decrypt it.*

### Setting InstallingNewCertificatesAndCryptoKey flag

This configuration flag, if turned on (true), ignores the certificates on the machine except for extracting the CryptoAlgorithm from the certificates. The flag is used by DSC, QDS and QVWS, but not by QMS, and is turned off (false) by default.

You need to turn on this flag when updating certificates, to be able to receive new certificates. After the certificates have been updated, you should set it to false for all services.

To turn on the flag, add the following line:

InstallingNewCertificatesAndCryptoKey=True  
to the following configuration files:

*C:\Program Files\QlikView\Distribution Service\QVDistributionService.exe.config*

*C:\Program Files\QlikView\Directory Service Connector\QVDirectoryServiceConnector.exe.config*

*C:\Program Files\QlikView\Server\Web Server\QVWebServer.exe.config*

### Backup

It is vital that the certificates are backed up and kept in a secure location. If the certificates are lost your sensitive data will be lost.

Backup the certificates using the MMC. The three QV certificates on the server running the QMS must be backed up. It is optional to back up the QV certificates on the other servers running QV services – these can be reconstructed by the QMS if lost.

To perform a backup, in the Certificate snap-in in the MMC, right-click the certificate you wish to export and choose the **All Tasks/Export...** context menu option.



*In the export wizard, make sure you export the private key and export all extended properties.*

### Restoring Certificates

If the certificates are missing for any reason, the services will close down and information can be found in the log files. Perform the following steps to restore the certificates:

1. Reinstall the three certificates from a backup.



*A manual restore will only work if you are restoring the certificates on a machine with the same name as from it was backed up. Otherwise you will need to distribute new certificates using the QMC.*

2. Depending on which service has a failure, different actions are needed:

If the QMS service fails, a new set of certificates with a new random SecretsKey is created at startup. The QMS may now be asked for certificates by other services.

If any of the other services fails, the service starts in a special mode where the service can handle certificates from the QMS. You need to browse to a certain port on the local machine and enter a password presented by the QMC. After this, the service must be restarted and will then run in its normal mode, using the newly received certificates and keys.

If there is no backup to use for restoring, the inaccessible data (the protected secret information) has to be cleared and later on reentered.

### Configuration Files

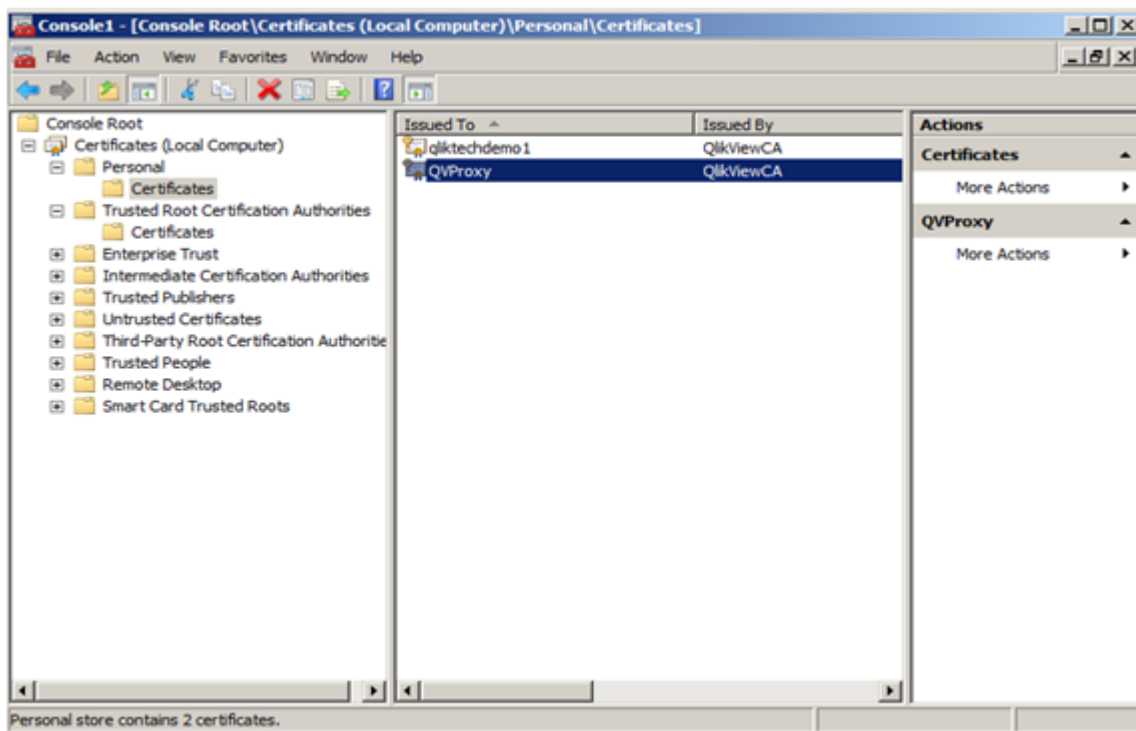
The following table lists the location of each of the configuration files that may need editing.

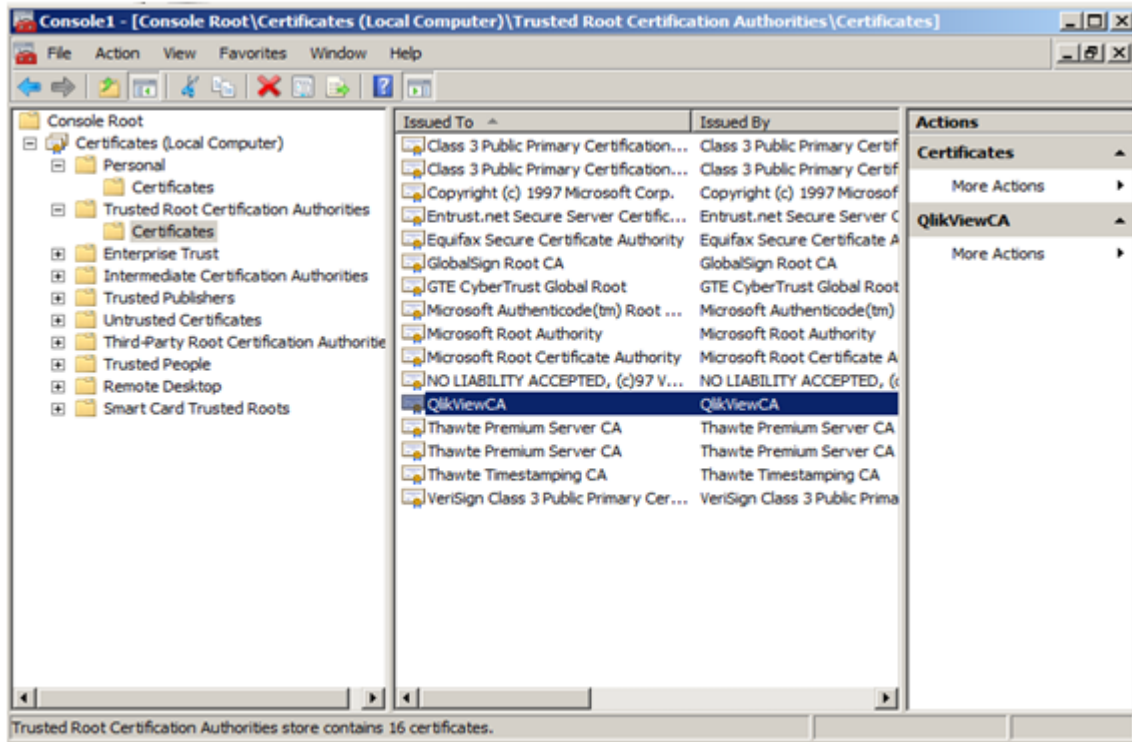
| Service | Default Path   |
|---------|--|
| QMS     | <i>C:\Program Files\QlikView\Management Service\QVManagementService.exe.config</i>                 |
| DSC     | <i>C:\Program Files\QlikView\Directory ServiceConnector\QVDirectoryServiceConnector.exe.config</i> |

|      |  |
|------|--|
| QDS  | <i>C:\Program Files\QlikView\Distribution Service\QVDistributionService.exe.config</i>   |
| QVWS | <i>C:\Program Files\QlikView\Server\Web Server\QVWebServer.exe.config</i>  |
| IIS  | <i>C:\Program Files\QlikView\Server\Web Server<br/>Settings\QVWebServerSettingsService.exe.config</i><br><br><i>C:\Program Files\QlikView\Server\QlikViewClients\QlikViewAjax\web.config</i> |
| QVS  | <i>C:\ProgramData\QlikTech\QlikViewServer\Settings.ini</i>   |

### Using Microsoft Management Console

Certificates can be visually confirmed in the MMC with the certificate snap-in added. The QlikView certificates are located in the **Personal**>**Certificates** and **Trusted Root Certification Authorities**>**Certificates** folders:





The figures above show properly installed certificates in a QlikView Server configuration. Within the MMC, all QlikView services on servers have certificates deployed as shown in the figures.

The following certificates are generated by QlikView and need to be backed-up:

| Location                      | Issued To               | Issued By  |
|-------------------------------|-------------------------|------------|
| Local Computer / Personal     | <i>the-machine-name</i> | QlikViewCA |
| Local Computer / Personal     | QVProxy                 | QlikViewCA |
| Local Computer / Trusted Root | QlikViewCA              | QlikViewCA |

## 2.4 Logs and error codes

All alerts from the QlikView services appear in the Windows event log.

### Logging from QlikView Server

Detailed session logs are found in the logging directory, which is specified on the **System>Setup>Logging** tab in QlikView Management Console (QMC). The default location is *%ProgramData%\QlikTech\QlikViewServer*.

Log files can be set to split (that is, create new) daily, weekly, monthly, yearly, or never. Performance log intervals can be set from one minute and higher.



*Setting the interval to be very small, for example, only one minute, may negatively impact the performance.*


### Session Log

A session is defined as a single user connected to a single document.



*The session log is updated each time a session ends. This means no log entry is created when a session starts.*

The file name of the session log is *Sessions\*.log*, where \* reflects the server name and the split interval. Each entry of the session log contains the fields listed below.

| Field              | Description   |
|--------------------|---|
| Exe Type           | Type of QVS build.<br><br>Example: "RLS64" = 64-bit release build   |
| Exe Version        | Full version number of QVS.<br><br>Example: "11.00.11076.0409.10"   |
| Server Started     | Date and time when QVS was started.   |
| Timestamp          | Date and time when the log entry was created.   |
| Document           | QlikView document that was accessed.  |
| Document Timestamp | File timestamp of the document that was accessed.   |
| QlikView User      | QlikView section access user ID (if used).  |
| Exit Reason        | Reason for session termination: <ul style="list-style-type: none"><li>• "Socket closed" = Client-induced termination</li><li>• "LRU" = Terminated as Least Recently Used in favor of new user</li><li>• "Shutdown" = Server-induced termination for other reasons</li></ul> <div> <i>This is not a complete list, as the exit value in some cases comes from the operating system.</i></div> |
| Session Start      | Time when the session was started.  |
| Session Duration   | Duration of session in hours:minutes:seconds.   |
| CPU Spent (s)      | CPU seconds spent by the session.   |



## 2 Planning QlikView Deployments

| Field                         | Description   |
|-------------------------------|---|
| Bytes Received                | Bytes received by the server during the session.  |
| Bytes Sent                    | Bytes sent by the server during the session.  |
| Calls                         | Number of QlikView calls during the session (bidirectional).  |
| Selections                    | Number of QlikView selections made during the session.  |
| Authenticated User            | Authenticated Windows NT® user ID (if any).   |
| Identifying User              | Client user identification.   |
| Client Machine Identification | <p>The client machine identification.</p> <p>By default, this is the universally unique identifier (UUID) receiver from the call to the Windows Management Instrumentation (WMI).</p> <p>If the UUID is unavailable, one of the following IDs may display instead:</p> <ul style="list-style-type: none"><li>• MAC address of the computer</li><li>• Computer name</li><li>• Unique machine ID (if the browser used in the session was in a private mode)</li></ul> |
| Serial Number                 | Serial number of the QlikView client (installed clients only, that is, QlikView Desktop and Internet Explorer plugin).  |
| Client Type                   | <p>Client type used:</p> <ul style="list-style-type: none"><li>• “Windows Exe” = QlikView Desktop and Internet Explorer plugin</li><li>• “Ajax” = all clients that use the QVPX protocol</li><li>• “Unknown”</li></ul>  |
| Client Build Version          | Build version of the QlikView client.   |
| Secure Protocol               | <p>Secure protocol used:</p> <ul style="list-style-type: none"><li>• “On” when encrypted communication is used (typically Windows clients).</li><li>• “Off” when non-encrypted communication is used.</li></ul>   |
| Tunnel Protocol               | “Tunnel” when QVS tunnel communication is used.   |
| Server Port                   | Port used by the server.  |

| Field           | Description   |
|-----------------|---|
| Client Address  | Client IP number for the client that is connected to the server (through the port specified in the Server Port field above).  |
| Client Port     | Client port.  |
| CAL Type        | Client Access License (CAL) type: <ul style="list-style-type: none"><li>• “User” = Named User CAL</li><li>• “Session” = Session CAL</li><li>• “Usage” = Usage CAL</li><li>• “Document” = Document CAL</li></ul> |
| CAL Usage Count | Number of Usage CALs.   |


### Performance Log

The performance log is updated at the interval specified on the **System>Setup>Logging** tab in QMC. The default interval is five minutes. Additional entries are added whenever the server is started or stopped. The file name of the session log is *Performance\*.log*, where \* reflects the server name and the split interval.




Each entry of the log contains the fields listed below.

| Field             | Description  |
|-------------------|--|
| Exe Type          | Type of QVS build.<br><br>Example: “RLS64” = 64-bit release build  |
| Exe Version       | Full version number of QVS.<br><br>Example: “11.00.11076.0409.10”  |
| Server Started    | Date and time when QVS was started.  |
| Timestamp         | Date and time when the log entry was created.  |
| EntryType         | Entry type: <ul style="list-style-type: none"><li>• “Server starting” = Startup</li><li>• “Normal” = Normal interval log entry</li><li>• “Server shutting down” = Shutdown</li></ul> |
| ActiveDocSessions | Number of document sessions* that has shown activity during the interval and still exists at the end of the interval.  |

## 2 Planning QlikView Deployments

| Field                      | Description  |
|----------------------------|--|
| DocSessions                | Total number of document sessions* that exists at the end of the interval.   |
| ActiveAnonymousDocSessions | Number of document sessions* with anonymous user that has shown activity during the interval and still exists at the end of the interval.  |
| AnonymousDocSessions       | Total number of document sessions* with anonymous user that exists at the end of the interval.   |
| ActiveTunneledDocSessions  | Number of document sessions* with tunneled connection that has shown activity during the interval and still exists at the end of the interval.   |
| TunneledDocSessions        | Total number of document sessions* with tunneled connection that exists at the end of the interval.  |
| DocSessionStarts           | Number of document sessions* that has been initiated during the interval.  |
| ActiveDocs                 | Number of documents loaded at the end of the interval in which there has been user activity during the interval.   |
| RefDocs                    | Number of documents loaded at the end of the interval for which there is a session at the end of the interval.   |
| LoadedDocs                 | Total number of documents loaded at the end of the interval.   |
| DocLoads                   | Number of new documents loaded during the interval.  |
| DocLoadFails               | Number of documents that has failed to load during the interval.   |
| Calls                      | Total number of calls to QVS during the interval.  |
| Selections                 | Number of selection calls during the interval.   |
| ActiveIpAddr               | Number of distinct IP addresses that has been active during the interval and still exists at the end of the interval. <div> <i>Tunneled sessions and multiple users originating from the same IP cannot be distinguished.</i></div> |

## 2 Planning QlikView Deployments

| Field                  | Description   |
|------------------------|---|
| IpAddrs                | <p>Total number of distinct IP addresses connected at the end of the interval.</p> <div> <i>Tunneled sessions and multiple users originating from the same IP cannot be distinguished.</i></div> |
| ActiveUsers            | <p>Number of distinct NT users that has been active during the interval and still exists at the end of the interval.</p> <div> <i>Anonymous users cannot be distinguished.</i></div>             |
| Users                  | <p>Total number of distinct NT users connected at the end of the interval.</p> <div> <i>Anonymous users cannot be distinguished.</i></div>   |
| CPUload                | <p>Average CPU load from QVS during the interval.</p>   |
| VMAllocated(MB)        | <p>Size in MB of the virtual memory allocated by QVS at the end of the interval**.</p>  |
| VMCommitted(MB)        | <p>Size in MB of the virtual memory actually used by QVS at the end of the interval. This number is part of VMAllocated(MB) and should not exceed the size of the physical memory in order to avoid unacceptable response times.</p>  |
| VMFree(MB)             | <p>Size in MB of the unallocated virtual memory available to QVS**.</p>   |
| VMLargestFreeBlock(MB) | <p>Size in MB of the largest contiguous block of unallocated virtual memory available to QVS. This number is part of VMFree(MB).</p>  |
| UsageCalBalance        | <p>“-1.00” = There are no Usage CALs.</p>   |
| CacheHits              | <p>Number of generic cache hits</p>   |
| CacheLookups           | <p>Number of generic cache lookups</p>  |
| CacheObjectAdded       | <p>Number of objects added to the generic cache</p>   |
| CacheBytesAdded        | <p>Number of bytes added to the generic cache</p>   |
| CacheTimeAdded         | <p>Time spent adding new objects to the generic cache</p>   |

| Field         | Description                                     |
|---------------|---|
| CacheReplaced | Number of objects replaced in the generic cache |

\*One user + one document = One document session.

\*\*VMAllocated(MB) + VMFree(MB) = Total maximum virtual memory space available to the QVS process.

### Event Log

The event log is updated each time a log entry is made in the Windows event log by QVS. The stored information is a mirror of the information written to the Windows event log. The file name of the event log is *Events\*.log*, where \* reflects the server name and the split interval.

Use the **Event Log Verbosity** radio buttons on the **System>Setup>QlikView Servers>Logging** tab in the QMC to set the verbosity level. Depending on the verbosity level selected, the following entries are written to the Event log:

- **Low**: Error messages
- **Medium**: Error and warning messages
- **High**: Error, warning, and information messages

Each entry of the log contains the fields listed below.

| Field          | Description  |
|----------------|--|
| Server Started | Date and time when QVS was started.  |
| Timestamp      | Date and time when the log entry was created.  |
| SeverityID     | ID for the severity level: <ul style="list-style-type: none"><li>• 1 = Error</li><li>• 2 = Warning</li><li>• 4 = Information</li></ul> |
| EventID        | Unique ID for the event type.  |
| Severity       | Event severity level: <ul style="list-style-type: none"><li>• Error</li><li>• Information</li><li>• Warning</li></ul>                  |
| Message        | Event description.   |

### End-user Audit Log

The end-user audit log contains information on user selections, including cleared selections, activated sheets, application of bookmarks, accessed reports, and maximized objects.

A log file called *AUDIT\_<machinename>* is saved to *%ProgramData%\QlikTech\QlikViewServer*.



*Tick the **Enable Extensive Audit Logging** check box on the **System>Setup>QlikView Servers>Logging** tab in the QMC to enable detailed audit logging (for example, logging of all selections that come with a bookmark). However, the logging of user selections in QVS is based on how the current selections object works and therefore larger selections may not be logged in detail.*

| Field          | Description   |
|----------------|---|
| Server started | Date and time when QVS was started.   |
| Timestamp      | Date and time when the log entry was created.   |
| Document       | Path and name of the document that was accessed.  |
| Type           | Type of selection made (for example, "Selection" or "Bookmark").<br><br>For an overview of the types available, see the table below.  |
| User           | User name.  |
| Message        | Information on the type of selection or application of bookmark that was made in the document (for example, "Apply Server\Bookmark15").<br><br>For an overview of the messages that can be posted in this field, see the table below. |

The types and messages that can be posted in the Type and Message fields in the end-user audit log are listed below.



*In the end-user audit log, "XXX" and "YYY" are replaced with values from the QlikView document.*

| Type | Message | Description |
|------|---------|-------------|
|------|---------|-------------|

## 2 Planning QlikView Deployments

| Action                                | action (#) [XXX] | <p>Action # was executed with XXX. The numeric value corresponds to one of the following actions:</p> <ul style="list-style-type: none"> <li>• Info = 0</li> <li>• Lock All = 2</li> <li>• Unlock All = 3</li> <li>• Clear All = 4</li> <li>• Clear All Including Locked = 5</li> <li>• Back = 6</li> <li>• Forward = 7</li> <li>• File Close = 8</li> <li>• Next Tab = 9</li> <li>• Previous Tab = 10</li> <li>• Export = 11</li> <li>• Launch = 12</li> <li>• Macro = 13</li> <li>• Recall Bookmark = 14</li> <li>• Replace Bookmark = 15</li> <li>• Create Bookmark = 16</li> <li>• Print Report = 17</li> <li>• Activate Sheet = 18</li> <li>• Print Sheet = 19</li> <li>• Print Object = 20</li> <li>• Restore Object = 21</li> <li>• Minimize Object = 22</li> <li>• Maximize Object = 23</li> <li>• Activate Object = 24</li> <li>• Select Excluded = 25</li> <li>• Clear Other Fields = 26</li> <li>• Select Possible = 27</li> <li>• Lock = 28</li> <li>• Unlock = 29</li> <li>• Pareto Select = 30</li> <li>• Set Value = 31</li> <li>• Field Select = 32</li> <li>• Field Toggle Select = 33</li> <li>• Open URL = 34</li> <li>• Document Chain = 35</li> <li>• Clear Field = 36</li> <li>• Reload = 37</li> <li>• Set state = 38</li> <li>• Transfer state = 39</li> <li>• Swap state = 40</li> <li>• Dynamic update = 41</li> </ul> |
|---------------------------------------|------------------|--|
| Deploying QlikView - QlikView 12, x.y |                  |  |

## 2 Planning QlikView Deployments

|                       |   |   |
|-----------------------|---|---|
| Bookmark              | Apply XXX   | Bookmark XXX was applied.   |
| Bookmark Selection    | XXX   | Selection XXX was made because a bookmark was selected. Entries of this type are only logged when detailed audit logging is selected. |
| Document              | Document XXX  | Document XXX was opened or closed.  |
| Export                | Sheet Object XXX                                      | Sheet object XXX was exported.  |
| Maximize              | Sheet Object XXX                                      | Sheet object XXX was maximized.   |
| Minimize              | Sheet Object XXX                                      | Sheet object XXX was minimized.   |
| Print                 | Sheet Object XXX                                      | Sheet object XXX was printed.   |
| Report                | Accessed report XXX                                   | Report XXX was accessed.  |
| Restore               | Sheet Object XXX                                      | Sheet object XXX was restored.  |
| Selection             | Clear All   | All selections were cleared.  |
| Selection             | XXX   | Selection XXX was made.   |
| SendToExcel           | Sheet Object XXX                                      | Sheet object XXX was sent to Microsoft Excel.   |
| Sheet Object          | Sheet Object XXX                                      | Various activities that can apply to Sheet object XXX.  |
| Session Collaboration | Session Collaboration Initiated, ID:XXX               | A session collaboration with ID XXX was initiated.  |
| Session Collaboration | Session Collaboration user XXX joined session, ID:YYY | User XXX joined the session collaboration with ID YYY.  |
| Session Collaboration | Session Collaboration user XXX left session, ID:YYY   | User XXX left the session collaboration with ID YYY.  |

The following example shows the resulting log entry when a bookmark ("Bookmark01") is selected. The log has been put in a table for better overview.

| Field          | Value                                      |
|----------------|--|
| Server started | 20130506T101733.000+0900                   |
| Timestamp      | 20130506T102328.000+0900                   |
| Document       | C:\ProgramData\QlikTech\Documents\Test.qvw |
| Type           | Bookmark                                   |
| User           | QlikTech\jsmith                            |
| Message        | Apply Server\Bookmark01                    |



If detailed audit logging is selected, the log entry above may be followed by one or more log entries that detail the selections that were made because the bookmark was selected. In these log entries, the Type field is set to "Bookmark Selection".

### Manager Audit Log

The audit logging provides the possibility to track changes to tasks and settings in the system in order to see who made the changes and when they were made.

The audit logs are stored in *%ProgramData%\QlikTech\ManagementService\AuditLog*. One folder per table is created. Each folder contains one file per day with the changes made to the tasks. The logs are tab separated files.

The entries found in the logs are listed below.

| Entry          | Description   |
|----------------|---|
| TransactionID  | Transaction ID, which is useful for keeping track of changes made simultaneously.   |
| ChangeType     | Type of operation, update (new or changed entries) or delete (entries have been deleted).   |
| ModifiedTime   | Time and date (in UTC) when the changes were made.  |
| ModifiedByUser | The user that made the changes in the user interface. system means that the change was initiated by the system and not by any user. |
| ID             | ID of the row (that was updated or deleted) in the table that was changed.  |

The following example comes from the `AlertEmail` table. The log has been put in a table for better overview.

|                       |                                      |
|-----------------------|--------------------------------------|
| TransactionID         | 455a241d-8428-4dc7-ba67-4ae7cb21cf3d |
| ChangeType            | Update                               |
| ModifiedTime          | 20100202T151254.000+0900             |
| ModifiedByUser        | MyDomain\mjn                         |
| ID                    | b3745325-cee7-4fe7-b681-9c9efe22fc5c |
| DistributionServiceID | 8846d7dd-bb3f-4289-9c9b-b0ca71b7c3b2 |
| EmailAddress          | mjn                                  |

The following example comes from the `QDSCluster` table. Note that `TransactionID` is the same for both examples. This means that the changes were made simultaneously.

|               |                                      |
|---------------|--------------------------------------|
| TransactionID | 455a241d-8428-4dc7-ba67-4ae7cb21cf3d |
| ChangeType    | Update                               |
| ModifiedTime  | 20100202T151254.000+0900             |

|                       |                                      |
|-----------------------|--------------------------------------|
| ModifiedByUser        | MyDomain\mjn                         |
| ID                    | a37f242c-6d80-42da-a10c-1742d2ec927f |
| DistributionServiceID | 8846d7dd-bb3f-4289-9c9b-b0ca71b7c3b2 |
| QDSWebAddress         | http://computer-mjn:4720/qtxs.asmx   |
| CurrentWorkorderID    | 96bff2dc-f1ea-84d2-b6c4-ea58bf5c98e5 |

### Task Performance Summary

The task performance summary is used to log task performance information.

Proceed as follows to activate the task performance summary:

1. Open the *Settings.ini* file in a text editor. The default location of the file is:  
`C:\Windows\system32\config\systemprofile\AppData\Roaming\QlikTech\QlikViewBatch`
2. Locate the following section in the *Settings.ini* file:  

```
[Settings 7]

InterfaceLanguage=English

InstalledLIBID110={4D121C39-415E-11D1-934D-0040333C91CC}
```
3. Add `EnableQVBProcessSummary=1` at the end of the section to activate the task performance summary.



*The last row in the Settings.ini file must be empty.*

4. Save the *Settings.ini* file.
5. Restart the QlikView Distribution Service (QDS).

Once the QDS has restarted, the task log is updated. Example of task output:

Name: qvb.exe, PID: 1360, Peak CPU: 50,0%, Peak Physical RAM: 26.00 Mb, Peak Virtual RAM: 21.69 Mb, Average CPU: 1,0%, Average Physical RAM: 24.47 Mb, Average Virtual RAM: 20.37 Mb, Peak Total CPU: 58,3%, Peak Total Physical RAM: 6143.49 Mb, Peak Total Virtual RAM: 12285.17 Mb, Elapsed Time: 00:00:36.4692722

## 2.5 Licensing

QlikView Server comes in a number of editions designed for different organizations and purposes. Upgrading is done through the license key.

To connect to a QlikView Server (QVS), each client needs a Client Access License (CAL). The CALs are purchased with QlikView Server and associated with the server serial number.



*CALs are used for licensing only and they have nothing to do with user authentication for data access purposes.*

### OEM

#### General

The OEM feature prevents abuse of QlikView Servers sold under an Original Equipment Manufacturer (OEM) license and protects the revenue streams of both the OEM products and the full QlikView product. In addition, the feature helps avoid channel conflicts between QlikView OEM partners, QlikView reseller partners, and QlikView direct account managers.

The OEM feature includes the following restrictions:

- A QlikView Server delivered to a customer by an OEM partner cannot run other QlikView applications than the ones delivered by the OEM partner.
- A QlikView application delivered to a customer by an OEM partner cannot run on another QlikView Server than the one delivered by the OEM partner.

#### Detailed Function Description

The functions of the OEM feature are as follows:

A tag with a key is defined in the QlikView Server License Enabler File (LEF) as `OEM_PRODUCT_ID`. This LEF tag is issued once for each OEM partner and their QlikView Desktop, and QlikView Server licenses with matching tags are delivered for each QlikView Server deployment requiring this feature.

The User Preferences dialog in QlikView Desktop allows an OEM developer to embed a hash key in the `.qvw` file. The hash key, which is based on the `OEM_PRODUCT_ID` key present in the QlikView Desktop license of the OEM partner, is a capitalized 40 character hex string that is stored in the Document Properties and Document metadata. In the dialog, the partner can label all keys generated for the `.qvw` files. The same key can be used for multiple documents belonging to the same customer.

A QlikView Server with the `OEM_PRODUCT_ID` tag in its LEF only permits the publishing of `.qvw` files with a matching key on that QlikView Server. A standard, non-OEM QlikView Server by default opens any `.qvw` file, with the exception of `.qvw` files containing a specific key that some OEM partners are issued with to prevent opening with any other QlikView Server than the one with a matching `OEM_PRODUCT_ID`.

The table below provides a few examples of the results of the OEM functionality.

|                 |                          | File              |                  |                  |
|-----------------|--------------------------|-------------------|------------------|------------------|
|                 |                          | <i>Normal.qvw</i> | <i>OEM 1.qvw</i> | <i>OEM 2.qvw</i> |
| QlikView Server | Normal QlikView Server   | File opened       | File not opened  | File not opened  |
|                 | OEM 1 (No license lease) | File not opened   | File opened      | File not opened  |
|                 | OEM 2 (No license lease) | File not opened   | File not opened  | File opened      |

In QlikView Desktop, a .qvw file containing a `PRODUCT_ID` is opened in user mode.

## 3 Installing QlikView

This section gives information on how to install QlikView. It also describes some maintenance tasks, such as how to update, repair or modify the installation.

### 3.1 Installing QlikView Server

#### Before Installing QlikView Server

Before installing QlikView Server, you need to consider:

- If Microsoft IIS is to be used as web server, it must be installed prior to QlikView Server.
- It is not possible to install QlikView Server to a server that acts as a domain controller.
- IPv4 is required for installation of QlikView Server. IPv6 is currently unsupported.
- When installing QlikView Server/Publisher, several security groups are created. Several other security groups must be created following the installation. These must be properly configured to ensure that the appropriate services can run, and to ensure that users can access the appropriate functionality. Before you begin the installation, see **Security Groups** in *QlikView Publisher Repository* (page 26).
- It is recommended not to move folder locations after the QlikView Server installation is complete, since many settings depend on the initial file locations. If the location of QlikView Server has to be changed after the installation, uninstall QlikView Server and then reinstall.
- Any previously defined tasks are deleted when the QlikView Publisher license is activated.

#### Setup Procedure

1. Run the QlikView Server installation executable:
  - Microsoft Windows x64 version: *QlikViewServer\_x64Setup.exe*
2. If the User Account Control dialog is displayed, click **Yes** to allow the program to make changes on this computer.
3. Click **Next** in the Welcome dialog.
4. Select the region for the location of the server. Click **Next** to continue.
5. Read the license agreement, select **I accept the terms in the license agreement**, and click **Next** to continue.
6. Enter the customer information for QlikView Server. Click **Next** to continue.
7. All files are installed in the specified folder. To change the root folder for the installed files, click **Change** to specify the preferred location. Finally, click **Next** to continue.
8. Select the type of installation you want to perform:
  - **Full installation, Single machine with QlikView Webserver:** Used to run all components on a single machine with QlikView Web Server as web server.
  - **Full installation, Single machine with Microsoft IIS:** Used to run all components on a single machine with Microsoft IIS as web server. This option is only available if IIS is installed on the target machine.

- **Custom installation, select profiles:** If this option is selected you select the profiles you want to be included in the installation from the Profiles section in the dialog:
  - **QlikView Server:** Installs QlikView Server, Directory Service Connector, and the QlikView Server example documents.
  - **Reload/ Distribute Engine:** Installs the Reload Engine and the QlikView Distribution Service.
  - **Management Console:** Installs the QlikView Management Service together with the QlikView Management Console (QMC).
  - **Webserver:** Installs the QlikView Web Server.

To make further configuration of features to be installed, click **Config**. When done, click **Next**.

To use pre-defined configuration of features, click **Next**.

9. Set the account that the QlikView Server and Publisher services are to run under. Click **Next** to continue.



*The account that is used to run the QlikView services must have local administrator privileges.*

You can also select **I want to specify the account to be used for the services later**.

10. Select the IIS Website from the drop-down list and click **Next**.



*This step is only applicable if **Full installation, Single machine with Microsoft IIS** was selected in **Step 8**. If not, proceed directly to the next step.*

11. Select the Service Authentication method:
  - **Use digital certificates:** Authenticate communication between QlikView servers using digital certificates and SSL. This alternative is recommended in environments where not all servers have access to a common Windows Active Directory or when the security provided by certificate authentication is required. Note that digital certificates are **only** supported by Windows Server 2008 R2 and later.
  - **Use QlikView Administrators Group:** Authenticate communication between QlikView services based on membership in the local Windows group QlikViewAdministrators. This alternative can be used in environments where all servers that are part of the QlikView installation can authenticate using a common Windows Active Directory.

Click **Next** to continue.

12. Click **Install** to start the installation.



*This may take several minutes to complete.*

13. Click **Finish** when the installation is complete.
14. Log off from Windows® and then log on again, so that group memberships added during the

installation are updated.



*It may be sufficient to log off from Windows and then log on again. However, it is recommended to restart the machine to enable the QlikView Server functionality.*

### Logging the Installation

The setup procedure is logged when running the QlikView Server installation executable. The log files are as follows:

- Microsoft Windows x64 version: *QlikViewServerx64.wil*

The log files are stored in the *Temp* folder of the user (for example, *%UserProfile%\AppData\Local\Temp*). Each time an installation is executed, a new file is generated, over-writing the previous log file.

### Obtaining the MSI package

If the MSI package is needed for the installation, proceed as follows to extract it from the .exe file:

1. Start the installation from the .exe file and wait until the first dialog opens.
2. Locate the MSI file (often stored with a random name, for example, *ed34g.msi*) in the *Temp* folder in *%UserProfile%\AppData\Local*.
3. Copy the .msi file to another location.
4. Exit the .exe installation.
5. Install QlikView Server using the .msi file.

### Completing the Installation

After successfully installing QlikView Server, a license must be registered in QlikView Management Console (QMC) to activate the installed software.



*If access is denied when starting QMC, log off from Windows and then log on again, so that group memberships added during the installation are updated.*



*Running real-time anti-virus protection on the server degrades the performance of QlikView Server. It is recommended that the user documents, source documents, log directories, and .pgo files are excluded from the anti-virus scanning.*

### Running Microsoft IIS

#### Handling Timeouts



*This is only needed when using very large QlikView documents that return timeouts.*

Proceed as follows to handle timeouts:

1. Open the `%ProgramFiles%\QlikView\Server\QlikViewClients\QlikViewAjax\web.config` file in a text editor (for example, Notepad).
2. Search for the following text:  
`<httpRuntime requestValidationMode="2.0" />`
3. Edit the text so that it becomes:  
`<httpRuntime requestValidationMode="2.0" executionTimeout="900"/>`
4. Save the file.

### Enabling ASP.NET

If Microsoft IIS is used as web server, enable ASP.NET to ensure proper operation of the QlikView Server sample pages and the extended functions (for example, QlikView Server tunnel).

### Optimizing the Performance

To optimize the performance when running Microsoft IIS and AJAX, turn on compression in the web server.

For information on how to configure IIS http compression:

 <https://www.iis.net/configreference/system.webserver/httpcompression>

### Licensing

The licensing is used to authenticate QlikView Server and allow it to run on a specific machine.

Proceed as follows to enter the license for QlikView Server:

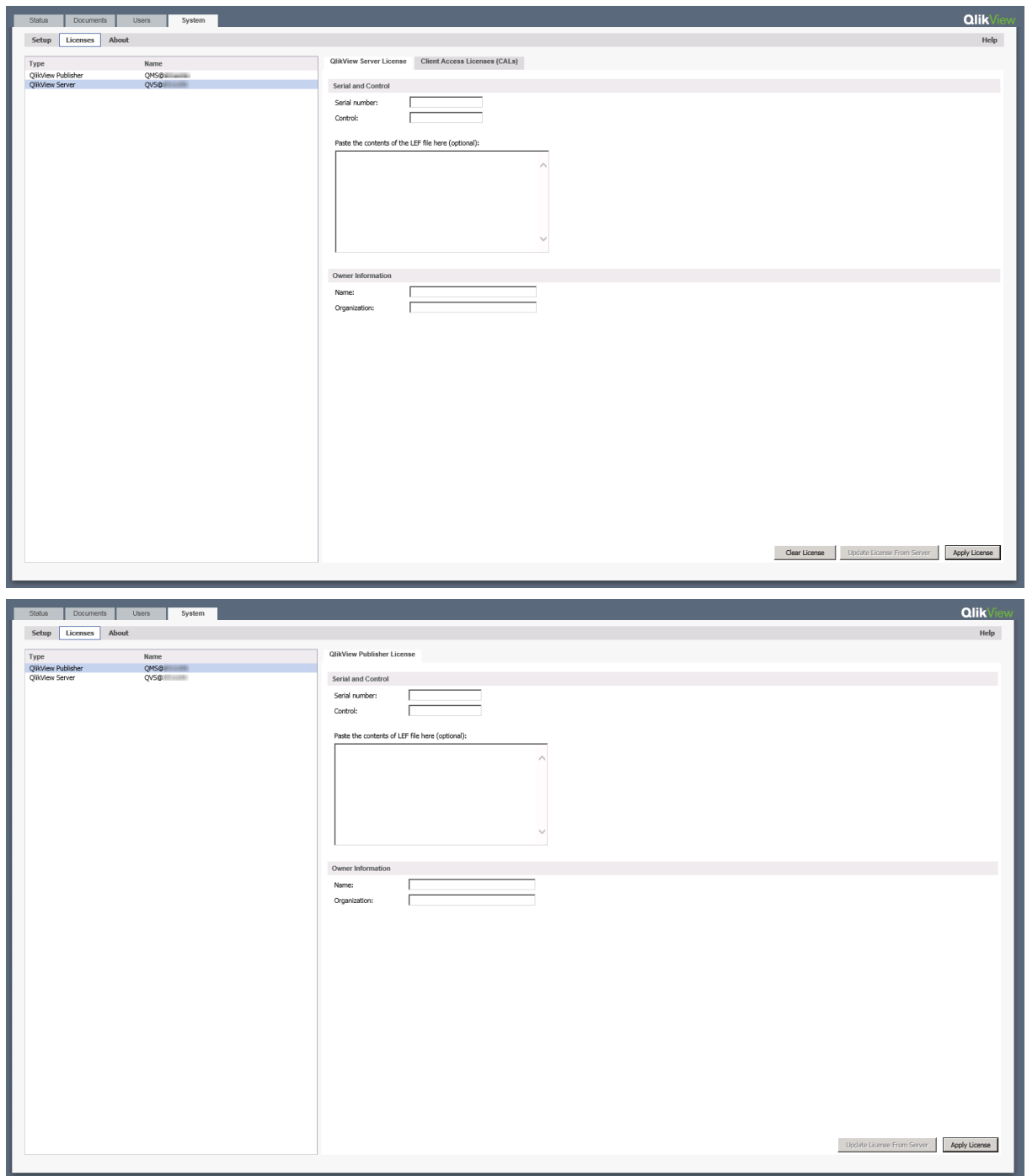
1. Go to **System>Licenses** in the QMC.
2. Select a QlikView Server or Publisher.
3. Fill in the **Serial number** and **Control** fields on the **QlikView Server License** or **QlikView Publisher License** tab (depending on whether QlikView Server or Publisher was chosen).



*Any previously defined tasks are deleted when the QlikView Publisher license is activated.*



### 3 Installing QlikView



*QlikView Server/Publisher License tab in QMC*

The license is checked every time a document is opened. If the time limit specified by the License Enabler File (LEF) is reached, the QVS automatically enters offline mode, which means that it is reachable from the QMC, but not operational.

The License Enabler File (LEF), *lef.txt*, for QlikView Server is automatically saved in *%ProgramData%\QlikTech*.

The *PubLef.txt* file for QlikView Publisher is saved in  
*%ProgramData%\QlikTech\ManagementService\Publisher LEF*.

Click **Update License from Server** to download a new *lef.txt* file from the QlikView LEF server. This is primarily used when updating the number of Client Access Licenses (CALs).

If the LEF information cannot be accessed through the Internet, it can be obtained from the local vendor. In that case, copy the entire *lef.txt* file to the location mentioned above, or paste the LEF data using the corresponding field on the QlikView Server/Publisher License tab in QMC. Contact the local vendor for specific instructions.

### 3.2 Silent Installation

When running a silent installation, QlikView is installed with a limited set of or no dialogs at all. This means all features, properties, and user selections have to be known when creating the silent installation package. There are also some standard properties in Windows Installer Service that may be required.

To prepare a silent installation, the MSI file has to be extracted from the QlikView *Setup.exe* file.

A silent installation can be run with different interface levels:

|            |                       |
|------------|-----------------------|
| <i>/qn</i> | Completely silent.    |
| <i>/qb</i> | Basic user interface. |

Add a + sign at end of the interface levels command to get a modal dialog at the end of the installation saying "Finished" and if it was successful or not.

The following silent installation command lines are recommended for QlikView:

```
msiexec /i QlikViewServerx64.msi Addlocal="all" IS_NET_API_LOGON_
USERNAME="Domain\username" IS_NET_API_LOGON_PASSWORD="password /qn+
```

Alternatively:

```
QlikViewServer_x64Setup.exe /s /v"/qn+ Addlocal="all" IS_NET_API_LOGON_
USERNAME="Domain\username" IS_NET_API_LOGON_PASSWORD="password"
```

The command line above installs all features completely silently with a modal dialog at the end of the installation.

If just a limited set of the features are to be installed, change *all* to the name of the feature instead. If several features are to be installed, separate them with commas.

The following features can be installed:

- DirectoryServiceConnector
- ManagementService
- QVS

- QvsDocs
- WebServer
- DistributionService
- SupportTools
- QvsClients with the sub-features Plugin and AjaxZfc
- MslIS with the sub-features QvTunnel and QlikView Settings Service



*For the sub-features to be included in the installation, they have to be included in the list of features to be installed.*

```
msiexec /i QlikViewServerx64.msi ADDLOCAL="all" DEFAULTWEBSITE="2" /qn+
```

This command line installs all features, including the virtual directories to another website than the default one. This requires a machine with Microsoft Internet Information Services (IIS) installed and more than one website on it. The site number also has to be known. Set *DEFAULTWEBSITE* to the site number where the virtual directories are to be installed. To find the number of the website, check IIS.

The installation procedure can be logged, using the following command:

```
msiexec /i QlikViewServerx64.msi ADDLOCAL="all" DEFAULTWEBSITE="2"/L*v log.txt /qn+
```

## Settings

The following settings are good to know when designing a silent installation package:

|   |  |
|---|--|
| <b>Prerequisites</b>                            | .NET Framework 4.5   |
| <b>Default installation folder (INSTALLDIR)</b> | ProgramFilesFolder\QlikView                                    |
| <b>Windows Installer Version</b>                | 3.1 Schema 301   |
| <b>Default language</b>                         | English (United States) 1033                                   |
| <b>Require Administrative Privileges</b>        | Yes  |
| <b>INSTALLEVEL</b>                              | 100, all features is set to 101 by default                     |
| <b>Features</b>                                 | There is a hidden feature called "Install". Do not remove it.  |
| <b>IIS</b>                                      | Four virtual directories and an Application pool are installed |
| <b>Services</b>                                 | Five services are installed                                    |

## Dialogs

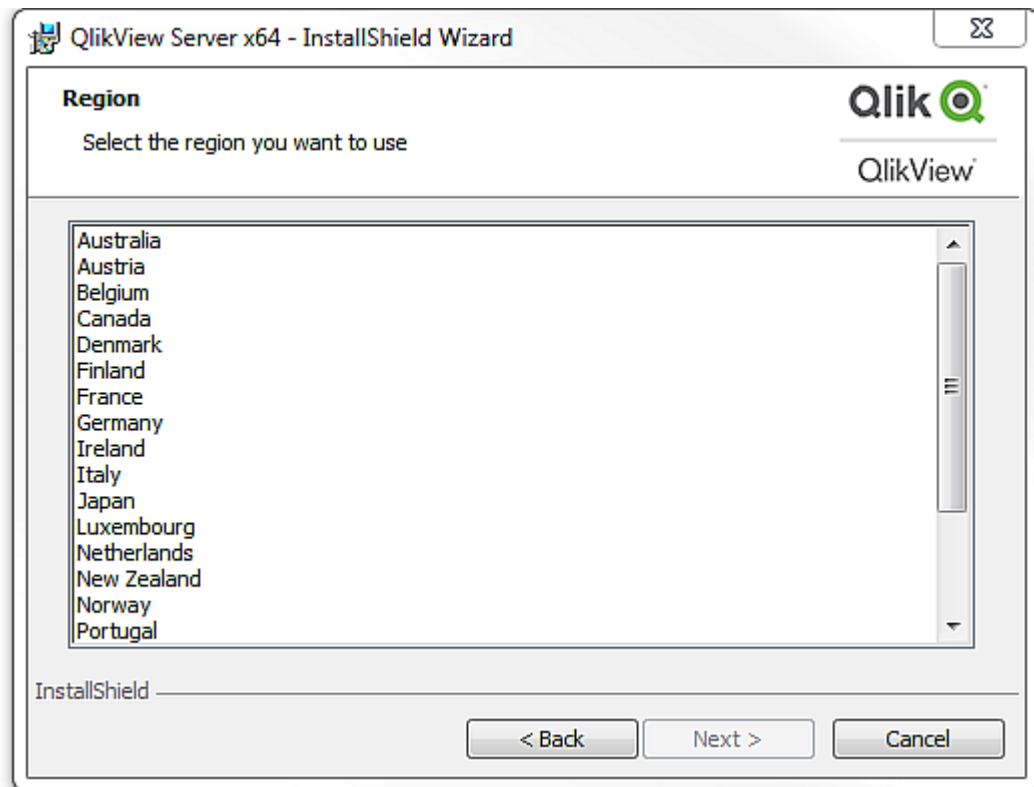
The QlikView installation has a number of dialogs, one of which is a Custom Setup dialog and one of which is a Website dialog. All dialogs set important properties. To find the value of a property, do a test installation with verbose logging. Note that the property values may differ depending on the language and operating

system used.

## Region

This dialog is used for specifying the region.

Property: *REGION\_LIST*

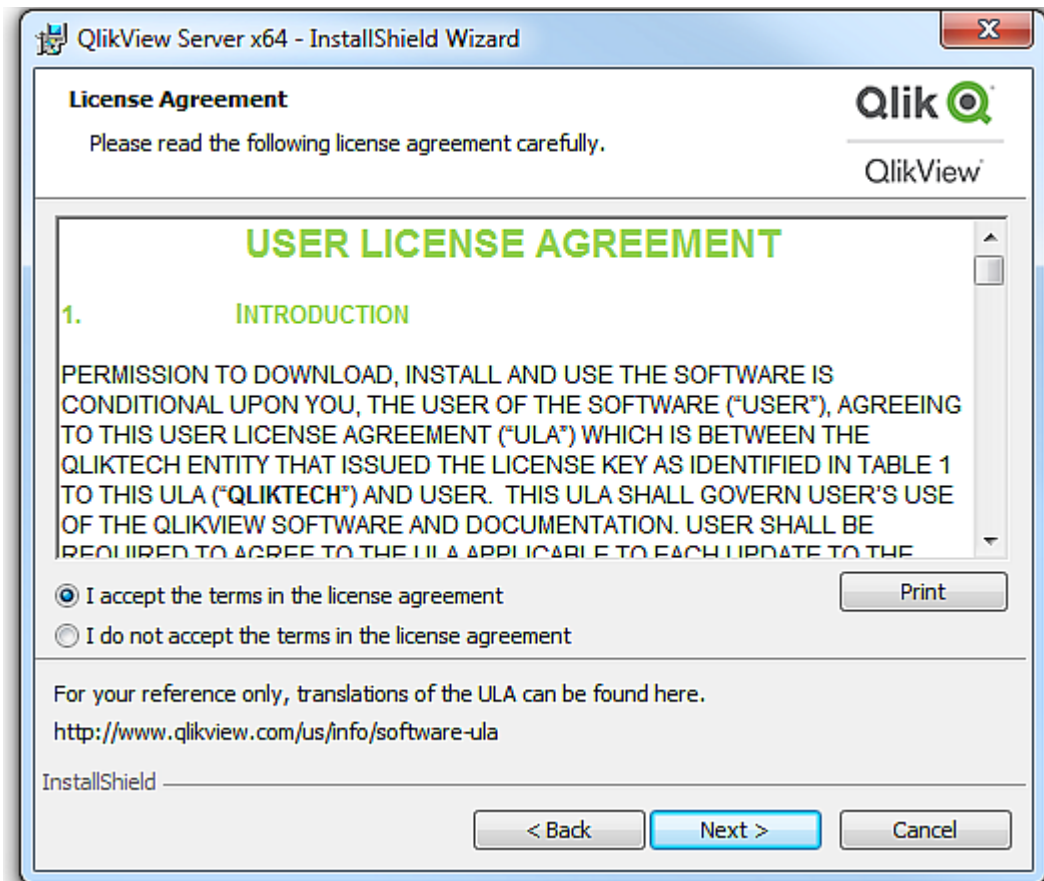


*Region dialog*

## License Agreement

This dialog displays the license agreement for the selected region.

Radio button: *AgreeToLicense* = "Yes"



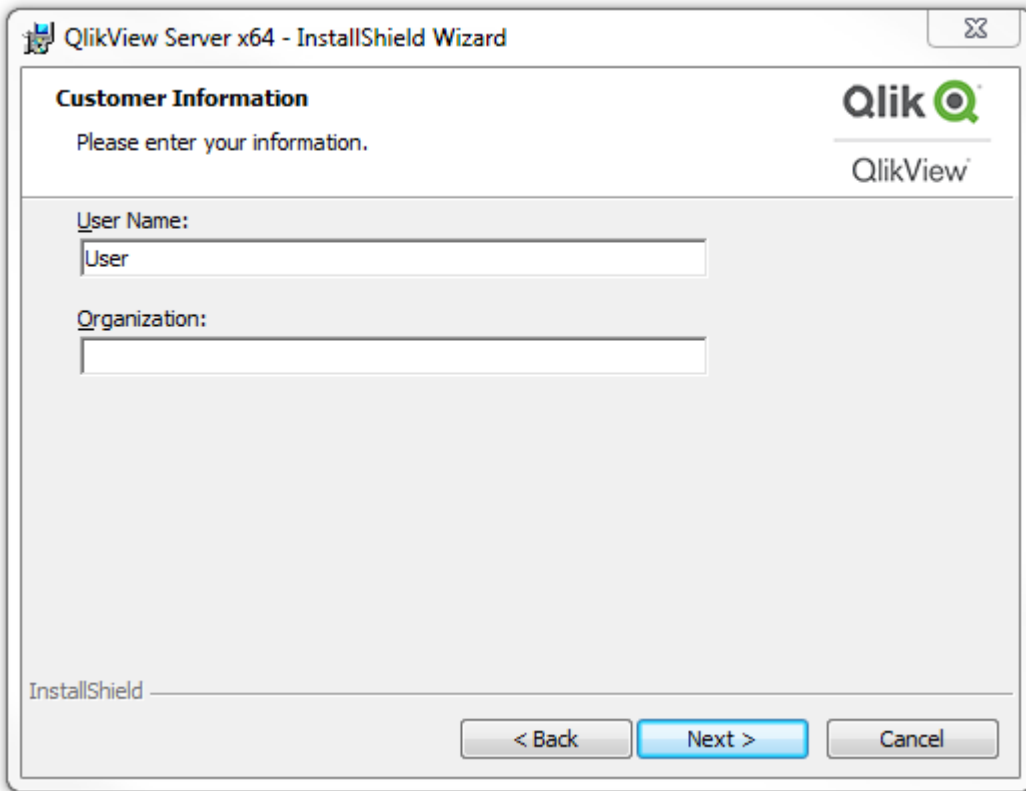
*License dialog*

## Customer Information

This dialog is used for entering the customer information.

Properties:

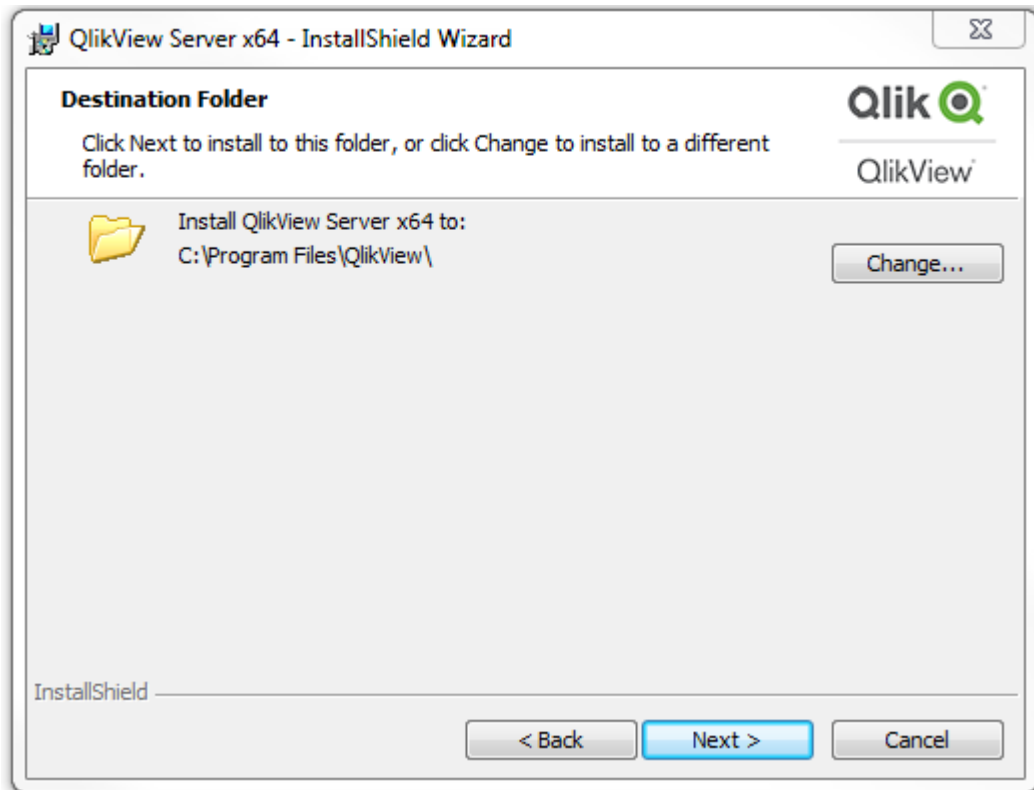
- *USERNAME*
- *COMPANYNAME*



*Customer information dialog*

#### Destination Folder

This dialog is used to set the default folder for the installation.



*Destination folder dialog*

## Profiles

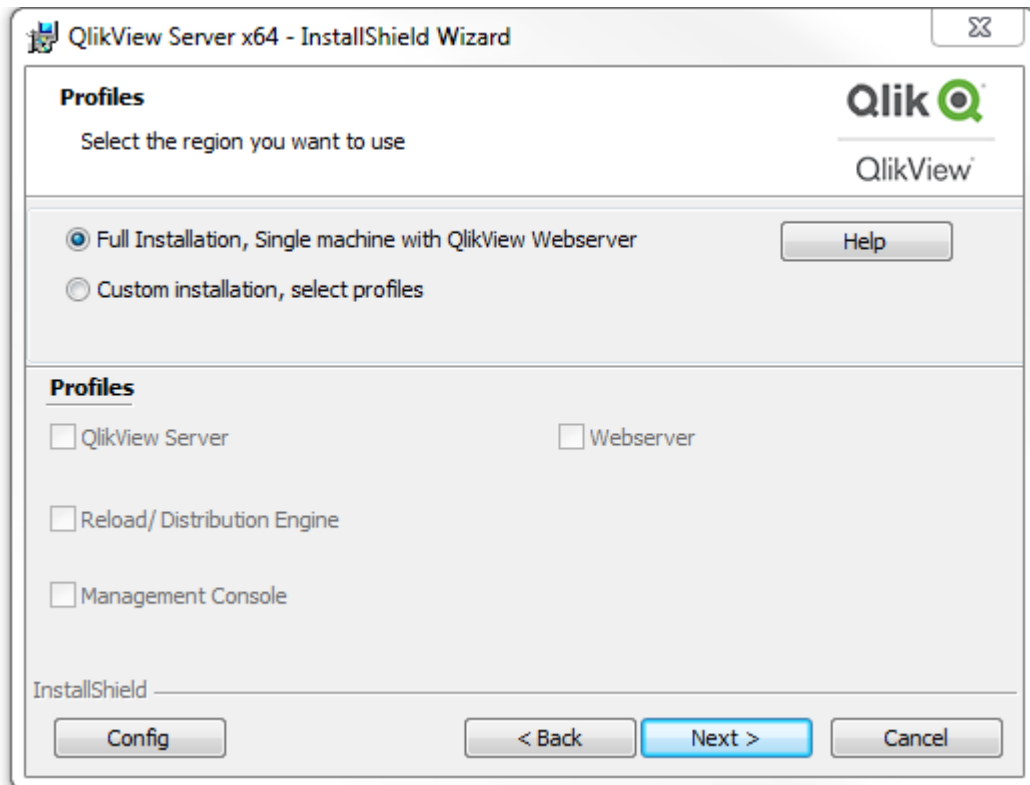
This dialog has several properties connected to it, since there are multiple profiles to choose from.

Select **Full Installation, Single machine with QlikView Webserver** to install everything, including QlikView Web Server, needed to run QlikView on a single machine. To use IIS instead, select **Full Installation, Single machine with IIS** (this option is only available if IIS is installed on the target machine).

To perform a custom installation, select **Custom installation, select profiles** and then select the profiles to install. The **Webserver** profile allows the user to choose between QlikView Web Server and IIS (if IIS is installed on the target machine).

Properties:

- *PROPQVS*: QlikView Server
- *PROPDS*: Publisher
- *PROPQMC*: Management Console
- *PROPWEB*, *PROPIIS* = 1 or 2: Webserver
- *PROPIIS* (if IIS is installed) or *PROPSTATE*: Single Machine Install



*Profiles dialog*

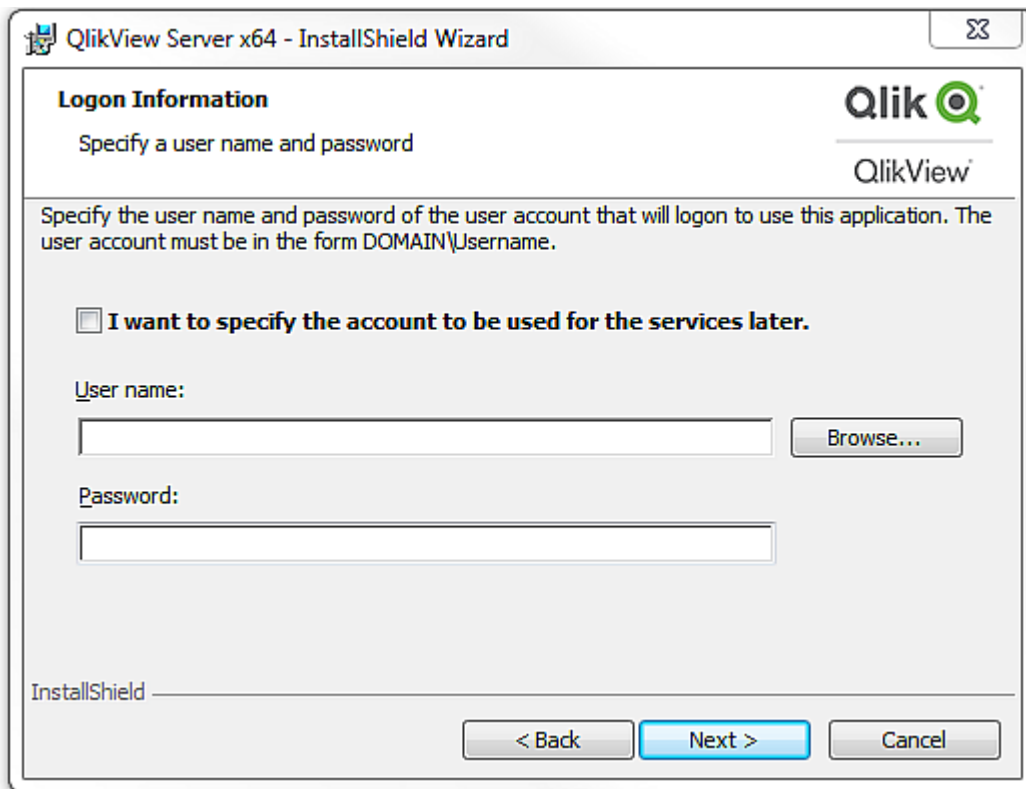
## Logon Information

This dialog, which is optional to use, is used to specify the user that is to run the services that are installed. When clicking **Next**, a Custom Action checks that the entered user is valid. The Custom Action, which is implemented by InstallShield, requires the machine to be part of a Domain to work properly.

Properties:

- `LOCALSERVICE`
- `IS_NET_API_LOGON_USERNAME`
- `IS_NET_API_LOGON_PASSWORD`



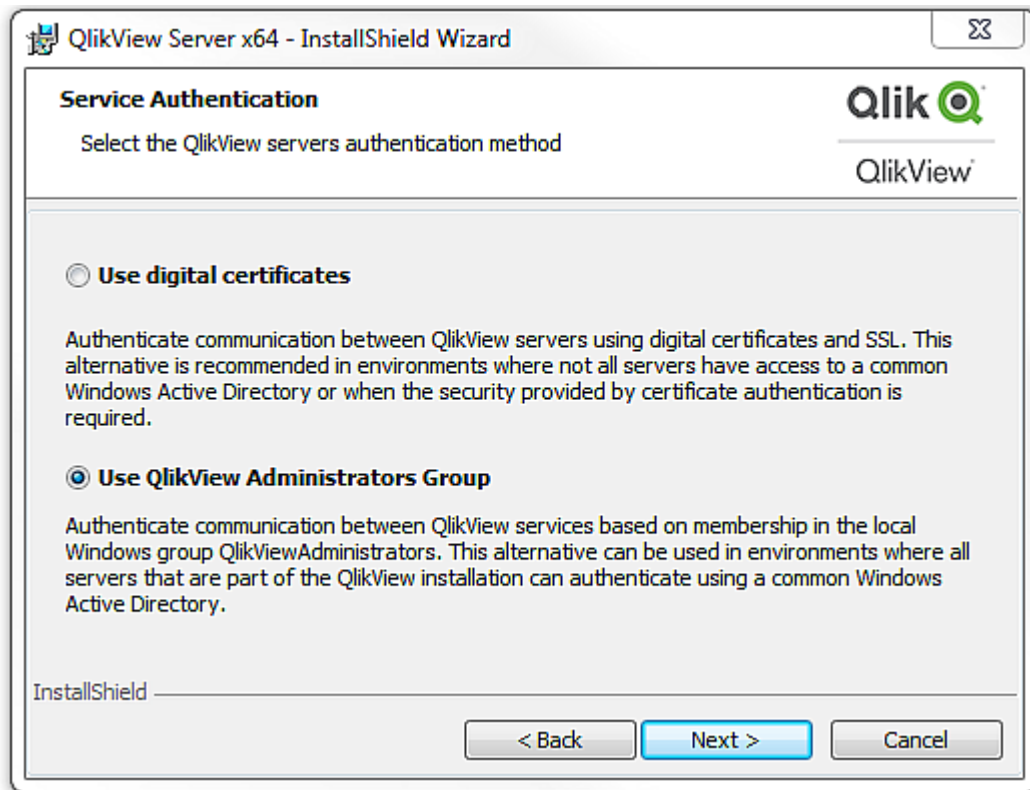


*Logon information dialog*

## Service Authentication

This dialog is used to select the type of service authentication. QlikView Administrators Group is selected by default.

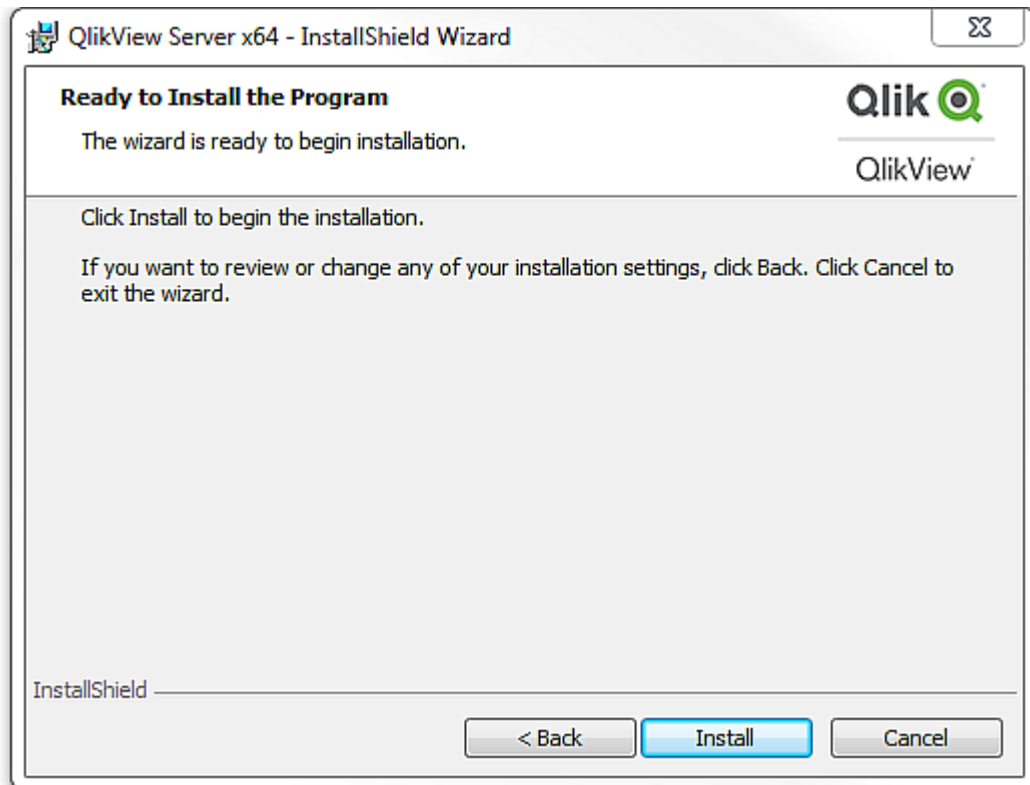
Property: *PROPCERT* (1 = Digital certificates, 2 = QlikView Administrators Group)



*Service authentication dialog*

## Ready to Install

This is the last dialog. Click **Install** to start the installation.



*Ready to install dialog*

## Additional Dialogs

### Custom Setup

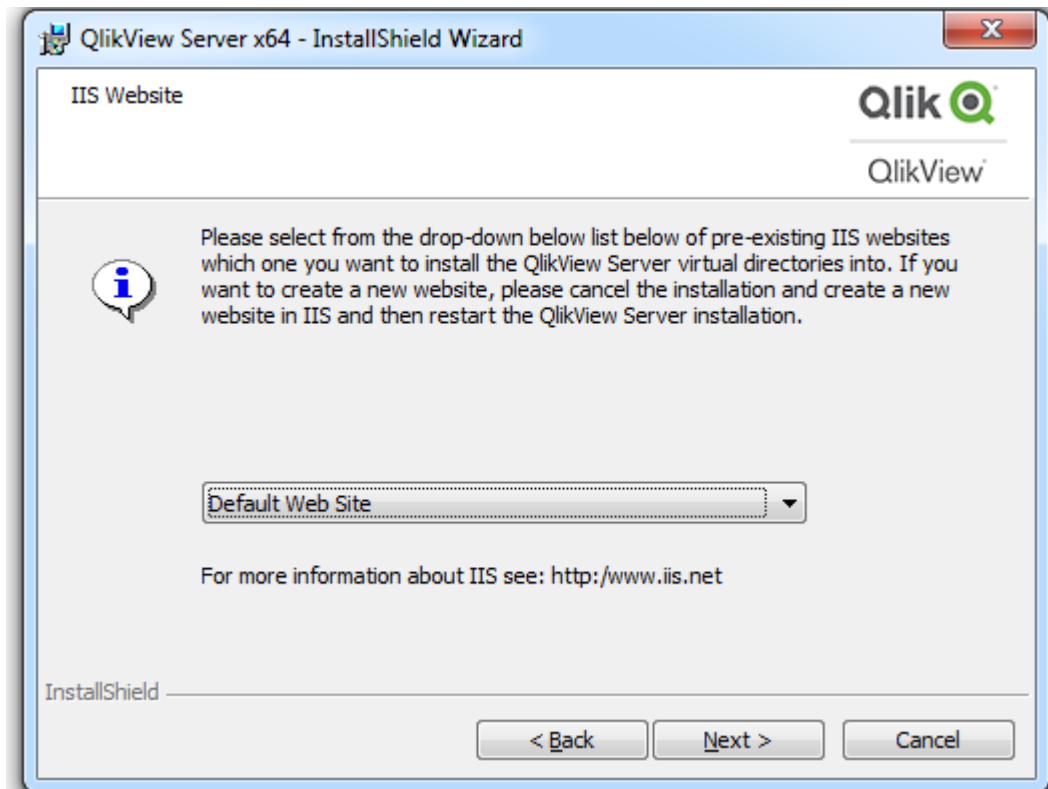
This dialog is displayed when clicking **Config** in the Profiles dialog.

*Custom setup dialog*

### Website

This dialog is displayed when selecting IIS as web server in the Profiles dialog.

Property: *DEFAULTWEBSITE*



Website dialog

### MST

When creating an MST file, the MSI file is customized without any changes being made directly in the MSI. The MST file works as a filter on top of the MSI and allows changes to be made to the installation. For example, the default installation folder for QlikView Server is `%ProgramFiles%\QlikView`, but if that is changed to `C:\QlikView` in the MST file, the default folder is changed. The same thing can be done with the dialogs, which means properties can be preset, so that the installation can be run with a limited set of dialogs.

To create an MST file, an MSI repackaging studio (for example, InstallShield AdminStudio) is needed.



*Qlik does not supply any MST files and does not take any responsibility for MST files created by customers or partners.*

### 3.3 Deploying MSI Packages with Group Policies



*This chapter is mainly intended for the Internet Explorer plugin.*

### General

A common problem today is how to deploy applications in a network environment where the users have limited rights, and how to deploy applications for a specific group of users. This section briefly describes how to deploy Microsoft Windows Installer (.msi) packages with group policies in an Active Directory environment.

The QlikView .msi packages require version 2.0 or higher of the Windows Installer service to be installed on the destination workstations.

### Deploying the MSI Package

When the .msi file has been obtained, it must be placed in a shared folder on the network. Make sure that all users and/or machines that are to install the application have read access to the folder. When the package has been made available to the users and/or machines, the Group policy object that will advertise the installation package can be created.

The package can be advertised to each user or each machine. Use the **User Configuration>Software Settings** container to advertise the package per user, and the **Computer Configuration>Software Settings** container to advertise per machine. Both containers are located in the Group Policy Object editor.

If the package is advertised per user, it can be either assigned or published. A package that is advertised per machine can only be published.

To publish a package per user means that it is listed (that is, advertised) in the Add programs from your network list in the Add/Remove programs dialog.



*Add/Remove programs dialog*

Each user must click the **Add** button to complete the installation.

To publish a package per machine means that the package is installed and accessible to all users on that machine the next time the machine is rebooted.

An advertised package that is assigned is also listed in the **Add programs from your network** list and can be added from there. This option also offers a few more ways to activate the installation package:

- Shortcuts (if the installation package adds any) on the desktop and/or Start Menu: The shortcuts are added and the installation package can be executed by clicking the appropriate shortcut.
- File association: The installation program is executed when the user tries to open a file that is associated with the advertised application.

There are a few more ways to execute the installation when it is advertised as assigned, but they are not applicable to any QlikView installations and therefore beyond the scope of this document.



*The Internet Explorer plugin installation package does not add any shortcuts or file associations. It is therefore not recommended to advertise QlikView installation packages with the assign option.*

### Advertising

To advertise means that the administrator gives the installation package permission to execute on an account with locked down permissions.

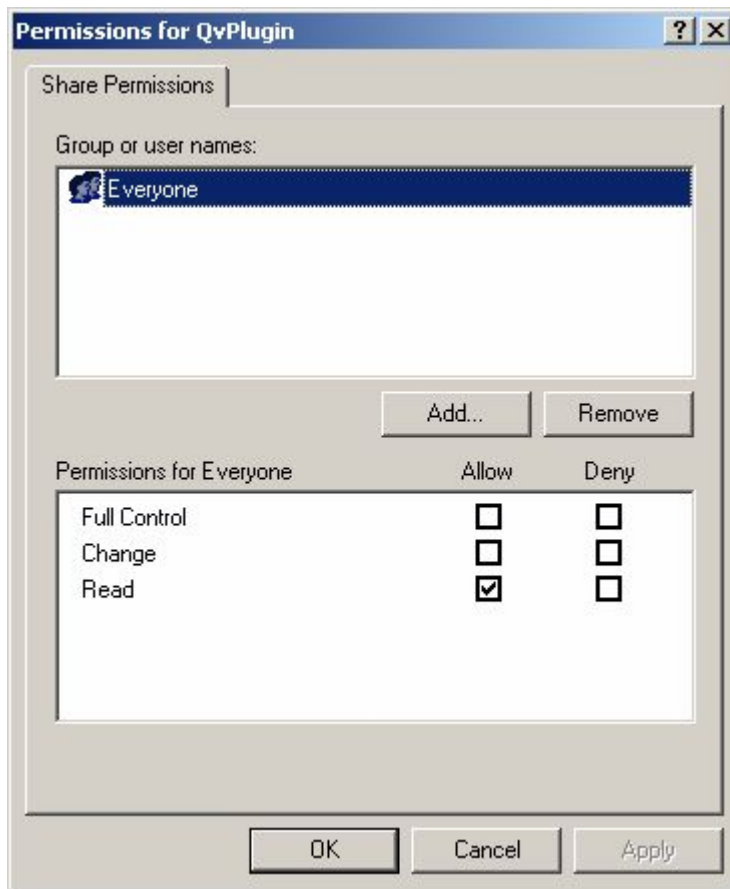
When the package is advertised, there are so called “entry points” loaded onto the destination system. Entry points are typically shortcuts, file associations, listing in the Add/Remove Programs dialog, and so on.

### Step-by-step Guide

This section provides a step-by-step guide for creating a group policy for advertising of the QlikView Internet Explorer plugin .msi package on a number of machines in the Active Directory.

Proceed as follows to create a group policy:

1. Browse to the folder containing the *.msi* package. Share the folder with the network users with permission to install the package.



*Sharing the folder*

2. Open **Active Directory Users and Computers** and highlight the **Organizational Unit (OU)** where the package is to be deployed.



*Highlighting the Organizational Unit where to deploy the package*

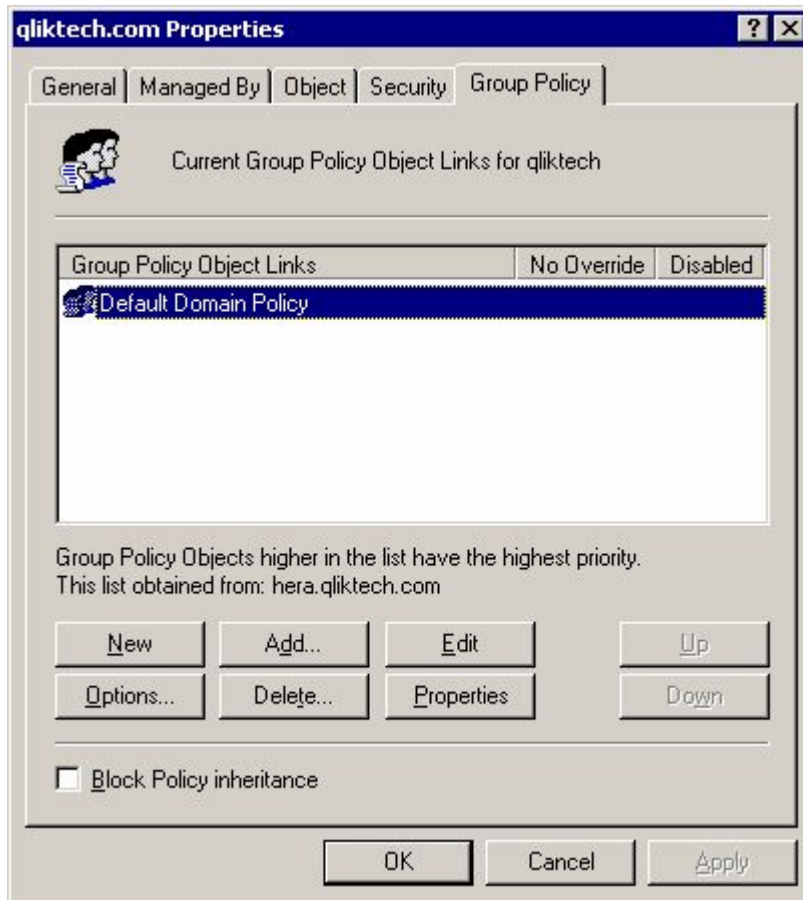
3. Right-click and select **Properties**.



*Selecting Properties*

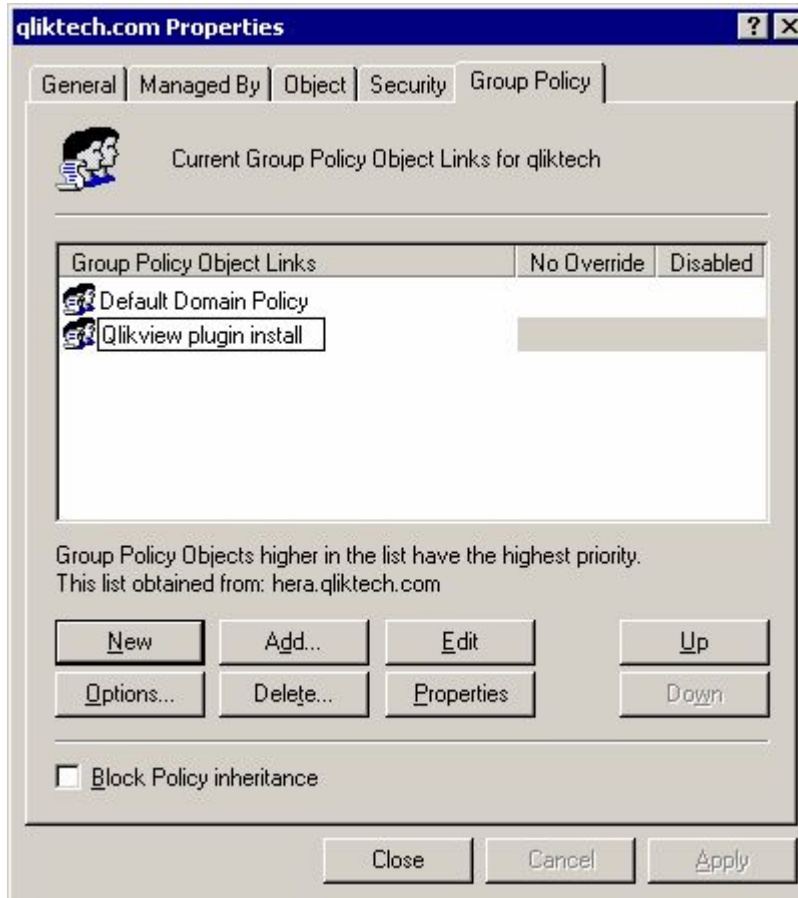


4. Go to the **Group Policy** tab, click **New**, and give the group policy object an appropriate name.



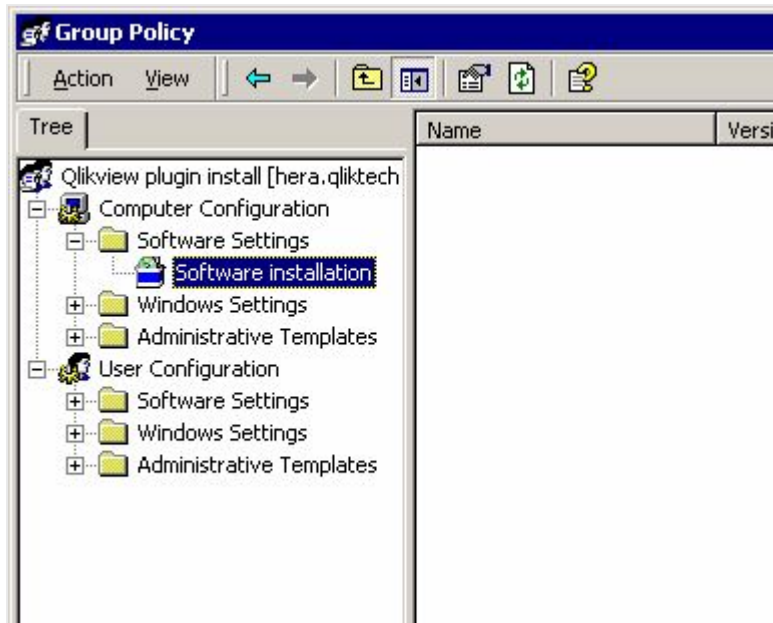
*Providing a name*

5. Highlight the new group policy object and click **Edit**.



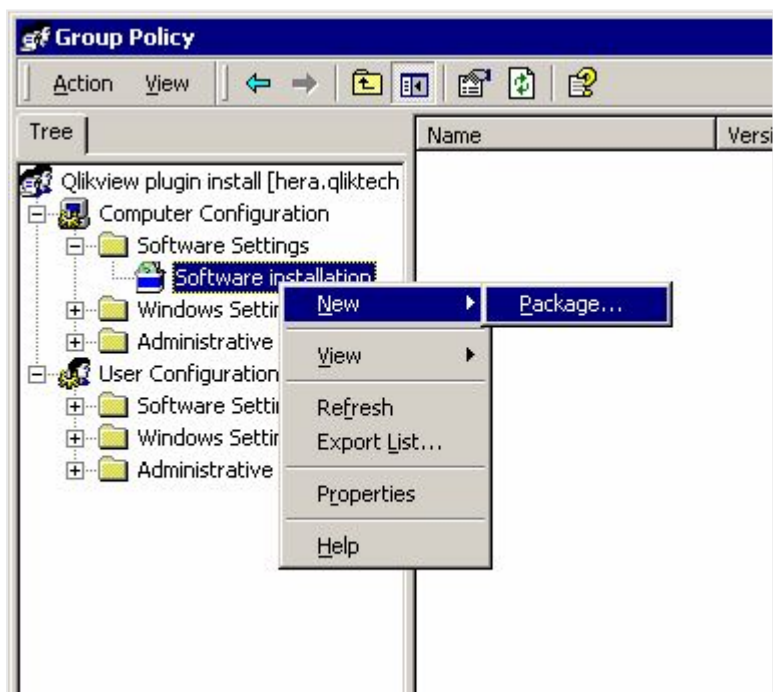
*Highlighting the new group policy object*

- Expand **Computer Configuration>Software Settings** or **User Configuration>Software Settings**, depending on how the package is to be deployed. In this case, **Computer Configuration>Software Settings** is selected.



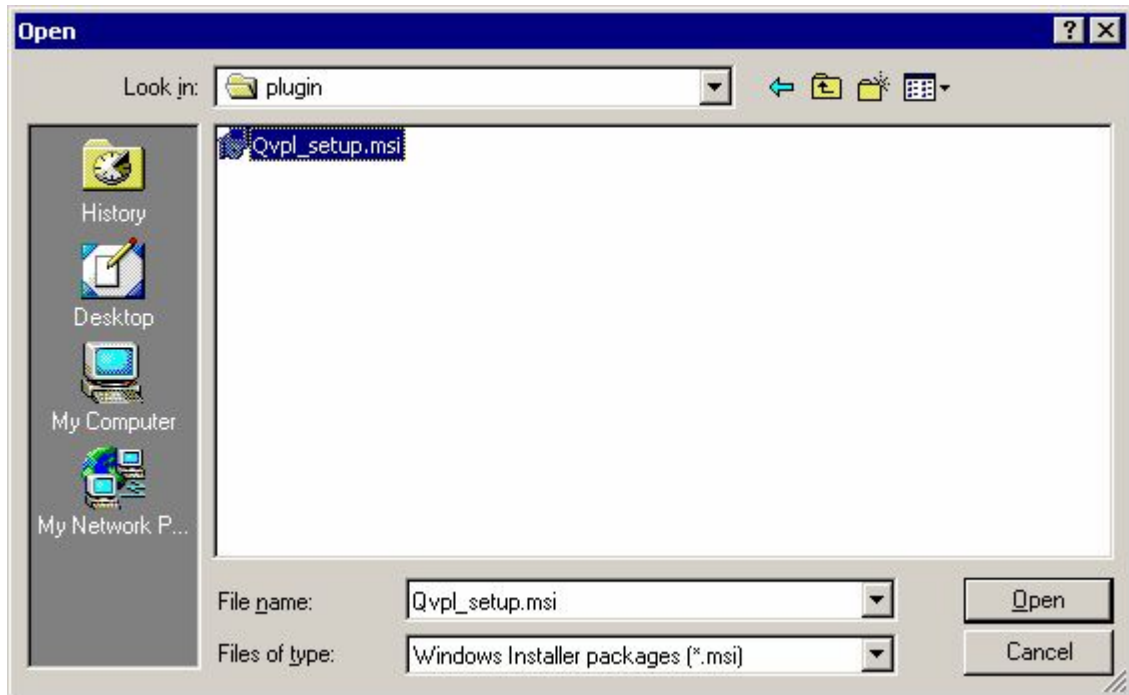
*Selecting Software Settings*

- Right-click **Software installation** and select **New>Package...** A pop-up window, asking where to locate the installation package, is displayed.



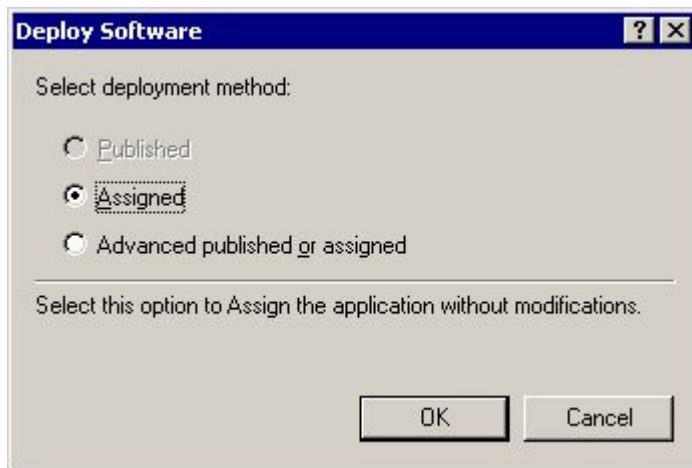
*Creating a new package*

8. Browse to the installation package (in this case, *QvPluginSetup.msi*), select it, and click **Open**.



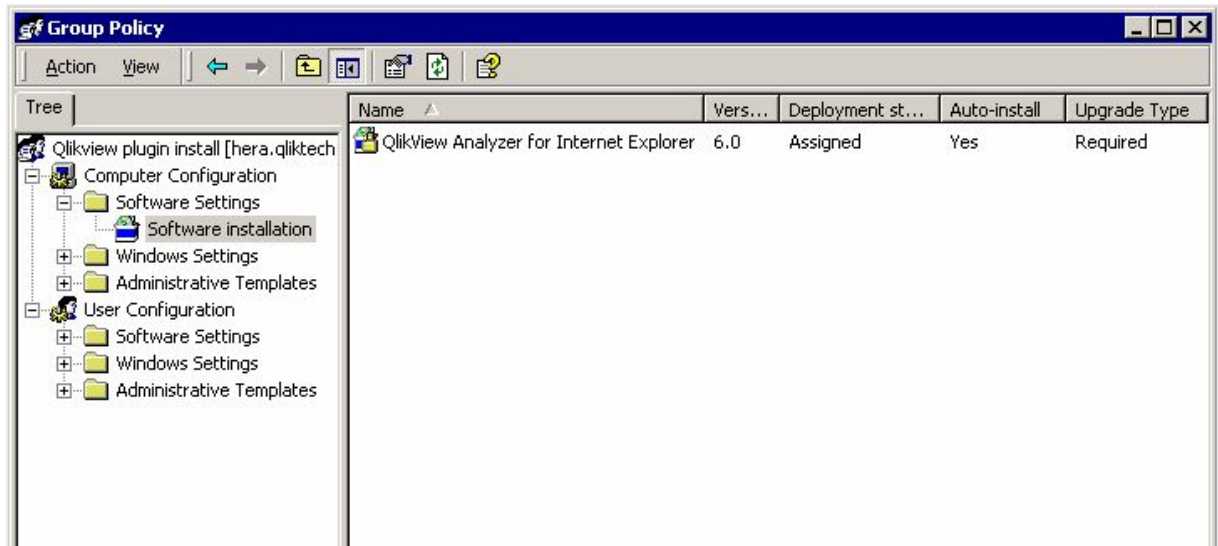
*Opening the installation package*

9. Select the deployment method **Assigned** and click **OK**. Since the installation is to be applied to the **Computer Configuration**, only the **Assigned** deployment method can be used.



*Selecting deployment method*

10. The deployment rule is now ready for use. All machines in the OU get this deployment automatically. What actually happens is that when a machine is rebooted, the installation program is executed, so that any user that logs on to a machine in that OU can run the installed program. The rule can be applied to many different OUs.



*Deployment rule is ready for use*

## 3 Upgrading QlikView

In this section, you find information on how to upgrade QlikView Desktop and QlikView Server to the latest release. Here, you can also read about the requirements necessary for the upgrade to be successful, such as creating a complete back-up of your installation, and having a valid *Maintenance contract on upgrade* (page 142). In the *Upgrade and Migration* (page 143) page you can also find information on how to migrate a QlikView Server deployment to a different machine or cluster of machines.

### 3.4 Maintenance contract on upgrade

When upgrading QlikView Server and QlikView Desktop, it is important that you have a valid maintenance contract. If you attempt to upgrade without a valid maintenance contract, the QlikView installation is restricted to an **unlicensed mode** with limited functionalities.

Every maintenance contract has a limited validity period and a specific end date. This means that your maintenance contract is not valid for a version of QlikView released after the end date indicated in your contract. However, the maintenance contract remains valid for any version of QlikView (Desktop or Server depending on what installation you have) released before the end date of the contract.

The information regarding the maintenance contract, including validity period and end date, is stored in the License Enabler File (LEF). You can verify the validity period of your maintenance contract by checking the end date stored in the LEF file. For QlikView Server, check the LEF file in the QMC. For QlikView Desktop, locate the LEF file in your local drive. Usually, the LEF is automatically transferred and stored in your computer during installation. However, there are instances when this procedure fails and the LEF file is not successfully transferred. For more information about this scenario, see the [License Enabler File Editor](#) page.

The validity of the maintenance contract is automatically checked during upgrade. If you are unsure whether your maintenance contract is valid for the QlikView version you want to upgrade to, you should refrain from upgrading. When you attempt to upgrade without a valid maintenance contract, your QlikView installation (Desktop or Server) is restricted to an **unlicensed mode** with limited functionalities. In QlikView Desktop, the unlicensed mode is called Personal Edition.



*If you launch the upgrade but your maintenance contract is not valid for that specific QlikView version, a warning message is displayed. This warning message is displayed **only** when the maintenance contract is not valid for the requested QlikView version.*

In case your QlikView Desktop or Server installation is restricted to the unlicensed mode, you can revert it to the previous version. Follow this downgrade procedure.

### Restoring an older QlikView Desktop installation

To restore a previous installation of QlikView Desktop, you must uninstall the current instance and install the older version for which you own a valid maintenance contract.

### Restoring an older QlikView Server installation

Before installing a newer version of QlikView Server, read carefully the "Upgrade QlikView Server" procedure in the *Upgrade and Migration (page 143)* page. This procedure helps you create a backup of your QlikView Server installation, including the QlikView Publisher Repository (QVPR) database. Creating this backup before upgrading allows you to successfully restore a previous version of QlikView Server in case the newer installation is restricted to the unlicensed mode.

To restore a previous installation of QlikView Server, follow these steps:

- Uninstall the unlicensed instance of QlikView Server
- Install an older version of QlikView Server for which you have a valid maintenance contract.
- Restore the QVPR and data directory backups.

## 3.5 Upgrade and Migration

### Best practices

For a successful upgrade of QlikView, take the following basic practices into account:

#### QlikView Desktop

- Ensure that you have a valid maintenance contract before upgrading QlikView Desktop. Attempting to upgrade without a valid maintenance contract will result in limited functionality of QlikView Desktop. See: *Maintenance contract on upgrade (page 142)*.
- If you are using a custom connector or an extension in your QlikView Desktop installation(s), verify that such feature is supported in the newer version before upgrading. You can check the supported features in the Download section at [qlik.com](http://qlik.com).

#### QlikView Server

- Perform the upgrade during a scheduled downtime. QlikView Server must be stopped for the upgrade to be successful.
- Licensing information and settings are saved by default when QlikView Server is removed. They are applied to any subsequent installation of QlikView Server on the system.
- Ensure that you have a valid maintenance contract before upgrading QlikView Server. Attempting to upgrade without a valid maintenance contract will result in limited functionality of QlikView Server. See: *Maintenance contract on upgrade (page 142)*.
- If Digital Certificate Authentication is used for QlikView Service communication, the new certificates that are created during the upgrade must be installed. See *Update certificates (page 144)*.



*The installation does not support upgrade from beta or release candidate versions of QlikView Server.*

### Upgrade QlikView Desktop

To upgrade QlikView Desktop, download the newer version you want to install from [qlik.com](http://qlik.com), and follow the installation wizard.

### Upgrade QlikView Server

To upgrade QlikView Server, proceed as follows:

1. Verify that backup media exists for the current release of QlikView Server and back up the current QlikView data directory (which includes most of the log and some of the configuration files, the document folders, HTML pages, licensing file, QlikView Server shared files, and so on). The files are typically located in %ProgramData%\QlikTech.
2. Back up the QlikView Publisher Repository (QVPR) database.



*The backup must be created before you attempt to install a newer version of QlikView Server.*

3. Install QlikView Server.
4. The installation of QlikView Server requires a reboot of the machine for proper operation.

### Multi-machine Preparation

When upgrading a QlikView Server installation that is spread over multiple machines, extra planning is required, since versions cannot be mixed arbitrarily.

### Simple Upgrade

This procedure requires no special planning and involves the smallest risk, but causes the system to be down for some time.

Proceed as follows to perform a straight-forward upgrade:

1. Perform a backup.
2. Stop all services running on all machines.
3. Upgrade the services on each machine (in any order).
4. Start all services on all machines.

### Update certificates

The new certificates that are created during an upgrade must be installed on machines running QlikView services. Certificates do not need to be installed on the machine running the QlikView Management Service.

Proceed as follows:

1. Open the QMC.
2. Click the **Status** tab, and then click **Services**.
3. Note any services that are disconnected.



4. Click the **System** tab, and then click **Setup**.
5. Select a service, and then click the **General** tab for the service.
6. Click the **Apply** button in the bottom right of the window, and then follow the instructions to install the certificate.
7. When you are done, navigate back to **Services** on the **Status** tab to verify that the service is now running.

### Maximize Uptime

This procedure requires more planning, but the system uptime (from an end user point of view) is maximized.

Proceed as follows to perform the upgrade:

1. Perform a backup.
2. Stop QMS (which means QMC becomes unavailable).
3. Upgrade in the following order (let the installer restart the services):
  - a. Web servers
  - b. Directory Service Connector (DSC)
  - c. QlikView Server (QVS)
  - d. QlikView Distribution Service (QDS)
  - e. QMS
4. Start QMS (which means QMC becomes available again).

### Migration to a New Machine



*This procedure is specific for migrating QlikView Publisher Repository (QVPR) only.*

An alternative way is to build the new environment on new servers.

Proceed as follows to perform a migration to a new machine:

1. On the new machine, install a running, licensed version of QlikView Server.
2. On both machines, stop all QlikView services.
3. On the new machine, remove or rename the `%ProgramData%\QlikTech\ManagementService\QVPR` folder.
4. On the new machine, remove or rename the `%ProgramData%\QlikTech\ManagementService\qvpr_<NewMachineName>.ini` file.
5. Copy the `QVPR` folder and the `.ini` file “as is” from the old machine to the new one (that is, keep the folder name).



*All QlikView servers must have the same regional settings. Different regional setting may cause errors when loading QlikView XML reference files.*

6. Rename the *.ini* file (that is, change *qvpr\_<OldMachineName>.ini* to *qvpr\_<NewMachineName>.ini*).
7. Change all references from *<OldMachineName>* to *<NewMachineName>* in the QVPR *.xml* files.
8. Start the QlikView services on the new machine.
9. In QMC, change the source folder path to the correct folder (or the tasks cannot be edited).
10. Shut down the old machine.