

Wakanda Server Administration

Managing Wakanda Server

Configuring the Connection between the Studio and the Server

You can manage Wakanda Server from Wakanda Studio. The actions that you are allowed to do depend on the way the two applications are connected:

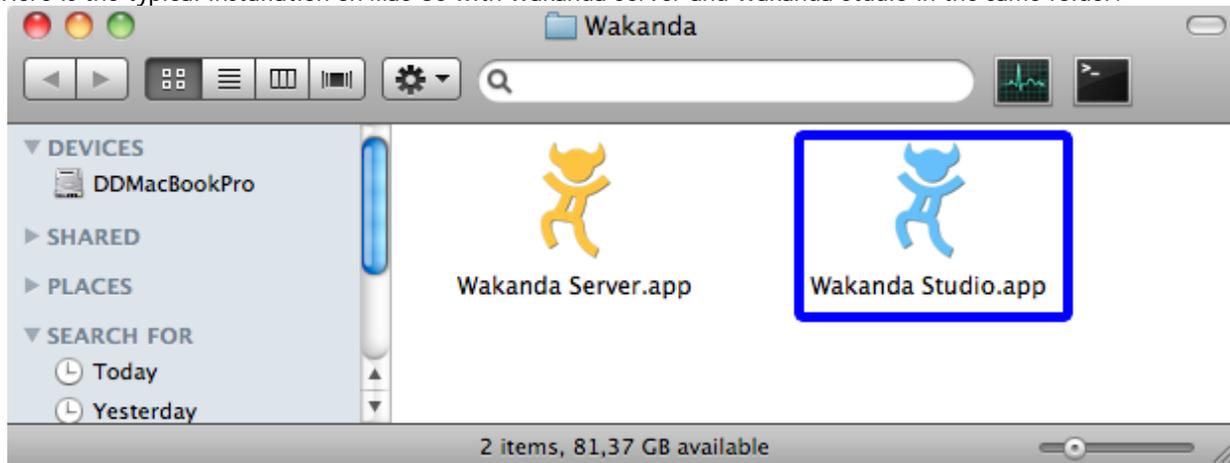
- **Local connection** (both Wakanda Server and Wakanda Studio are running on the same computer and share the same solution on the file system). In this case, you can:
 - Start and stop the Wakanda Server
 - Load a solution
 - Open and close a debug connection
- **Remote connection** (Wakanda Server and Wakanda Studio are not running on the same computer). Server and Studio must use the same solution on each machine. You need to make sure that both solutions are identical (for example, using standard synchronization tools).
In this case, the only action available from the Studio is:
 - Open and close a debug connection

Configuring a Local Connection

To be able to open a local connection to Wakanda Server from Wakanda Studio (both applications are running on the same machine), you have two possibilities:

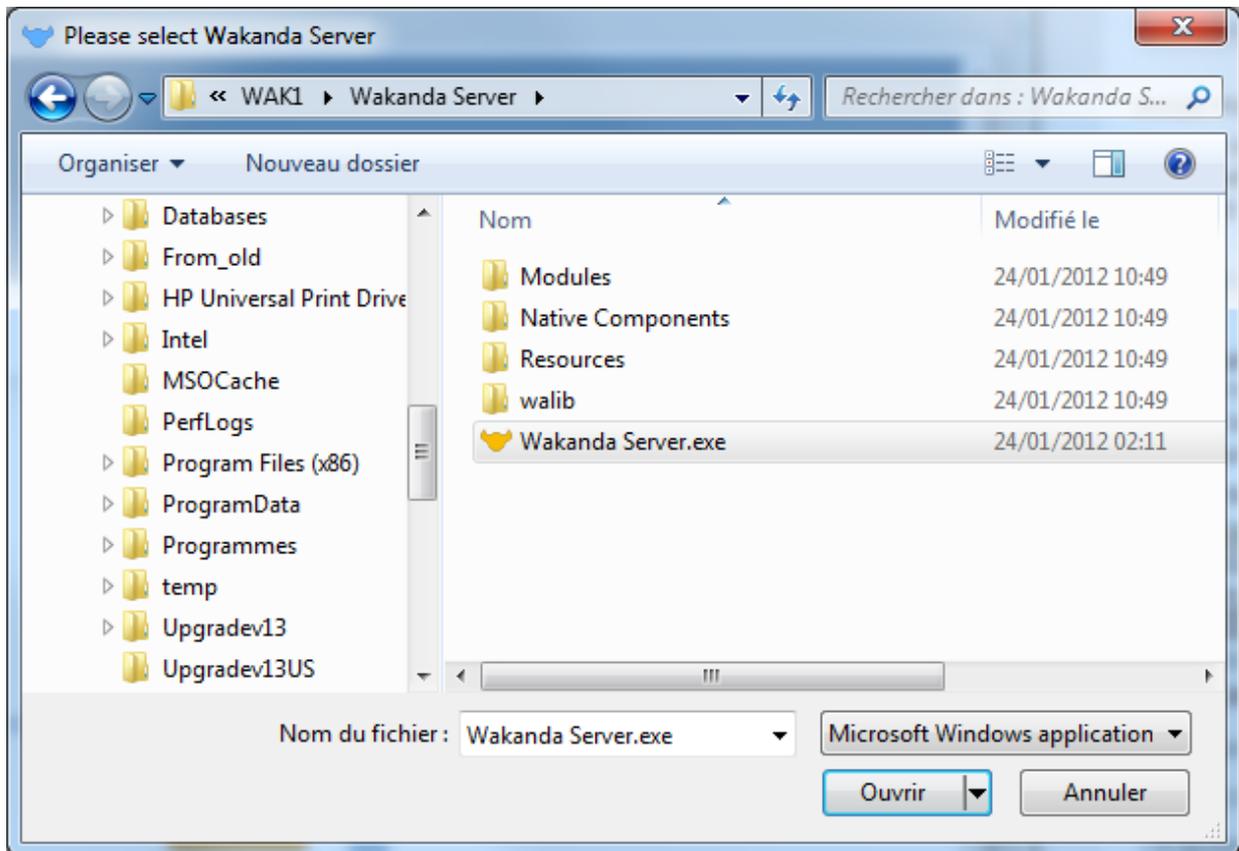
- Install Wakanda Server and Wakanda Studio subfolders (*packages* on Mac OS) in the same folder and name them "Wakanda Server" and "Wakanda Studio".

Here is the typical installation on Mac OS with Wakanda Server and Wakanda Studio in the same folder:



In this case, both applications will be connected automatically (unless another "preferential server" has been set for a solution, see [Defining a Preferential Server](#)).

- Install Wakanda Server and Wakanda Studio subfolders wherever you want on the same machine and launch Wakanda Studio. In this case, the first time that the Wakanda Studio will need to launch the server, a standard open dialog box will be displayed, allowing you to designate the Wakanda Server application:



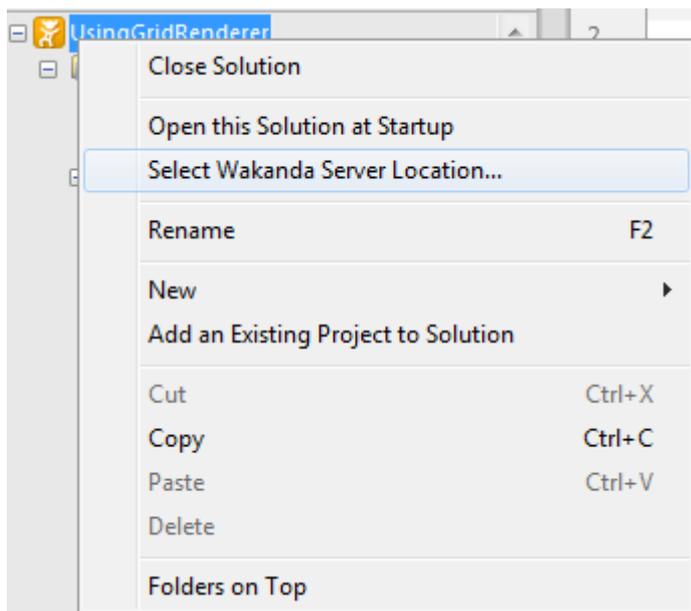
Connecting to a Remote Server

You can connect to a remote Wakanda Server from Wakanda Studio (applications are running on different machines). This functionality only allows you to open a debug connection with the server. You can trace the code execution at runtime and add/remove break points.

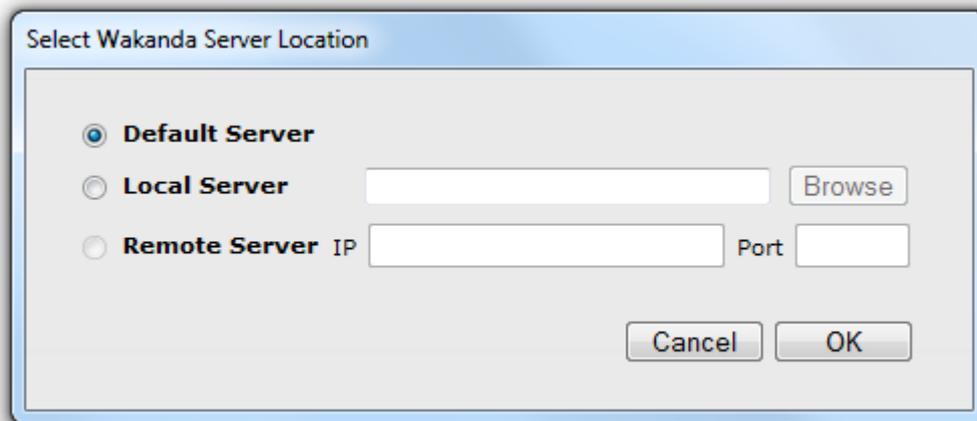
Defining a Preferential Server

You can define a "preferential server" (local or remote) for a given solution. Once set, the preferential server is stored in the solution settings and will be used by default with the solution.

To define a preferential server for a solution, right-click on the solution name in the Explorer area and choose **Select Wakanda Server Location...** in the contextual menu:



The following dialog box is displayed:



Three options are available to define the preferential server:

- **Default Server:** use a local Wakanda Server located beside the Wakanda Studio on the same machine (see [Configuring a Local Connection](#))
- **Local Server:** use a local Wakanda Server but not the Default Server. You need to define the location of the Wakanda Server application file on your disk by clicking the **Browse** button.
- **Remote Server:** use the remote Wakanda Server to which Wakanda Studio is currently connected through a debug connection (see [Connecting to a Remote Server](#)). This option is not available if no debug connection is currently open.

Starting and Stopping Wakanda Server

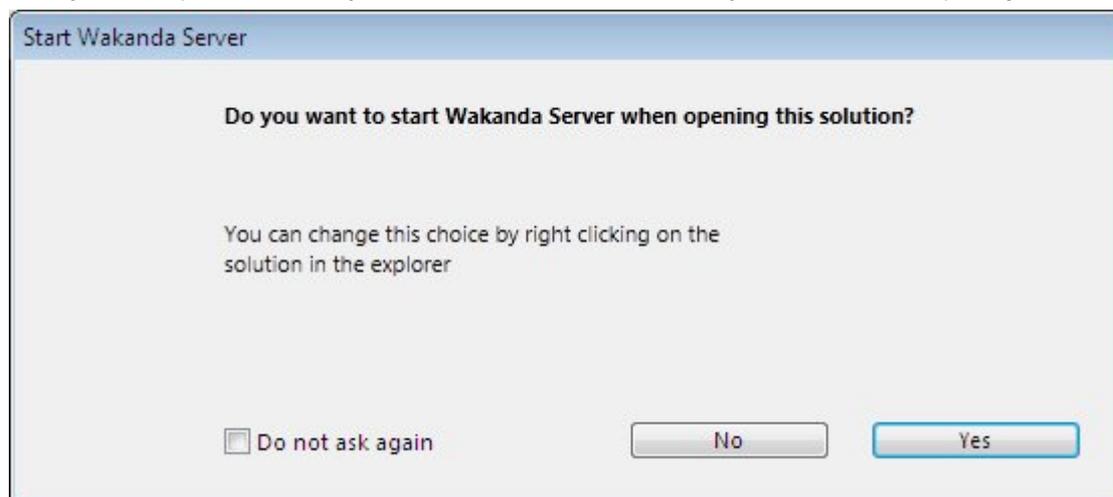
In Wakanda Studio, specific features allow you to:

- Start and stop the Wakanda Server automatically when opening or closing a solution
- Start and stop the Wakanda Server manually

Note: For these features to work, the Wakanda Studio must know the current location of the Server. For more information, please refer to the [Configuring the Connection between the Studio and the Server](#) section.

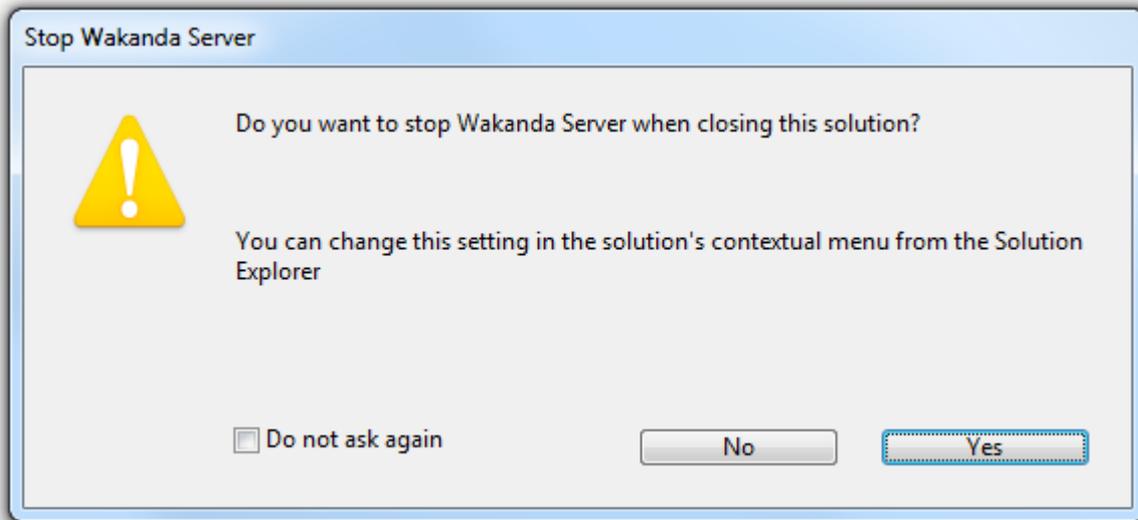
Automatic Handling

When you first open a solution, you are asked to start automatically the server when opening the solution in the Studio:



- If you click **Yes**, the server will be automatically launched when you open the solution
- If you click **No**, the server will not be automatically launched when you open the solution. You will need to launch it manually.

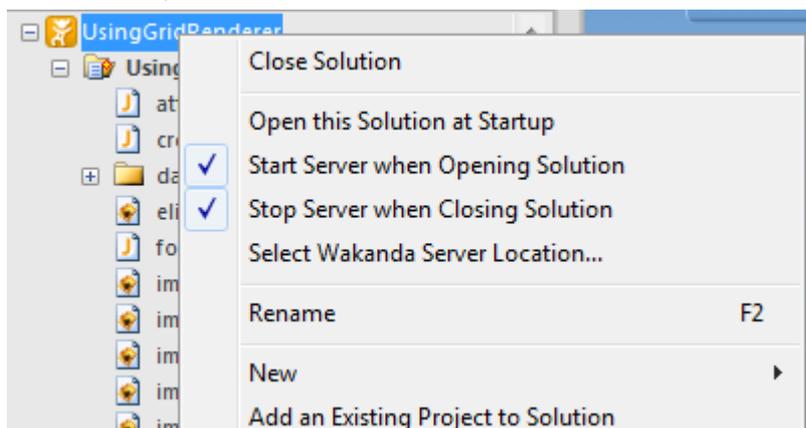
The same setting can be set when you close the solution. The following dialog box is displayed:



- If you click **Yes**, the server will be automatically stopped when you close the solution
- If you click **No**, the server will not be automatically stopped when you close the solution. You will need to stop it manually.

If you check "Do not ask again", the corresponding dialog will not be displayed anymore for any solution.

You can also manage the automatic opening and closing of the Wakanda Server for each solution using the contextual menu in the explorer:



Manual Handling

You can start and stop the Wakanda Server at any time when the Wakanda Studio is launched and when a valid solution is open.

To start the server, you select **Start Solution Server** in the **Run** menu or click on the Start button:



Once the server is running, you can select **Stop Solution Server** in the **Run** menu or click on the Stop button:



Launching Wakanda Server using a Command Line

You can use a command line to launch the Wakanda Server on all platforms (Windows, Mac OS and Linux). As parameters, you can pass the solution to open as well as the ServerAdmin project HTTP port.

Thanks to this feature, you can open a solution automatically, for example at startup. You can also launch several Wakanda servers with different default administration projects.

The syntax is:

```
Wakanda_server_name [solution_path] [--admin-port=http_port_number]
```

where:

- *Wakanda_server_name* is the full pathname of the server application ("Wakanda Server.exe" on Windows and "Wakanda Server.app" on Mac OS)
- *solution_path* is the full pathname of the solution to open. The path should be expressed in the system syntax. If you do not pass this parameter, the default solution is opened.
- *--admin-port=http_port_number* is an optional parameter that sets the HTTP port number of the built-in ServerAdmin project. The ServerAdmin project is the default administration project. It is published on port 8080 by default. By setting a different value, you can publish this default project on another port, allowing you to launch several Wakanda Servers running the default administration project. The HTTP port that you set is used during the entire server session, even if another solution is opened. If the opened solution already contains an administration project (project with key administrator="true" in the *myproject.waSettings* file), the *--admin-port* parameter is ignored.

Warning: The shells do not accept spaces or / symbols in command lines. To avoid interpretation errors, insert parameters between double quotes "" (see examples).

Examples

- (Windows) Launching the Wakanda server and opening the default solution (containing the ServerAdmin project, published on port 8080)

```
"C:\Wakanda\Wakanda Server.exe"
```

- (Windows) Launching the Wakanda server, opening the default solution (containing the ServerAdmin project) and publishing the ServerAdmin project on the HTTP port 8090

```
"C:\Wakanda\Wakanda Server.exe" "--admin-port=8090"
```

- (Windows) Launching the Wakanda server and opening the "invoices" solution. If this solution does not contain an administration project (administrator="false" in settings), the ServerAdmin project is published on the HTTP port 8090

```
"C:\Wakanda\Wakanda Server.exe" "C:\solutions\invoices.waSolution" "--admin-port=8090"
```

- (Mac OS) Launching the Wakanda server and opening the "invoices" solution. If this solution does not contain an administration project (administrator="false" in settings), the ServerAdmin project is used and published on the default HTTP port (8080)

```
"Mac HD:Applications:Wakanda:Wakanda Server.app" "Mac HD:solutions:invoices.waSolution"
```

- (Linux) Launching the Wakanda server and opening the default solution (containing the ServerAdmin project, published on port 8080)

```
./Wakanda
```

- (Linux) Launching the Wakanda server and opening the "invoices" solution. If this solution does not contain an administration project (administrator="false" in settings), the ServerAdmin project is used and published on the HTTP port 8080

```
./Wakanda /home/AdminUserName/invoices.waSolution
```

Configuring secure connections in Wakanda Server (SSL/TLS)

The Wakanda HTTP server provides support of secure connections through HTTPS. To implement secure connections in Wakanda, you need to:

- get a SSL/TLS certificate
- install the appropriate files in your Wakanda project folder
- configure the relevant settings for your project

How to get a certificate?

A Wakanda server working in secure mode means that you need a digital SSL/TLS certificate. This certificate references various information such as the site ID as well as the public key used to communicate with the site. This certificate is transmitted to the Web clients (browsers) connecting to this site. Once the certificate has been identified and validated, the communication is made in secure mode.

SSL/TLS certificates can be delivered by a Certification Authority or be self-signed. A well known Certification Authority (such as Verisign® or Thawte®) will provide certificates that will be authorized by many browsers (i.e. automatically validated), but the price will be expensive.

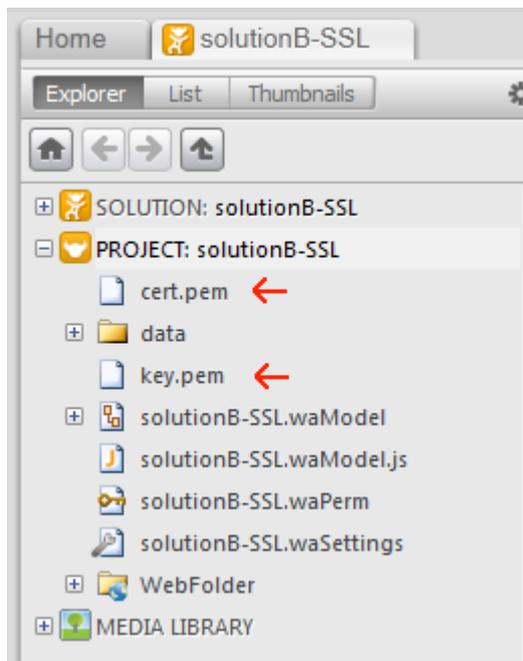
Note however that the certificate only manages the 'authentication' part of the secured connection. When a certificate is not referenced in the browser's properties (this is the case for self-generated certificates), the user is just asked to validate it manually. Once the certificate is accepted, the connection is established in secure mode and is encrypted. During the development phase, it is a good idea to use self-signed certificates.

Installing SSL/TLS files in Wakanda

If you want to use the SSL/TLS protocol in your Wakanda application, the following files must be installed in the project folder:

- **key.pem**: This is the file containing the private key.
If necessary, rename your private key file name manually to "key.pem".
- **cert.pem**: This is the file containing the certificate.
It must be named "cert.pem" and be in PKCS format. If you get a file in a different format, you may need to convert it into .pem format (see above).

Both files must be installed at the first level of your project folder:



Note: You can copy the cert.pem file to another location, provided you enter its actual path in the Certificate Path parameter in the .waSettings file of the project.

Configuring the Settings

In order for HTTPS connections to be accepted by the Wakanda Web server, you must make sure that SSL/TLS is activated and configured in the project settings.

Several parameters are accessible in the "Secure Connections" area of the `{projectName}.waSettings` file:

Secure Connections (SSL -TLS)

Enable secure connections
Port Number: + -
Certificate Path:

 Mandatory secure connections

- **Enable secure connections:** By default, secure connections are allowed. You can uncheck this option if you do not want to use HTTPS functionality with your Web server, or if another Web server allowing secure connections is operating on the same machine.
- **Port Number:** The TCP port dedicated to SSL/TLS data exchange is 443 by default. This port number can be modified in order, for example, to reinforce the security of the Web server or to resolve conflicts on the machine. The TCP port 443 is used for standard mode Web server connections.
- **Certificate Folder Path:** This area allows you to use an alternate location for your `cert.pem` and `key.pem` files. By default, if this area is left blank, Wakanda will look for these files at the root level of your project folder. To enter a custom path, click on the **Browse...** button and select the folder containing your `.pem` files. You can also type in a path manually in the entry area. The path can be absolute or relative to the project root folder. If you enter a relative path, it must not contain the root folder itself. For example, for the following project file organization:


```
{projectfolder}
  [sslFiles]
    cert.pem
    key.pem
```

 ... enter "sslFiles/" in the entry area.
- **Mandatory secure connections:** Check this option to force the use of the SSL protocol for all resources in the application. When checked, only HTTPS connections to the server will be allowed.

Configuring Hosting

The Wakanda HTTP Server supports multi-domain hosting, that is, a solution can contain several projects available through different domain names.

By default, Wakanda Server runs your different projects on different TCP ports, but you can easily configure your projects to support virtual multi-domain hosting.

Default Configuration for Multiple Projects

For each project in a solution, you can specify:

- a hostname or IP address
- a port number

The basic rule is that for any project in the same solution, at least one of these parameters must be different so that the server can know to which project the requests should be routed.

By default when you create a new project in a new solution, it is assigned to the port 8081 and Wakanda Server increments this number for each project added in the solution. Since these ports are different from the default port (80), this means that you need to add the port number to the hostname in the URL. For example:
`http://127.0.0.1:8081`

The reasons that Wakanda uses these specific port numbers are:

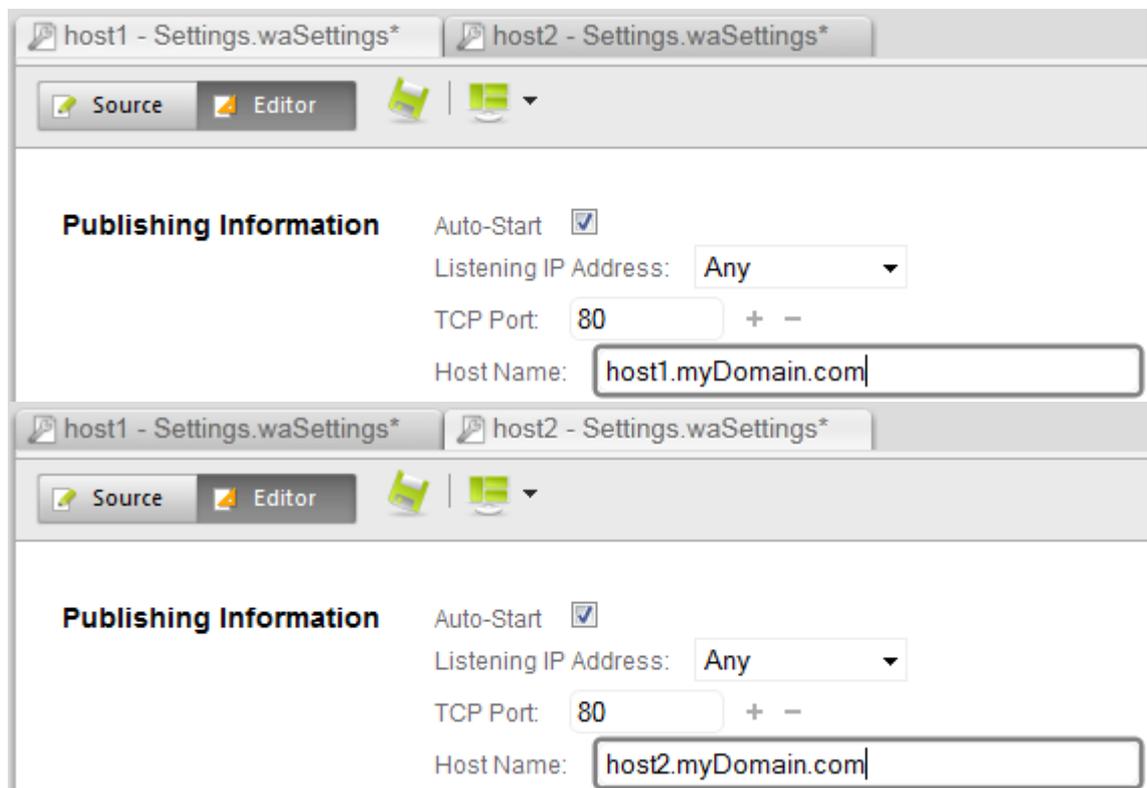
- to prevent conflict with another HTTP server on the developer's machine
- to make all the projects accessible via different URLs without having to specify (and buy) a domain name
- to not rely on an internet connection to reach the server using a domain name

However, in a production context, you would use the TCP port 80 for incoming HTTP connections, even if your Wakanda Server runs several projects. In this case, you need to configure the virtual multi-hosting features of the Wakanda server.

Creating Virtual Hosts

This paragraph explains the steps required to configure a single Wakanda Server and the DNS to support several virtual hosts. Below you will find a screencast that illustrates the whole process.

1. Create a solution containing at least two projects (for example `host1` and `host2`).
2. For each project:
 - select the same listening TCP port. For example, you can choose the port **8081**. For a production server, you would choose the port **80**, which is the default HTTP port.
 - enter a different hostname, for example `host1.myDomain.com` and `host2.myDomain.com`. Both parameters are to be defined in the `.waSettings` file of each project:



3. Configure the DNS accordingly.
In this case, you will configure `host1.myDomain.com` and `host2.myDomain.com` to route requests to the IP address of your server.
Note that domain names can be tested on a developer machine, even in offline mode. You just need to configure the local "hosts" file (for more information, please refer to <http://support.ecenica.com/domain-names/edit-hosts-file-mac-windows/> or watch [this part](#) of the screencast below).
4. To connect to your projects, enter in a browser:
 - `host1.myDomain.com` (if using default port 80) or `host1.myDomain.com:8081` (if using specific port) to connect to the first project
 - `host2.myDomain.com` (if using default port 80) or `host2.myDomain.com:8081` (if using specific port) to connect to the second project

The following screencast shows an example of the configuration of two projects running on port 8081 and tested locally:

Configuring Settings Files

Settings files are loaded by Wakanda Server and allows you to define many parameters for your Web application. Two main settings files are available:

- **{Solution Name}.waSettings** (located at the solution level): This XML file defines the settings for your solution.
- **{Project Name}.waSettings** (located at the project level): This XML file defines the customizable settings for your project.

Solution Settings File

Solution Settings File

The Solution Settings file for your Wakanda solution is an XML file containing the following properties:

```
<settings>
  <solution>
    <serverStartup stopIfProjectFails="true"/>
    <directory authenticationType="basic"/>
  </solution>
  <database
    adaptiveCache="false"
    memoryForOtherApplications="512"
    memoryForCache="50"
    minimumSize="100"
    maximumSize="400"
    fixedSize="200"
    keepCacheInMemory="true"
    flushDataCacheInterval="15"/>
</settings>
```

In the development branch of Wakanda, we have added a graphical interface to modify some of the settings available in this file.

Project Settings File

Project Settings File

The "*{Project Name}.waSettings*" file is an XML file that contains the settings for your project.

In the development branch of Wakanda, we have added a graphical interface to modify some of the settings available in this file.