Communication Ports

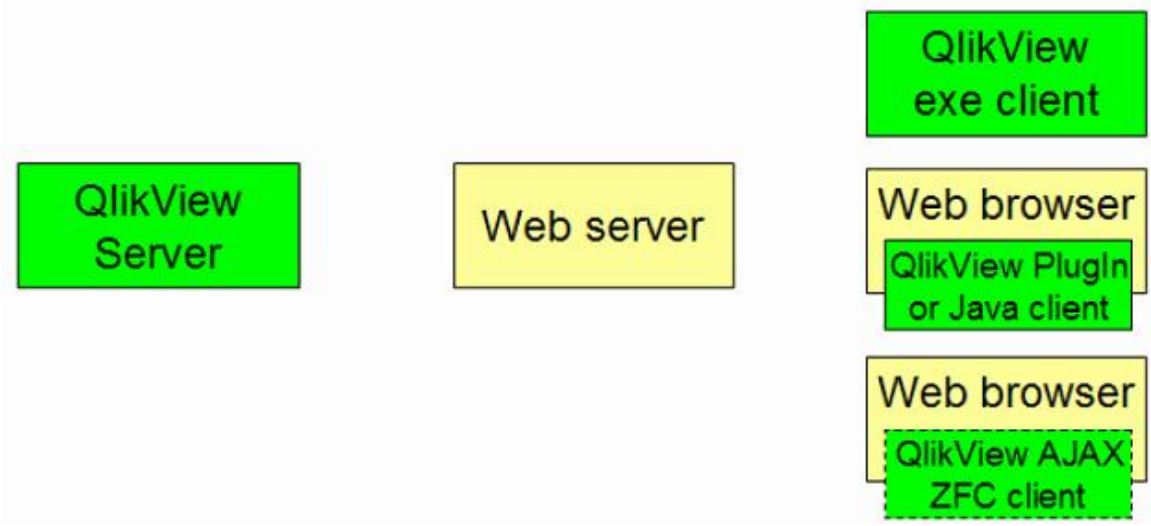
The QlikView client communicates with the QlikView Server over one or more TCP ports. The default port is 4747, and this port should work for most environments. If you have a firewall, you can either open Port 4747 for QlikView sessions (e.g. for public Internet 60 use) or tunnel QlikView via http (typically Port 80) by using the QVSTunnel.dll that is installed in the Scripts virtual directory during installation.

Multiple ports can be specified. QlikView Server will try to initiate communication with the client over each port in the order they are listed until a working port is found. This is usually only required for Internet sessions where your users may be behind a firewall outside of your control. If the user's firewall has blocked communication over your default port, QlikView Server will continue with the other listed ports until an unblocked port is found.

QLIKVIEW SERVER FUNCTIONAL ARCHITECTURE

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 computers and multiple network connections, as well as other subordinate processes.



The three primary processes are:

1 The **QlikView Server (QVS)**, which provides QlikView functionality to the client. The machine that is hosting this service must be running in a Microsoft Windows Operating System.

2 The **Client**, running in a web browser or a QlikView shell that provides a container for the client code. The client communicates with QlikView Server either directly or through the Web Server to provide the QlikView interface and functionality to the end user.

3 The **Web Server (or HTTP)**, running an HTTP server, which can be used to serve up the HTML web page to the client, assist with authentication of the user, and enable communication between the client and QlikView Server.

In the simplest scenario, all three processes can be running on a single machine, with a single user. The complexity of this relationship can increase quickly, however, as separate machines, Internet connections, multiple firewalls, and multiple Web Servers are introduced. Finally, multiple users who require security authentication and authorization from a myriad of Directory Services are added, and a QlikView Server – Client communication architecture can become quite involved.
There are, of course, a large number of possible network configurations that QlikView Server can participate in, but there are a few considerations to keep in mind regardless of the final configuration:

- QlikView Server runs as a Windows OS Service only
- At least one network communication path must exist between the QlikView Server and the Client
- The authentication of the Client user must be performed either through Windows Authentication, QlikView Authentication (section access), or any third party system that can authenticate the user

QlikView Server Functional Description

There will be one QlikView Server process per logical computer, which must be running a Windows Operating System. QlikView Server can run as a 32- bit or 64-bit process (OS and hardware dependent). The QlikView Server process can be identified as qvs.exe.

Client Access License (CAL)

All client access to QlikView Server must be licensed. This is accomplished through the use of Client Access Licenses (CALs) linked to the specific instance of the QlikView Server through the LEF file. In this context, it is important to understand the definitions of anonymous user and authenticated user.

Anonymous user – an unidentified or unknown user (any user)
There is no authentication for anonymous users, they can be anyone.

Authenticated user – an identified user whose identity can be verified

Authenticated Windows OS user (e.g. NTNAME, NT User, NTDOMAINSID)

* Authenticated non-Windows user

Authenticated QlikView user (e.g. section access: USERID, PASSWORD)
QLIKVIEW SERVER FUNCTIONAL ARCHITECTURE

* Authenticated third party (build partner) user
* As of release 8.2+

The type of CAL will affect how users are allowed to connect to QlikView Server, based on the Client type and Authentication settings in the Web Server and/or QlikView Server.

A SESSION CAL allows any user – authenticated or anonymous – to connect to QlikView Server. There is no limit on the number of sessions over time, but only one session at any time is allowed per session CAL.

A USAGE CAL allows any user – authenticated or anonymous – to connect to QlikView Server. Each usage CAL is linked to a specific user – document session, and can only be initiated once per 28 day cycle.

A USER CAL allows only specific users (or machines) to connect to QlikView Server. The Java and AJAX Clients will not allow anonymous OS users or machine names as user CALs. Therefore, anonymous access should be disabled in the Web Server (or in QlikView Server if using the QVS http web server) when using these clients.

Client Functional Description
QlikView Server can support three categories of Clients:

1 Windows Clients – this includes the QlikView program variants of QlikView Analyzer, Analyzer+, Professional, and Developer. This category also includes the Internet Explorer plug-in ActiveX client running as a full window or object only (QlikX). All Windows Clients require installation with Administrator level rights. QlikView Analyzer+, Professional, and Developer require licensing on the Client machine in addition to the QlikView Server CAL.

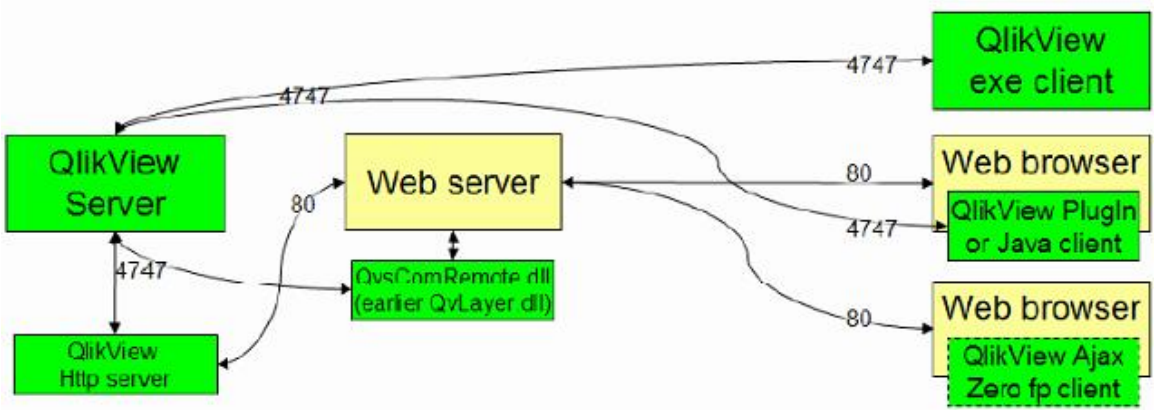
2 Java Clients – this includes Sun/Apple Java running in a browser window, with either a full window or objects support. Java Clients require installation of the Java applet, although this installation is handled automatically through the Web Server, and does not require user intervention. No Client side licensing is required.

3 AJAX (ZFC) Clients – this includes the AJAX Client, which supports HTML objects only. No Client side installation or licensing is required.

Client Communication to QlikView Server
Windows Clients can communicate directly with QlikView Server, using QVP (QlikView Protocol) and, typically, port 4747. These clients do not require a Web Server to establish and maintain a connection with QlikView Server.

Java Clients can communicate directly with QlikView Server, typically, over port 4747, but they must first establish a connection with the Web Server (e.g. IIS or the QVS http web server) serving the page containing the Java applet, typically, over port 80 (http).

AJAX (ZFC) Clients can never communicate directly with QlikView Server. They must establish and maintain a connection through a Web Server (e.g. IIS or the QVS http web server). This is typically accomplished through port 80 (http).



Web Server Functional Description

Traditionally, the standard web server in a QlikView Server configuration has been Microsoft Internet Information Services (IIS). As of QlikView version 8, however, an alternative solution is offered, and is included with the QlikView Server installation. This is the QVS http web server. This web server can act as a stand alone service, but is not configured to handle asp pages.

Other web servers can be utilized in a QlikView Server environment, but there are some restrictions. If the other web server is able to direct traffic to the QVS http web server (running on the same machine as QlikView Server), the possibilities are many, including the configuration with the other web server running under a non-Windows OS. If the other web server must utilize a local QlikView Server dll (QvComRemote.dll) to communicate with QlikView Server (e.g. for tunneling), then the other web server must be running under a Windows OS.

Web Server on Separate Machine from QlikView Server

If the IIS or QVS http web server is running on a separate machine from the QlikView Server, you will need to configure the location of the QlikView Server, and optionally, the port, to allow the web server to locate the QlikView Server. The configuration requirement will vary, based on which web server you are using.

IIS web server

Edit the file QvClients\settings.js to point to the QlikView Server, and optionally, the port. Change the vars QvsHost and Qvs- Port to match your environment, and remove the comments. The QvsViewClient.asp is configured to include the settings.js code, but you will need to remove the comment tags.

QVS http web server

Edit the file HttpServer\config.xml to point to the QlikView Server. Change the tags QvsHost and QvsTunnel to match your environment.

QlikView Server Tunnel

If the standard communication port to QlikView Server (4747) is blocked in any way (typically by a firewall limitation), the Windows and Java Clients will attempt to reroute their connection through port 80 (http) This connection path must then include the QvsComRemote.dll or the QVS http server so that the QlikView Tunnel communication can be established. All communication through the QVS Tunnel must include the secure communication packet, so this will significantly increase the network traffic (along with response times) required between the QlikView Server and the client. The infrastructure might also interfere, for example, if the traffic is

routed through proxy servers. This is especially true if tunneling using HTTPS. It is recommended to set up rules to bypass proxy servers when tunneling using HTTPS.

The QlikView Tunnel is installed into the Web Server process and allows the QlikView Client to be tunneled over the HTTP protocol to the HTTP process and then forwarded onwards to the QVS process.
When there is a requirement for the HTTP process to run on a third machine (perhaps since it is not a Microsoft Windows server) but communication between the Client and the HTTP machine is restricted, then the setup is similar. The HTTP machine having a Tunnel installed to redirect the QlikView Client protocol on the QVS machine Communications between the QVS and HTTP cannot be restricted in any way.
Finally, if the HTTP process must run on a third machine and communication between the Client and HTTP machine is not restricted in any way, then another process can come into play. This is a TCP/IP Redirector (or Redirect) that runs on the HTTP machine. It is required because (in the case of Java) the Client applet can only connect to the machine that served the web page containing the applet. The redirect process accepts the connection from the applet for the QlikView Client protocol and forwards it onto the actual QVS machine. The Redirect process may be a separate program, part of the Operating System of the HTTP machine or even a function of the firewall/proxy system in use between the HTTP machine and the Client machine. All that matters is that both the machine name and the IP address of the Redirect is the same as the HTTP machine.

For Tunneling on a Windows Server using IIS or the QVS built-in http server
If you have defined a path to the QVSTunnel.dll in the Control Panel, you have the possibility to tunnel the communication between the server and the client. The dll-file is by default copied to the following directory during installation:
C:\Program Files\QlikView\Server\QvTunnel

A virtual directory is set in IIS and the QVS built-in http server configuration as:

Scripts
If the client cannot connect via the default TCP connection, the client will by default try to connect via http (Port 80).

QVS HTTP:
Edit the Config.xml file to specify the location of the <QvsHost> and <Qvs- Tunnel>. <QvsHost> is used in all non-tunnel-cases and <QvsTunnel> when tunnelling is requested. It is thus possible to have one Qvs handling all non-tunneling and another handling tunnelling. Note that if you omit <Qvs- Tunnel> the HttpServer will NOT support QVS tunnel.

```
<?xml version="1.0" encoding="utf-8" ?>
<Config>
<QvsHost>HIC-HP</QvsHost>
<QvsTunnel>HIC-HP</QvsTunnel>
<Url>http://HIC-HP/</Url>
```

Microsoft IIS:
Two entries are required in the registry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\QlikTech\QlikTunnel]
"QVSPort"=dword:000012a6
"QVSServer"="QvsHost"
```

The QVSPort entry should already exist, but the QVSServer must be added manually.

Tunneling from Windows clients
Tunneling from Windows clients is achieved by adding:http as protocol in the pseudo-URL describing the server or document address.

QlikView Tunnel Test Procedure
You can test the QlikView Tunnel by entering the following URL from a Client browser window:
http://Server/scripts/qvstunnel.dll?testtunnel
Where
Server is the Web Server name or address
If the QlikView Tunnel is set up correctly, the webpage should return with: