

Qizx Server control - Online help

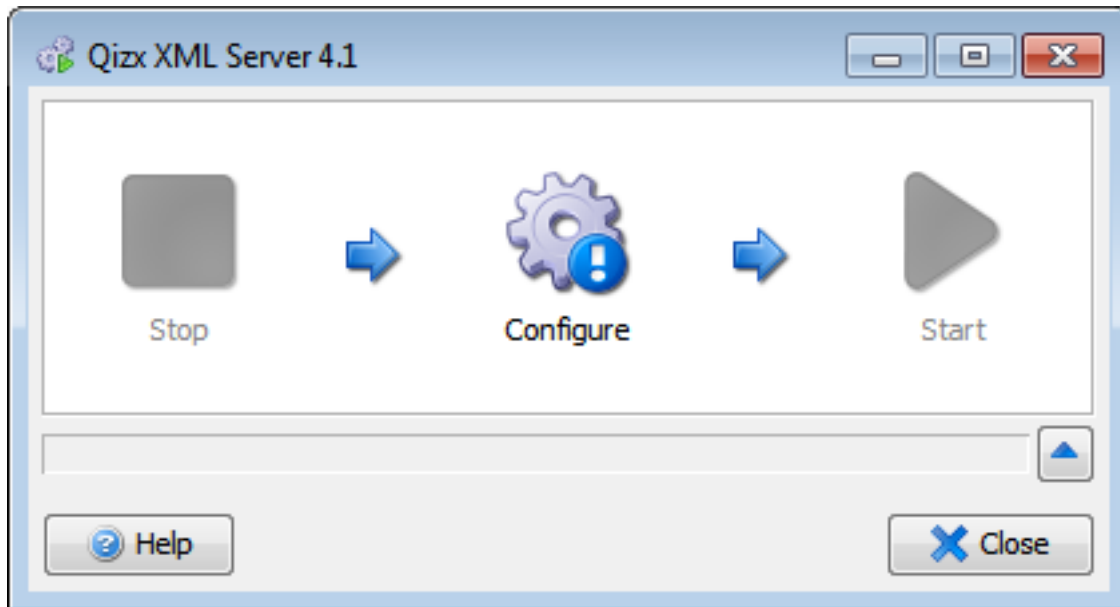
Table of Contents

Main window	1
Console mode	3
Choose between a normal Web server (HTTP) and a secure Web server (HTTPS)	4
Choose between a self-signed server certificate and an actual server certificate	4
Specify an existing server certificate	5
Generate a self-signed server certificate	6
Specify a TCP port for the Web server	7
Open the server port in Windows® firewall	7
Choose an authentication scheme	8
Declare users	9
Run Qizx XML Server as a Windows® native service	9
Running Qizx XML Server as a Unix daemon	10
Review what you have specified	10
Troubleshooting	11

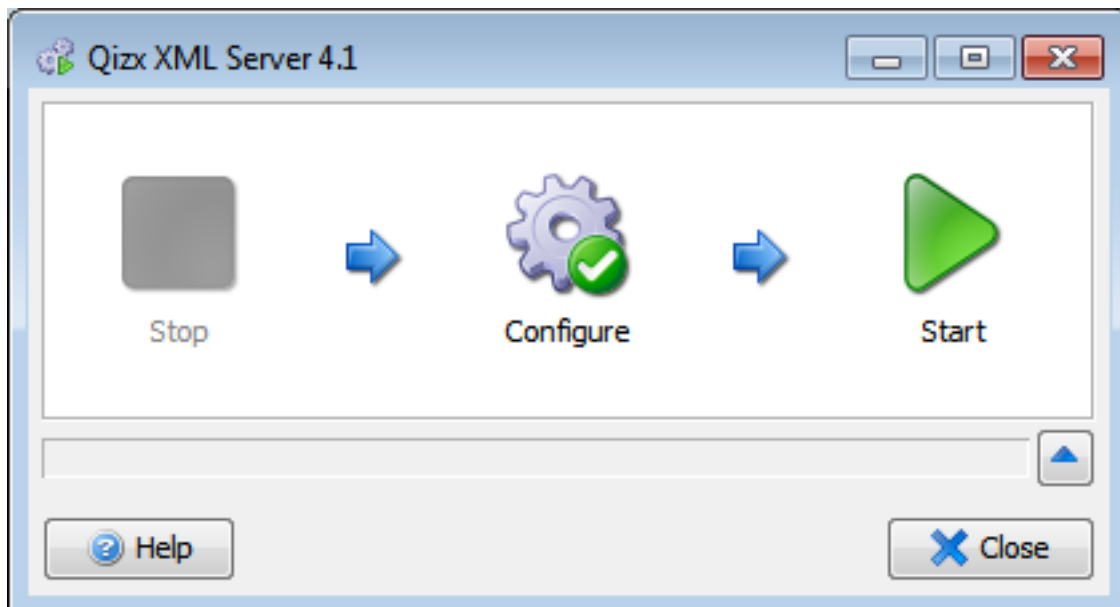
Main window

Application `qizxserver` allows to stop, configure or reconfigure, start or restart Qizx XML Server. It does this by configuring and controlling the bundled [Jetty 7 Servlet Container](#).

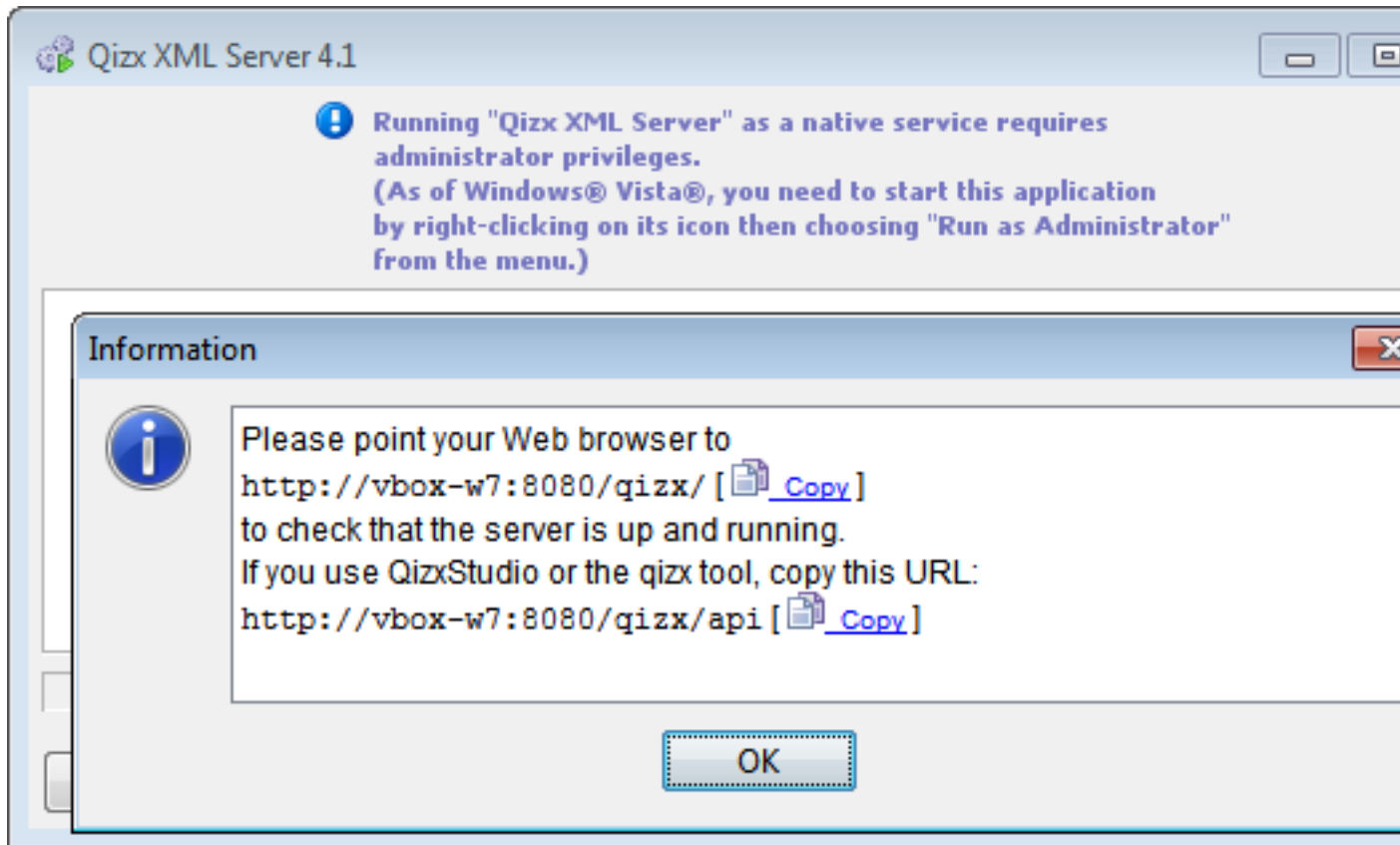
You must first configure Qizx XML Server before being able to start it. This is done by clicking the **Configure** button.



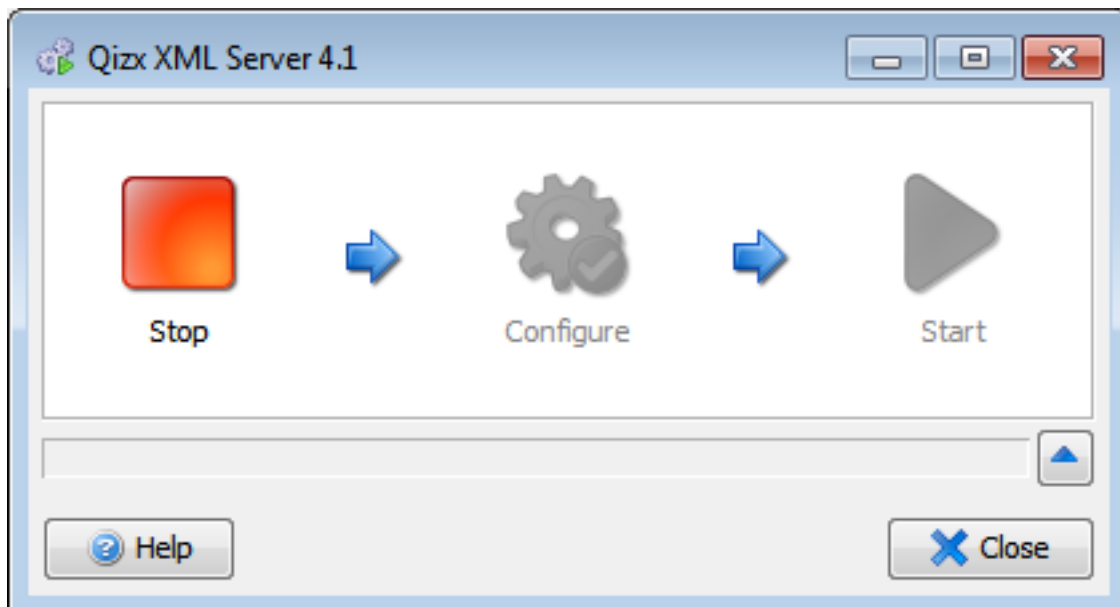
Once Qizx XML Server is configured, you can start it. This is done by clicking the **Start** button.



After doing that, a dialog box is displayed to show you how to connect to Qizx XML Server.

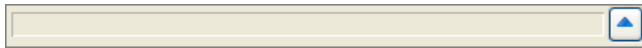


Once Qizx XML Server is started, you can close *qizxserver*. After that, you can re-run *qizxserver* at any time to check whether Qizx XML Server is still running, to stop it, to reconfigure it, etc. Stopping Qizx XML Server is done by clicking the **Stop** button.

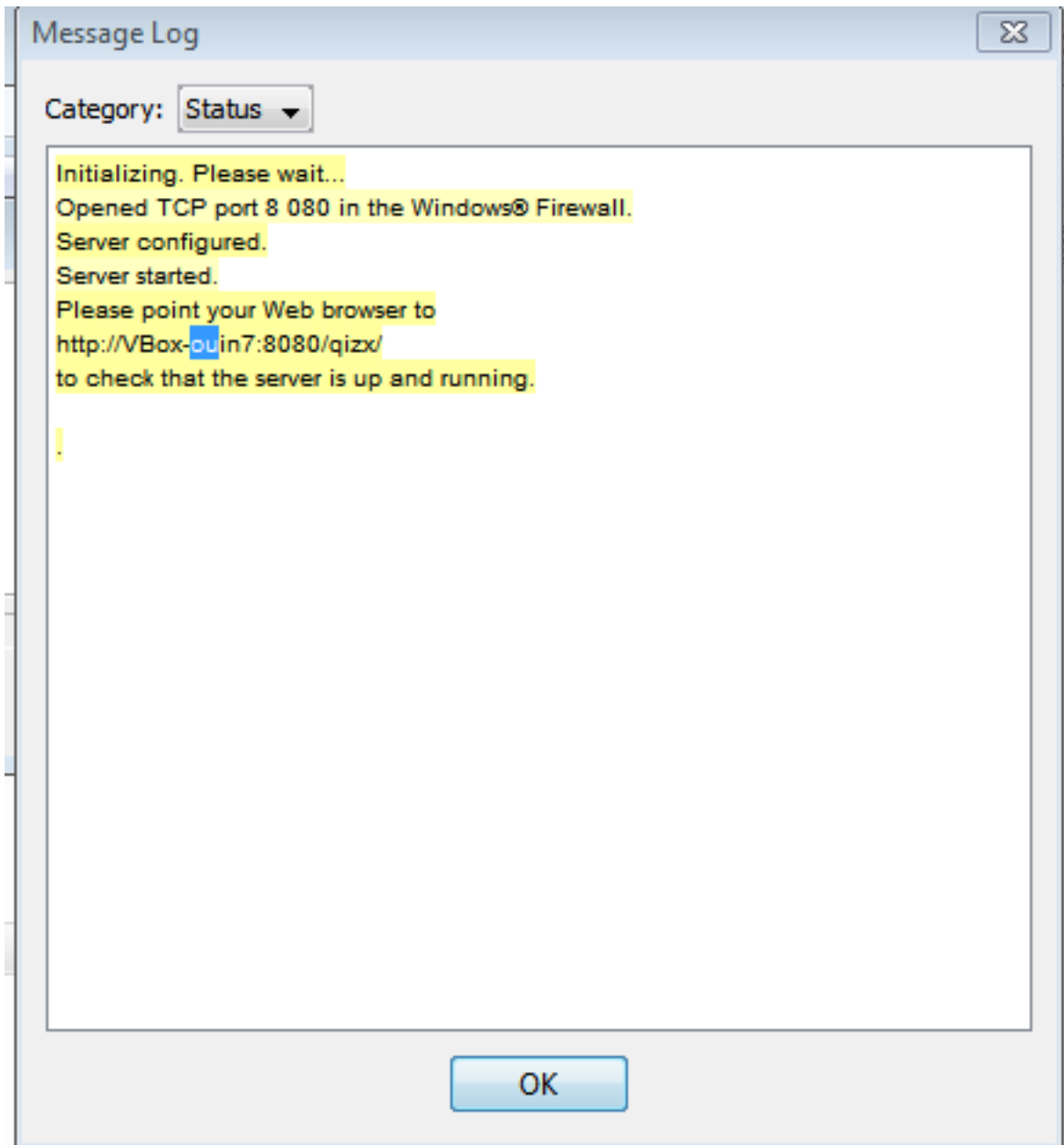


Message log

If you cannot remember what has been displayed by the above **Information** dialog box or if you want to copy the URL of Qizx XML Server to the clipboard, right-click on the status line or click the **"Show message log"** button found at the right of the status line.

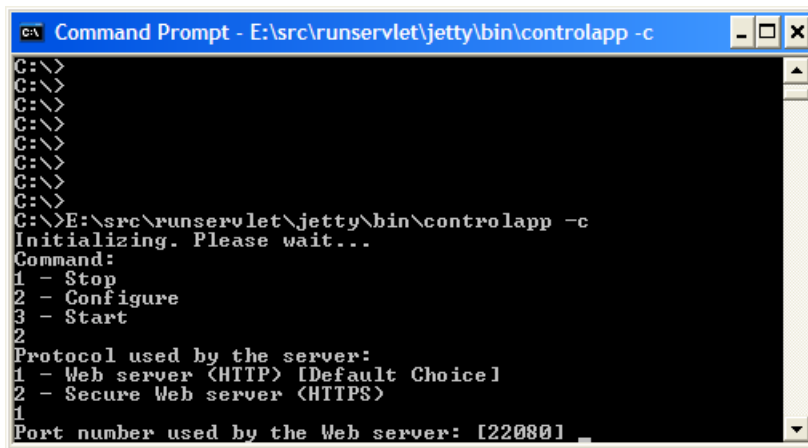


Doing this displays the following message log dialog box:



Console mode

Application `qizxserver` works in both graphical and *console mode*. In order to start it in console mode, simply add the `-c` option to the command-line. Example:

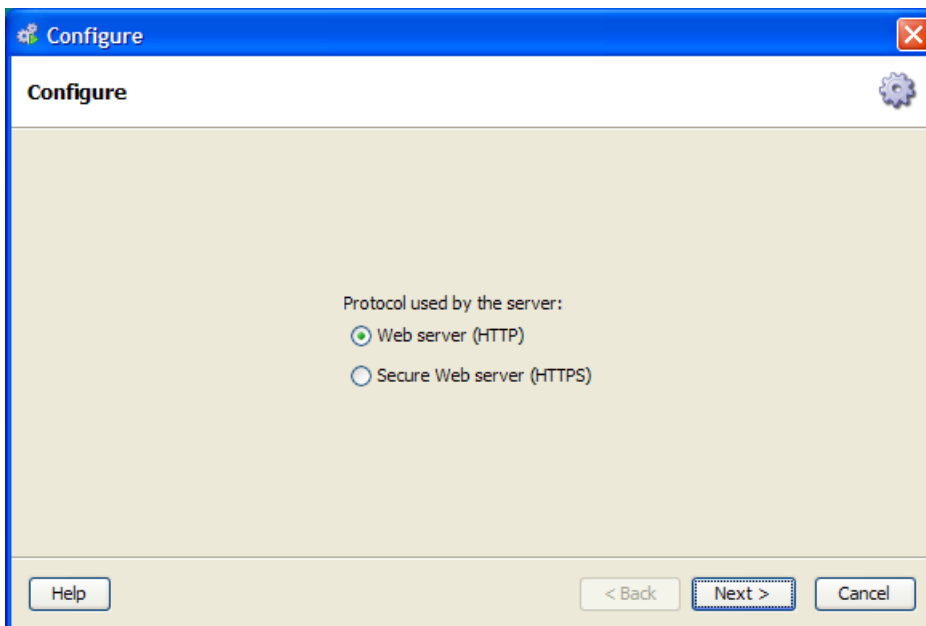


```
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>E:\src\runservlet\jetty\bin\controlapp -c
Initializing. Please wait...
Command:
1 - Stop
2 - Configure
3 - Start
2
Protocol used by the server:
1 - Web server <HTTP> [Default Choice]
2 - Secure Web server <HTTPS>
1
Port number used by the Web server: [22080]
```

Choose between a normal Web server (HTTP) and a secure Web server (HTTPS)

Choose "**Secure Web server (HTTPS)**" if you want all the data exchanged between Qizx XML Server and its clients (e.g. a Web browser) to be encrypted. Otherwise, choose "**Web server (HTTP)**".

Note that the confidentiality provided by option "**Secure Web server (HTTPS)**" is really not useful if you intend to operate Qizx XML Server on your local network.



Choose between a self-signed server certificate and an actual server certificate

If you have selected option "**Secure Web server (HTTPS)**" during the [preceding configuration step](#), you'll now have to specify which kind of server certificate to use.

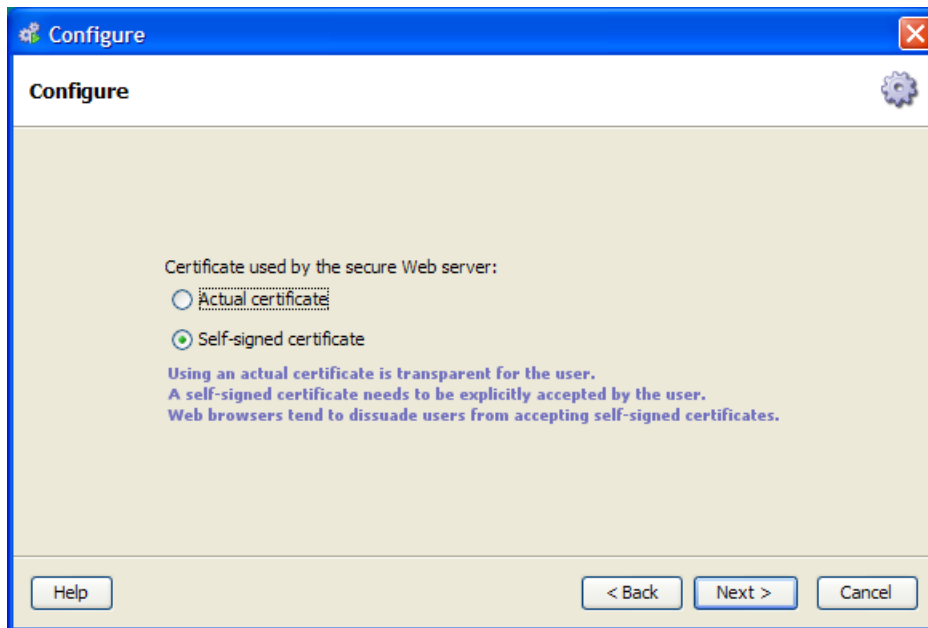
A *server certificate* fulfills two purposes:

1. Encrypt the data exchanged between the client and the server.

2. Let the user be sure that she/he is connecting to the right server (i.e. not a malware impersonating the right server).

An "actual certificate" fulfills both purposes. However you'll need to purchase such certificate from a *certificate authority* such as [VeriSign](#), [Thawte Digital Certificates](#), [Comodo](#), etc.

A self-signed certificate is free because it is automatically generated by `qizxserver`. It is only useful to encrypt the data exchanged between the client and the server. On the other hand, some clients may refuse to connect to servers presenting a self-signed certificate. Also, most Web browsers will strongly discourage the user from connecting to a server presenting a self-signed certificate. (They'll do this by showing the user pretty scary messages.)



Specify an existing server certificate

If you have selected option "Actual certificate" during the [preceding configuration step](#), you'll now have to specify which server certificate to use.

You'll need to specify:

- The file containing your certificate. This file is a *keystore* created using the `keytool` command-line utility which is part of all Java™ distributions.
- The type of the keystore. This type is almost always `JKS`. Another keystore type is `PKCS12`.
- The password used to protect both the keystore and the server certificate found in the keystore.
- The alias — a short name — identifying the server certificate in the keystore. This is needed because, though not recommended, a keystore may contain several entries.

The screenshot shows a 'Configure' dialog box with a blue title bar and a gear icon in the top right. The main area is titled 'Actual certificate'. It contains several input fields: 'Filename of the key store containing your certificate:' with the value 'C:\Documents and Settings\john\keystore' and a folder icon; 'Type of the key store:' with the value 'JKS'; 'Password used to protect both the key store and your certificate:' and 'Confirm password:' both with masked dots; and 'Alias identifying your certificate:' with the value 'gizmo'. At the bottom, there are buttons for 'Help', '< Back', 'Next >', and 'Cancel'.

Generate a self-signed server certificate

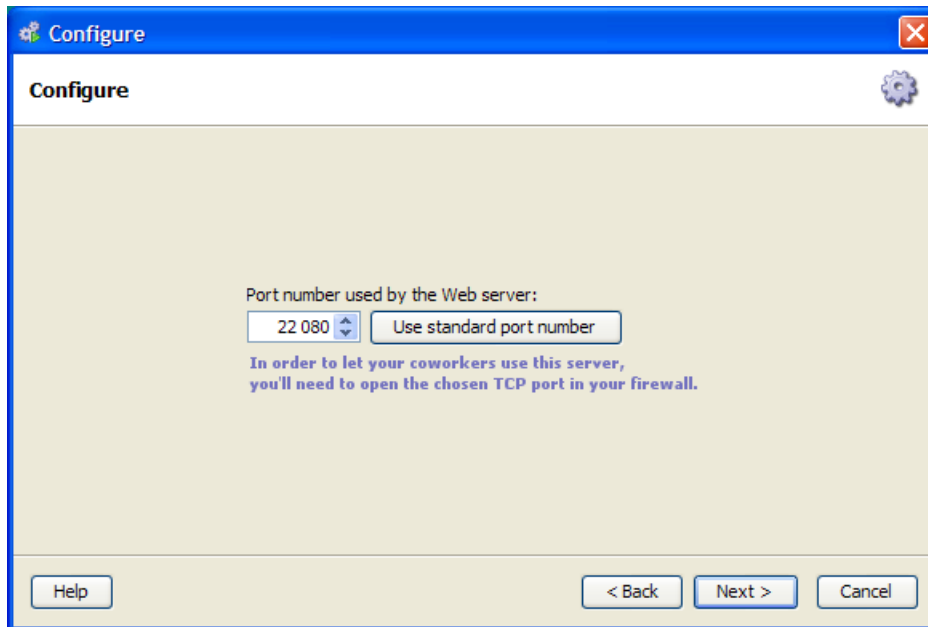
If you have selected option **"Self-signed certificate"** during the [preceding configuration step](#), you'll now have to specify the signer of the server certificate to be generated by qizxserver.

The only required field is the first one, the common name (CN) of the signer of the certificate. In the case of a server certificate, this common name must be the host name (e.g. `gizmo.acme.com`) or the IP address (e.g. `192.168.1.96`) of your machine.

The screenshot shows a 'Configure' dialog box with a blue title bar and a gear icon in the top right. The main area is titled 'Self-signed certificate'. It contains several input fields: 'Common name of the signer of the certificate (CN):' with the value 'gizmo.acme.com'; 'Organization unit (OU):' with the value 'Office Division'; 'Organization name (O):' with the value 'ACME Inc.'; 'Locality name (L):' with the value 'Ann Arbor'; 'State or province name (ST):' with the value 'Michigan'; and 'Country (C):' with the value 'US'. Below these fields, there is a note: 'The only required field is CN. This field must contain the host name of the Web server.' At the bottom, there are buttons for 'Help', '< Back', 'Next >', and 'Cancel'.

Specify a TCP port for the Web server

Clients (e.g. Web browsers) will see Qizx XML Server as a Web server. This Web server will run on the machine on which you have launched `qizxserver`. Many other network services are also running on this machine. Each of these network services is uniquely identified by its *TCP port*. Therefore you need to specify a TCP port — an integer between 2 and 65535 — which is not already used by another network service⁽¹⁾.



The "Use standard port number" button allows to specify the standard TCP port number associated to the chosen Web server protocol:

- Port 80 for "Web server (HTTP)".
- Port 443 for "Secure Web server (HTTPS)".

Using a standard port number is useful if you intend to operate Qizx XML Server on the Internet because:

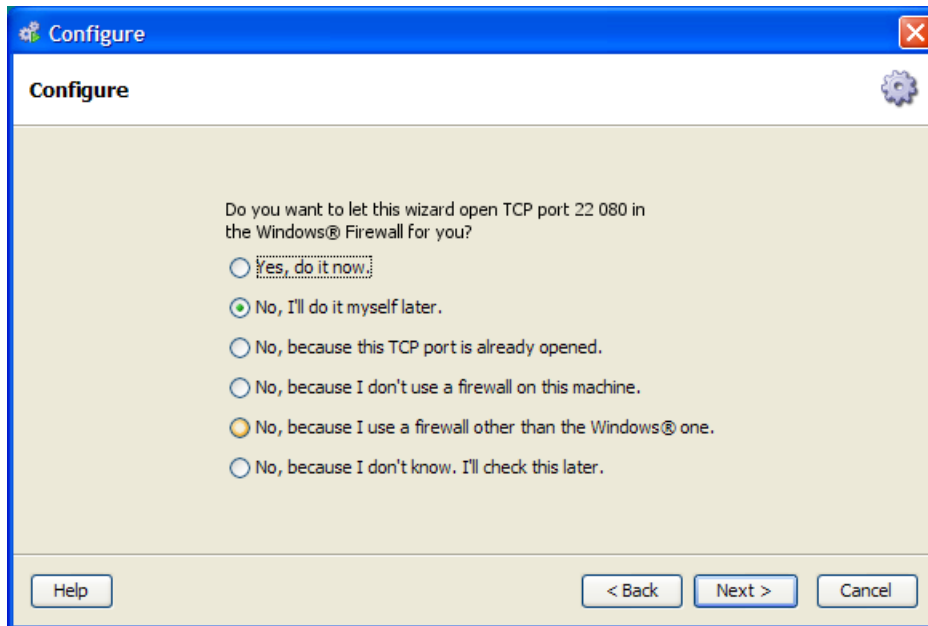
- Such ports are generally not blocked by enterprise firewalls or proxies.
- You don't need to type the port number in the address bar of your Web browser. Example: when the TCP port is 22080, you'll need to type "`http://gizmo.acme.com:22080/foo/bar`" in the address bar of your Web browser. When the TCP port is 80, you'll just need to type "`http://gizmo.acme.com/foo/bar`".

Open the server port in Windows® firewall

In order to let your coworkers use the Qizx XML Server which is running on your machine, the TCP port chosen during the preceding configuration step must not be blocked by the firewall running on your machine (if any).

This configuration step allows you to open this TCP port in Windows® built-in firewall. As you can see it in the screenshot below, this configuration step suggests several good reasons not to do this.

⁽¹⁾Do not worry, if you specify a TCP port which is already in use, the **Configure** wizard will instruct you to choose another port.



If you want `qizxserver` to open the chosen TCP port in Windows® built-in firewall, select **"Yes, do it now"**.



- Option **"Yes, do it now"** requires you to have administrator privileges. On Windows® Vista® and all following Windows® versions⁽²⁾, this means right-clicking on the icon of `qizxserver` and then selecting **"Run as Administrator"** from the popup menu.
- Choosing option **"Yes, do it now"** will not make your system unstable. If you really need your coworkers to access Qizx XML Server on your machine, do not hesitate to choose this option.
- If you do not select option **"Yes, do it now"**, Windows® itself may prompt you to "allow" one of the following programs: `java.exe`, `jsl32.exe` or `jsl64.exe`, just after Qizx XML Server is started. It's okay to accept what's suggested by Windows®. Doing this will have more or less the same effect as choosing option **"Yes, do it now"**.

Related information

↪ I can connect to Qizx XML Server from my machine (my machine is the one which is running Qizx XML Server), but my coworkers cannot connect to Qizx XML Server from their machines.

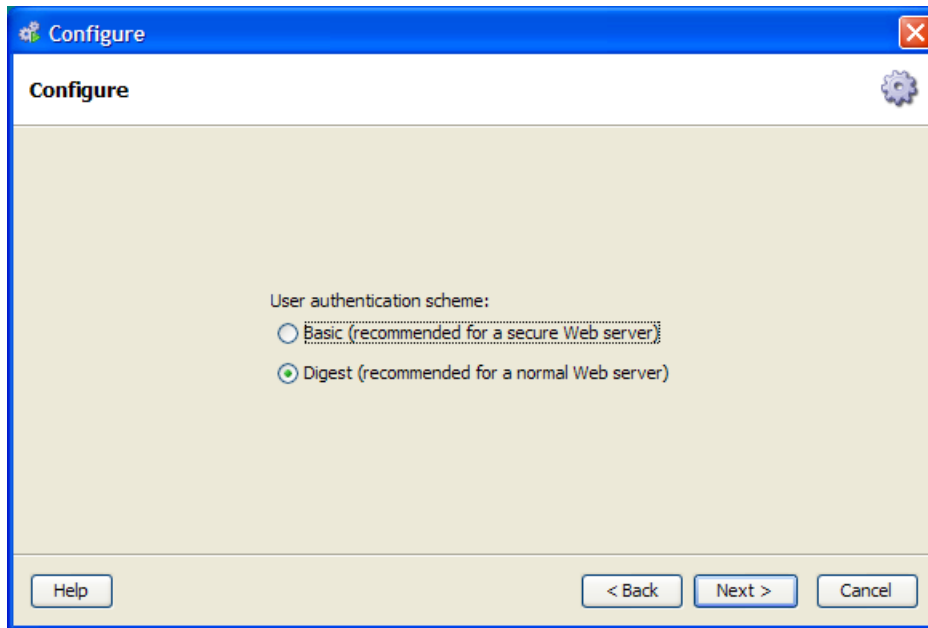
Choose an authentication scheme

A user of Qizx XML Server will be prompted by the client (e.g. the Web browser) for a username and a password each time she/he will use the service. An authentication scheme specifies how the credentials of the user are passed from the client to the server.

Choose **"Digest (recommended for a normal web server)"** if you have selected option **"Web server (HTTP)"** during the [preceding configuration step](#). The Digest authentication scheme transmits encrypted passwords to the server.

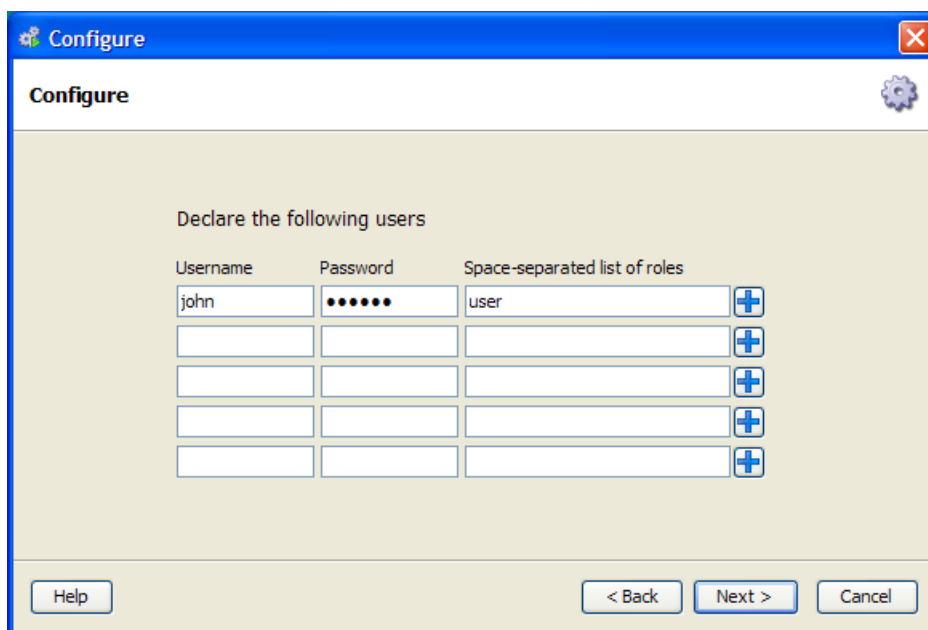
Choose **"Basic (recommended for a secure web server)"** if you have selected option **"Secure Web server (HTTPS)"** during the [preceding configuration step](#). The Basic authentication scheme transmits clear text passwords to the server. However in the case of HTTPS, the communication channel between the client and its server is itself encrypted.


⁽²⁾ Not on Windows XP where suffice to *be* an administrator.



Declare users

This step allows to declare up to five users of Qizx XML Server. Each user has a username, a password and one or more roles.



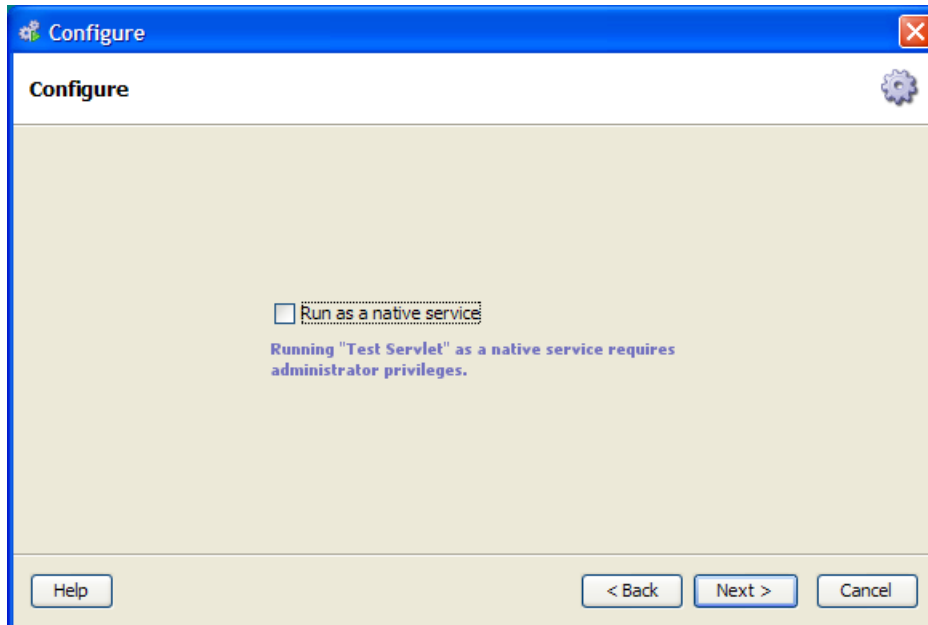
The  "Add a role to the list" button allows to append a role, which is well-known to Qizx XML Server, to the current list of roles. However the "Space-separated list of roles" field will accept any role you want.

Run Qizx XML Server as a Windows® native service

Running Qizx XML Server as a Windows® native service allows Qizx XML Server

- to keep running when you log off from Windows®,
- to be cleanly stopped and then to be automatically restarted each time you reboot your machine.

A Windows® native service is useful if you want to put Qizx XML Server in production. It is not really useful if you just want to evaluate Qizx XML Server.



If you want to run Qizx XML Server as a Windows® native service, check "**Run as native service**". Note that this requires you to have administrator privileges. On Windows® Vista® and all following Windows® versions⁽³⁾, this means right-clicking on the icon of `qizxserver` and then selecting "**Run as Administrator**" from the popup menu.

Running Qizx XML Server as a Unix daemon

Currently `qizxserver` does not allow to run Qizx XML Server as a *Unix daemon*. On the Mac and on Linux, you may use `qizxserver` to configure Qizx XML Server but after that, you'll need to write the `init` script by hand and then install this script in the proper place.

In order to startup Qizx XML Server, your `init` script needs to do something like this:

```
(cd directory_containing_start.jar; java -Xss2m -Xmx512m -jar start.jar)
```



Command `java -jar start.jar` will not work unless its current working directory contains `start.jar`.

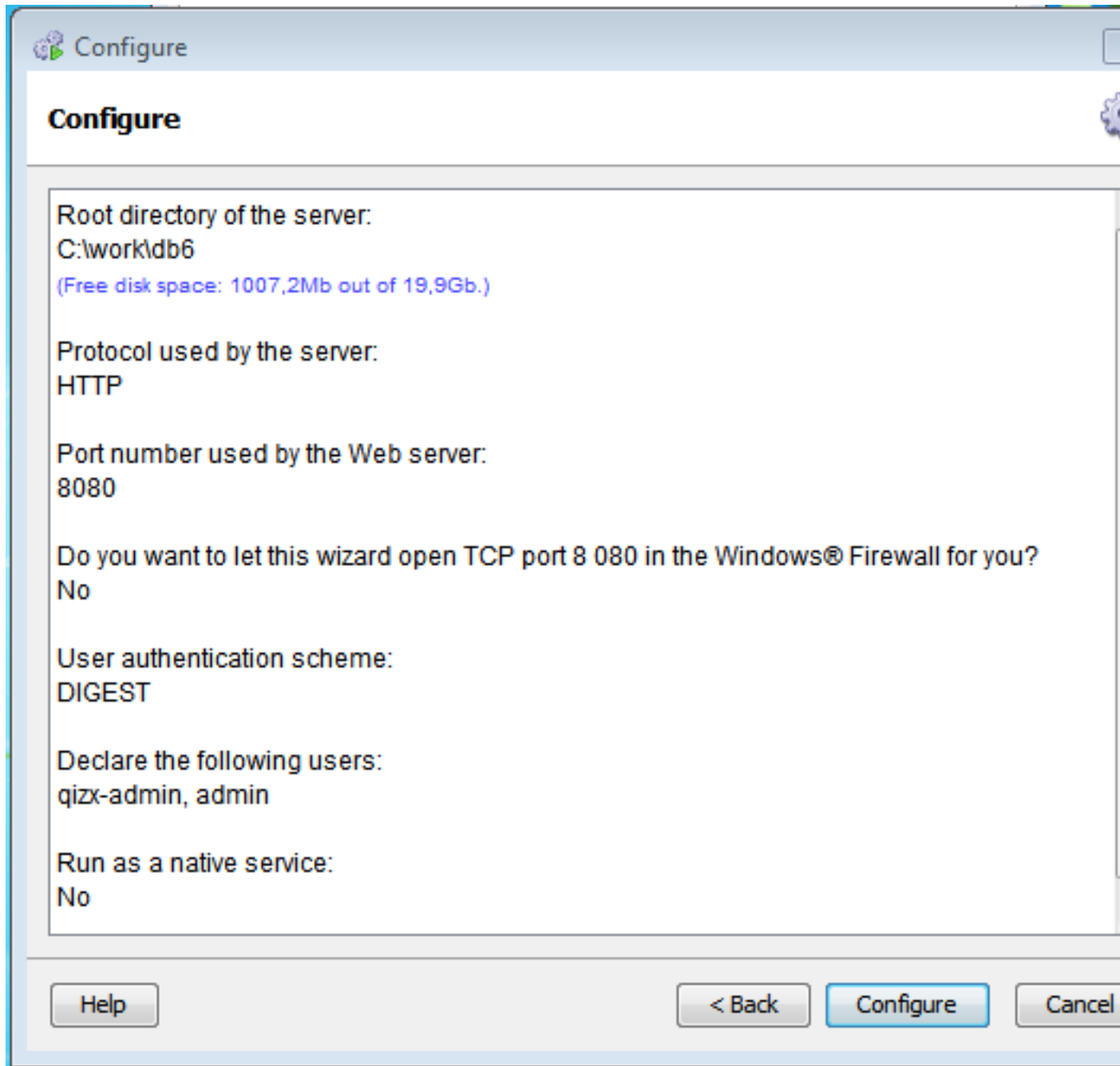
In order to shutdown Qizx XML Server, your `init` script needs to do something like this:

```
(cd directory_containing_start.jar; java -Xss2m -Xmx512m -jar start.jar --stop)
```

Review what you have specified

This configuration step, the final one, lets you review what you have specified during all the preceding steps. Click **Configure** to confirm that you really want to create or modify the configuration of Qizx XML Server.

⁽³⁾Not on Windows XP where suffice to *be* an administrator.



The screenshot shows a Windows-style configuration window titled "Configure". The window has a light blue header bar with the title and a gear icon. The main content area is white and contains several configuration options, each with a label and a text input field. The options are: "Root directory of the server:" with the value "C:\work\db6" and a note "(Free disk space: 1007,2Mb out of 19,9Gb.)"; "Protocol used by the server:" with the value "HTTP"; "Port number used by the Web server:" with the value "8080"; "Do you want to let this wizard open TCP port 8 080 in the Windows® Firewall for you?" with the value "No"; "User authentication scheme:" with the value "DIGEST"; "Declare the following users:" with the value "qizx-admin, admin"; and "Run as a native service:" with the value "No". At the bottom of the window, there are four buttons: "Help", "< Back", "Configure", and "Cancel".

Configure

Root directory of the server:
C:\work\db6
(Free disk space: 1007,2Mb out of 19,9Gb.)

Protocol used by the server:
HTTP

Port number used by the Web server:
8080

Do you want to let this wizard open TCP port 8 080 in the Windows® Firewall for you?
No

User authentication scheme:
DIGEST

Declare the following users:
qizx-admin, admin

Run as a native service:
No

Help < Back Configure Cancel

Troubleshooting



The workarounds described below apply mainly to Windows®.

Application `qizxserver` fails to configure Qizx XML Server.

This could be a problem of insufficient privileges.

When you run `qizxserver`, you must have enough privileges to read and write any file or directory belonging to the distribution of Qizx XML Server.

You'll need to have the privileges of an administrator if you want `qizxserver` to [open the server port in Windows built-in firewall](#) or to [run Qizx XML Server as a Windows native service](#).

The cure is generally to re-run `qizxserver` but this time, with the privileges of an administrator.

On Windows XP, suffice for your user account to be an administrator account.

On Windows Vista and subsequent versions, not only your user account must be an administrator account but also, you need to right click on the icon of `qizxserver` and pick **"Run as Administrator"** from Windows popup menu.

Qizx XML Server is up and running, but `qizxserver` sees it as being stopped (the Stop button is disabled, the Configure and Start buttons are enabled).

Here's what happens: when you [choose the server port](#), a shutdown port is automatically chosen for you. This shutdown port is the first free port nearby the server port. For example, if the server port is 8080, the shutdown port will often be 8079, 8081 or 8082. This shutdown port is probed to determine whether Qizx XML Server is running. This issue occurs when `qizxserver` picks a free shutdown port and when, once started, Qizx XML Server fails to open it.

First you need to stop Qizx XML Server by hand in order to reconfigure it.

If Qizx XML Server is running as a Windows® native service, stop the service by using Windows Service Administrative Tool. Alternatively execute `net stop "qizxserver"` using a command prompt.

If Qizx XML Server is running as a normal program, terminate it using the Windows Task Manager.

Now you need to re-run `qizxserver` in order to reconfigure Qizx XML Server. This time, choose a server port belonging to a very different range. For example, if the first time you have chosen 49152, this time choose 10000.

I can connect to Qizx XML Server from my machine (my machine is the one which is running Qizx XML Server), but my coworkers cannot connect to Qizx XML Server from their machines.

First try to connect using the IP address of the server rather than its host name. That is, specify something like `http://192.168.1.96:8080/foo/bar` rather than something like `http://speedo:8080/foo/bar`.

On Windows, the IP address of your machine may be determined as follows: open a command prompt, type `ipconfig /all` then press Enter. This command prints a lot of information on the console. Look for "IP Address".

On Windows, host names are often resolved to IP addresses using the NetBIOS name resolution rather than using a real DNS. When this is the case, there may be interoperability problems between machines running Windows XP and machines running Windows Vista/7 belonging to the same local network.

If specifying an IP address does not solve the problem, then a firewall running on your machine is almost certainly blocking [the port which has been chosen for the server](#).

On Windows, re-run `qizxserver` as an administrator, stop Qizx XML Server, reconfigure it, and this time make sure to [select option "Yes, do it now"](#).

If you are using a firewall other than Windows built-in one, then you are on your own to open a port in this firewall.

My coworkers can connect to Qizx XML Server from their machines, but this connection is quite slow (takes more than 1 minute to be established).

Try to connect using the IP address of the server rather than its host name. That is, specify something like `http://192.168.1.96:8080/foo/bar` rather than something like `http://speedo:8080/foo/bar`.

On Windows, the IP address of your machine may be determined as follows: open a command prompt, type `"ipconfig /all"` then press Enter. This command prints a lot of information on the console. Look for "IP Address".

If doing this solves the problem, then the culprit is the NetBIOS name resolution which can be quite slow. You may consider switching to a `hosts` file based resolution or to a real DNS.