

# Reliance 4

**DATA SERVERS**  



# Reliance 4

**DATA SERVERS**



**© 2019 GEOVAP, spol. s r. o. All rights reserved.**

GEOVAP, spol. s r. o.  
Cechovo nabrezi 1790  
530 03 Pardubice  
Czech Republic  
+420 466 024 618  
<http://www.geovap.cz>

Products that are referred to in this document may be trademarks and/or registered trademarks of the respective owners.

While every precaution has been taken in the preparation of this document, GEOVAP assumes no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall GEOVAP be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

# Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>1</b>
<b>2</b>	<b>Data Server Startup</b> .....	<b>2</b>
<b>2.1</b>	<b>Reliance Control Server</b> .....	<b>3</b>
<b>2.2</b>	<b>Reliance Server (Windows Service)</b> .....	<b>4</b>
<b>2.2.1</b>	<b>Reliance Server (Multiple Service Instances)</b> .....	<b>5</b>
<b>3</b>	<b>Data Server Web Pages</b> .....	<b>8</b>
<b>3.1</b>	<b>Access to Data Server Web Pages</b> .....	<b>9</b>
<b>3.2</b>	<b>Web Pages Structure</b> .....	<b>10</b>
<b>3.3</b>	<b>Project Web Page</b> .....	<b>12</b>
<b>3.4</b>	<b>Thin Clients Web Page</b> .....	<b>13</b>
<b>3.5</b>	<b>Administration Web Page</b> .....	<b>14</b>
<b>3.6</b>	<b>API Web Page</b> .....	<b>16</b>
<b>4</b>	<b>Web Server Settings</b> .....	<b>17</b>
<b>4.1</b>	<b>HTTPS and Certificates</b> .....	<b>18</b>
<b>4.2</b>	<b>MIME</b> .....	<b>22</b>
<b>4.3</b>	<b>User-Agent</b> .....	<b>23</b>
<b>4.4</b>	<b>Root Directory</b> .....	<b>24</b>
<b>5</b>	<b>Other</b> .....	<b>25</b>
<b>6</b>	<b>Glossary</b> .....	<b>26</b>

# 1 Introduction

**Reliance Control Server** and **Reliance Server** are runtime software modules collectively called *data servers*.

They provide data to **Reliance**'s other runtime software modules (including other data servers). Their built-in Web server allows the data servers to:

- Provide Web pages for browser users
- Provide **Reliance Smart Client**'s Web pages
- Provide data to **Reliance Web Client**
- Provide data to **Reliance OPC Server** (both UA and DA)
- Process external requests (e.g., receiving messages from IoT devices)
- Exchange data with third-party applications using APIs (SOAP and REST)

## 2 Data Server Startup

Starting a data server depends on whether it is **Reliance Control Server** or **Reliance Server**. A data server works properly only when a visualization project is loaded by this server. For example, **Reliance Control Server** can be started without a visualization project, which can be selected and loaded later.

## 2.1 Reliance Control Server

**Reliance Control Server** (R\_CtlSrv.exe) and the other visual runtime software modules – **Reliance View** (R\_View.exe) and **Reliance Control** (R\_Ctl.exe) – have the same startup rules. There are a number of ways of running a runtime software module (in our case, **Reliance Control Server**) and loading a visualization project:

### 1. By starting the runtime software and selecting the visualization project

Upon runtime software startup, run the visualization project by choosing the *File > Run Project* command. If multiple computers are defined within the project, the user will be prompted to select the computer on which the project should be running.

### 2. By choosing a command from Reliance Design

In the **Reliance Design** development environment, open the visualization project by choosing the *File > Open Project* command. Then, select **Reliance Control Server** in the *Project Options* dialog (*Project > Options > Runtime > Start and Termination*). To run the runtime software and load the visualization project, choose the *Project > Run* command.

### 3. From a command prompt (shortcut)

Every visual runtime software module can be started from a command prompt using the following syntax: `"exe_file" "main_project_file" "project_computer_name"`.

For example, to start the project named `Test` on the computer named `Server` in **Reliance Control Server**, the command line will be as follows:

```
"C:\Program Files\GE\GEOVAP\Reliance4\R_CtlSrv.exe" "C:\MyProjects\Test\Test.rp4" "Server"
```

If no computer is specified, it will be determined according to the same rule as in paragraph 1.

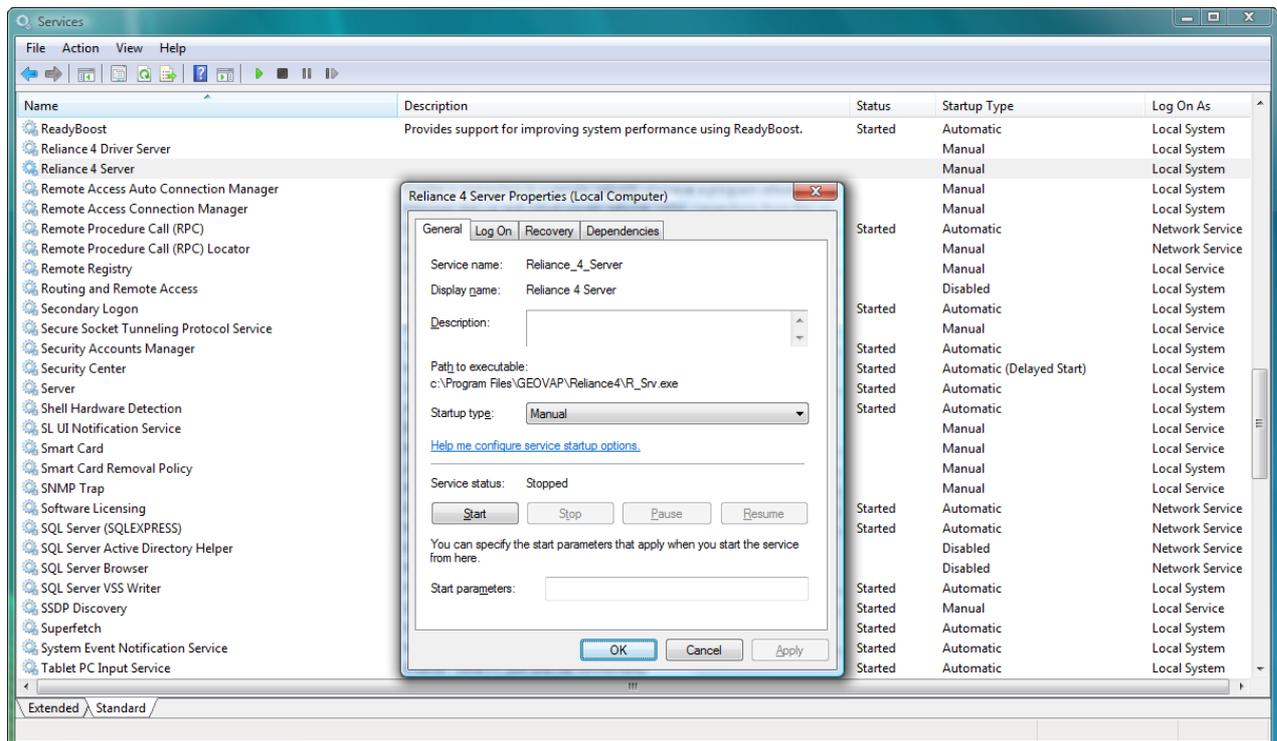
**Note:** The **Reliance Design** development environment allows automatically creating a shortcut with a command according to the above syntax – see the *Project > Create Shortcut* command.

## 2.2 Reliance Server (Windows Service)

**Reliance Server** (`R_Srv.exe`) is a non-visual runtime software module of **Reliance**. It runs as the so-called *Windows service* and is thus intended to run on computers without user assistance (usually on servers). In Microsoft Windows, a Windows service is a special program that runs as long as Windows is running and that is not in direct contact with the user (unlike common applications). A Windows service can be started at Windows startup before user log-on. The **Reliance Server** service must be registered to Windows before running it. To register the service, go to **Reliance Design** and choose the *Project > Register As Service* command.

After registering the service, the `R_Srv.ini` file is generated. It contains the path to the main project file and the computer name. This file is used to perform the registration itself as well as other service-related operations (startup, termination, unregistration). Typically, the file's location is `C:\Users\Public\Documents\GEOVAP\Reliance4\Settings\R_Srv.ini`.

If you want to check whether the license has been successfully registered, go to the Services control panel. The default name of the service is *Reliance\_4\_Server*.



**Reliance Server – Windows service**

Depending on the configuration during registering the service, you can choose to start the **Reliance Server** service manually. Should the service start automatically after Windows startup, set the *Startup type* to *Automatic*. The Services control panel allows performing the following service-related operations: *Start, Stop, Pause, Resume*. The service is registered to run under the *Local System* user account. If **Reliance Server** is a client of an OPC server, this OPC server must also operate as a service and under the same user account. The same applies to **Reliance Driver Server**. To change the account under which the service runs, use the Services control panel (*Properties > Log On*).

The status of all **Reliance** services can also be controlled directly from the **Reliance Design** development environment (*Environment > Windows Services*).

Another way of registering the license is to do it manually by running the `R_Srv_RegService.bat` file. Before that, however, you must edit the `R_Srv.ini` file:

```
[ Main]
Language=0
[ Parameters]
Project=C:\MyProjects\Project1\Project1.rp4
ProjectName=Project1
Computer=Computer1
ProjectAutoStart=0
```

When the registration is completed, batch files can be used to perform the same license-related operations as in the Services control panel. `R_Srv_StartService.bat` is used to start the **Reliance Server** service. To stop the service, run the `R_Srv_StopService.bat` file. If you want to unregister the service, the `R_Srv_UnRegService.bat` file must be run. All the batch files can be found in the `BatchFiles` folder, typically `C:\Program Files (x86)\GEOVAP\Reliance4\BatchFiles`.

**Note:** The `R_Srv_StartApp.bat` file runs **Reliance Server** as a common application, not as a service. After that, the **Reliance Server** icon appears in the taskbar tray. This way of starting **Reliance Server** can sometimes be used for debugging purposes.

### 2.2.1 Reliance Server (Multiple Service Instances)

All **Reliance** runtime software modules can run in multiple instances, which means that multiple runtime software modules can run at a time (each module with a different

visualization project). For **Reliance Server** (Windows service), the list of instances is defined in the `R_Srv.ini` file:

```
[ Main]
```

```
Language=0
```

```
[ Parameters]
```

```
Project=C: \MyProjects\Tannery\Tannery.rp4
```

```
ProjectName=Tannery
```

```
Computer=pc1
```

```
ProjectAutoStart=0
```

```
[ Parameters1]
```

```
Project=C: \MyProjects\Heating\Heating.rp4
```

```
ProjectName=Heating
```

```
Computer=Server1
```

```
ProjectAutoStart=0
```

```
[ Parameters2]
```

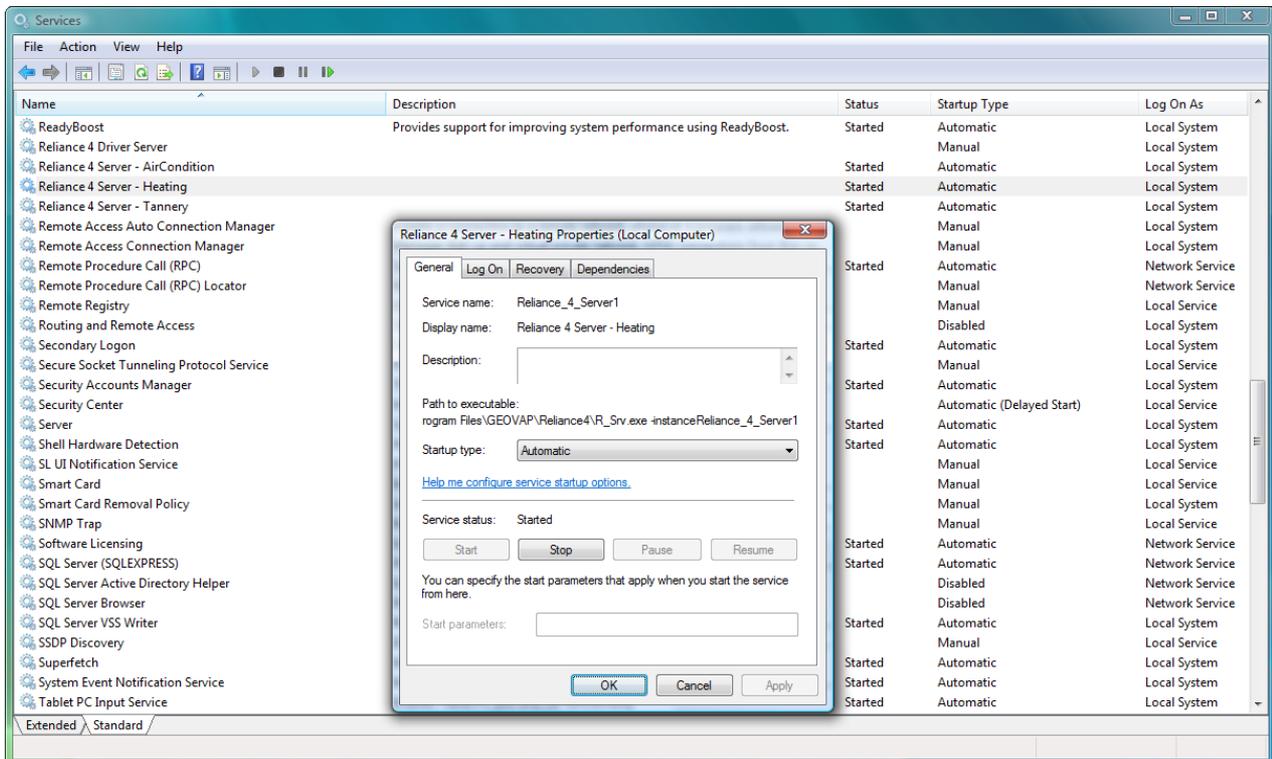
```
Project=C: \MyProjects\AirCondition\AirCondition.rp4
```

```
ProjectName=AirCondition
```

```
Computer=Computer1
```

```
ProjectAutoStart=0
```

`R_Srv_RegService.bat` and `R_Srv_UnRegService.bat` are batch files intended to register (by running the `R_Srv_RegService.bat` file) and unregister (by running the `R_Srv_UnRegService.bat` file) all instances of the **Reliance Server** service defined in the `R_Srv.ini` file.



## Multiple instances of Reliance Server

The name of each registered instance of the service in the Services control panel has the following format: *Reliance Server - <project\_name>*, where *<project\_name>* represents the visualization project name specified for the relevant instance of the service in the `R_Srv.ini` file as the `ProjectName` parameter's value.

### 3 Data Server Web Pages

The server Web pages are available at an address (URL) in the following format:

`http(s)://<computer_name>:<tcp_port_number>`, where

`<computer_name>` can be the computer's hostname, IP address, or Web address  
and

`<tcp_port_number>` is the built-in Web server's TCP port number.

The server Web pages' address can be:

<http://MyServer:40000> or <https://MyServer:40363> for a computer within a local network

and

<http://www.reliance.cz:40000> or <https://www.reliance.cz:40363> for a Web server.

You can change the built-in Web server's TCP port number through **Reliance Design's** Project Structure Manager (*computer > Web > Port number*). 40000 is the default value for HTTP and 40363 is the default value for HTTPS (secure communication). HTTP URLs use TCP port 80 by default. If the data server is to run on port 80, the port number doesn't necessarily be part of the address. The same applies to HTTPS, which uses port 443 by default.

The address format will be as follows:

`http(s)://<computer_name>`

For example:

<http://MyServer> or <https://MyServer>

**Note:** The default port number (40000 or 40363) is different from the standard port number (80 or 443) to avoid conflict with another Web server running on the same computer.

### 3.1 Access to Data Server Web Pages

You can open a data server's Web pages by entering the address (see the previous chapter) in a Web browser. The **Reliance Control Server** module's Web pages can be accessed from **Reliance Design** by choosing the *Tools > Server Web page* command. The pages open in the default Web browser using the name of a computer within a local network.

For example: <http://MyServer:40000> or <https://MyServer:40363>

**Note:** If both protocols (HTTP and HTTPS) are enabled in the project, the command will open the pages using HTTPS.

Before accessing the Web pages from another computer, make sure this remote computer allows access to the server under its name (e.g., from a command prompt using the `ping myserver` command). Also, the TCP port (40000 or 40363) must not be blocked, for example, by a firewall.

## 3.2 Web Pages Structure

The server Web pages are divided into sections. Each section has the following layout:

- Header – graphically indicates the data server type (**Reliance Server** or **Reliance Control Server**); in addition, it contains a link to the Welcome! page
- Menu – located on the left side, it contains links to individual pages
- Content area – fills the remaining space
- Footer

The pages have a responsive layout. It means that the display size adjusts to low-resolution devices to save as much space for the content area as possible. Specifically, the pages have a drop-down menu and their header is hidden on such devices. The controls and links are also designed for comfortable use on touch devices.

The menu contains the following commands:

### **Home**

Displays the Welcome! page. Depending on the project settings, custom content can be embedded in this Web page.

### **Čeština (Czech)**

Is used to switch to Czech.

### **Log On**

Displays the page where the user logs on. Only project-defined users can log on. Once logged on, the user is prompted to change the password.

### **Change Password**

Displays the page for changing the logged-on user's password. The password must meet the requirements for safe (strong) passwords specified in the project.

### **Log Off**

Logs off the logged-on user.

### **Project**

Displays the page that contains information on project-defined objects (devices, tags, current alarms/events, reports, custom reports) – see the chapter Project Web Page.

### **Thin Clients**

Displays the page designed to run the thin clients (**Reliance Web Client** and **Reliance Smart Client**) – see the chapter Thin Clients Web Page.

### **Administration**

Displays the page designed for data server administration – see the chapter Administration Web Page.

### **API**

Displays the page describing APIs for connecting third-party applications to the data server – see the chapter API Web Page.

### **Help**

Displays this help system as a Web page.

Some commands may not be available depending on:

- The project settings
- Whether or not the user is logged on
- The logged-on user's access rights

### 3.3 Project Web Page

The **Project** page is divided into four tabs:

#### **Devices**

Shows a list of devices. Clicking a device will show a list of the device's tags. For each tag, its alias, current value, time stamp, and quality are displayed. The list updates automatically (without having to manually update the page).

#### **Alarms/Events**

Shows a list of current alarms/events. For each alarm, its text, last occurrence time, occurrence count, end time, and acknowledge time are displayed. The list updates automatically (without having to manually update the page).

#### **Reports**

Shows a list of reports. Clicking a report will open it in the report viewer. The viewer allows you to set a date range for generating a report, change the appearance of the page, and export the report to various formats. The available formats are listed depending on whether or not Microsoft Office is installed on the computer (server).

#### **Custom Reports**

Shows a list of custom reports. Clicking a custom report will open it in the custom report viewer. The viewer allows exporting the report to various formats. The available formats are listed depending on whether or not Microsoft Office is installed on the computer (server).

The lists display object aliases that can be localized into an unlimited number of languages. Every tab, therefore, allows switching between the project languages. All lists of objects can be downloaded as CSV.

### 3.4 Thin Clients Web Page

The **Thin Clients** page is designed to run the thin clients. It has two tabs – **Web Client** and **Smart Client** –, each of which contains a list of configurations, basic information, and thin-client startup requirements. For more information, refer to the documentation that accompanies each thin client.

**Note:** Depending on the project settings, one of the tabs may be unavailable.

## 3.5 Administration Web Page

The **Administration** page provides server administration and basic diagnostics. The page is divided into five tabs:

### System Information

Contains information about the running data server and the loaded project.

### Connected Clients

Contains two tables. The upper table lists network connections to other runtime software modules. The lower one lists Web server sessions: thin clients, OPC servers, third-party applications (SOAP only). You can *Disconnect* each thin client. Once disconnected, the thin client's session is ended. The user (client) will then be notified of this fact. Disconnecting the client will release its license. The *Disconnect all* command is used to disconnect all clients at a time. Both the tables update automatically (without having to manually update the page).

### Server Diagnostics

Allows displaying both current and historical server-related and system-resource-related diagnostic data as a trend. You can switch between the data types using the drop-down list above the trend. Also, you can change the time range using the controls below the trend (*Minute, Hour, Day, Week, All*). If it is set to *Minute*, the trend updates automatically (without having to manually update the page). Trends with two or more series can stack the series (the *Stacked* option).

### Security

Allows managing the lists of IP addresses whose requests are accepted (whitelist) or denied (blacklist) by the server. This function is described directly on the page. The lists are stored in the project subdirectory in `Main\Config\IPBlackList.txt` and `Main\Config\IPWhiteList.txt` (e.g., `C:\MyProject\BoilerRoom>Main\Config\IPBlackList.txt`). If they cannot be found in this location, they will be loaded from the files `Config\IPBlackList.txt` and `Config\IPWhiteList.txt` located in the **Reliance** program files directory, typically `C:\Program Files (x86)\GEOVAP\Reliance4\Config\IPBlackList.txt`.

### Logs

The upper part lists the current log file's latest records. The list updates automatically (without having to manually update the page). Updates can be enabled/disabled by activating/deactivating the *Auto-scroll* option. The lower part lists all log files of the data server. Clicking a list item will download the appropriate log file.



## 3.6 API Web Page

The **API** page describes application programming interfaces (API) intended to create applications that allow connecting to a data server. It is divided into two tabs:

### SOAP

Describes an interface for connecting to the data server's Web service (SOAP, the successor of XML-RPC). From the page, you can download a WSDL (Web Services Description Language) file. Many of today's programming tools (Microsoft Visual Studio, Borland/Embarcadero Delphi, Java, etc.) allow creating a program that works as a Web service client. The WSDL file or its URL will be sufficient to establish a basic framework for such a program. An example application, which communicates with the data server via the Web service interface, is part of the **Reliance** installation. The example is provided for the following programming languages (environments): Object Pascal (Borland Delphi 7), C# (Visual Studio), and Java.

### REST

Describes a simple and powerful application programming interface of type REST for connecting to the data server. It explains the principle of data exchange and the method of request security. There are also examples of calls there.

Connecting to the data server via an API requires a password, which can be specified through **Reliance Design's** *Project Options* dialog (*Project > Options > Web > API*).

**Note:** Depending on the project settings, one of the tabs may be unavailable.

## 4 Web Server Settings

A data server's Web server and Web pages can be configured in **Reliance Design** at the following locations:

### **Project Options > Web**

Contains options and properties common to all computers defined within the project. Among other things, it allows setting:

- Gzip compression
- HTTP headers
- Default page
- Custom content embedded in the Welcome! page
- Access rights for the Web pages
- DoS attack protection
- Support for IoT
- API

### **Project Structure Manager > computer > Web**

Contains options and properties specific to a selected computer. Among other things, it allows you to:

- Activate/deactivate starting the Web server (for HTTP and/or HTTPS)
- Set port numbers (for HTTP and/or HTTPS)
- Set the SSL version (for HTTPS)

These options and properties are described in detail in the **Development Environment** manual.

## 4.1 HTTPS and Certificates

To turn on HTTPS, you must activate the *HTTPS* option in the **Reliance Design** development environment's Project Structure Manager (*computer > Web > HTTPS*). Optionally, you can specify the port number and SSL version (SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2). For data encryption, an SSL certificate is required. The well-known and proven OpenSSL library is used by the **Reliance** data servers' (**Reliance Server** or **Reliance Control Server**) built-in Web server to encrypt data. The **Reliance** system requires that the certificate be divided into the following parts:

- Certificate
- Private key
- Root certificate
- Certificate password

The certificate's primary location is the Project Structure Manager (*computer > Web*). This allows using different certificates for different computers defined within the project.

If a certificate is not specified in the Project Structure Manager, it is loaded by the **Reliance** system from the files:

- `cert. pem` (a file that consists of the certificate)
- `key. pem` (a file that consists of a private key)
- `root. pem` (a file that consists of a root certificate)
- `pass. txt` (a text file whose first line consists of the password for the certificate – the file is required only when the certificate is password protected)

The certificate files can be located in two directories:

1. In the project's `Main\Cert` subdirectory. For example: `C:\MyProject\BoilerRoom\Main\Cert`. This is the recommended location for the certificate. The certificate will only be used for this project.

2. In the `Cert` subdirectory of **Reliance's** program files, typically `C:\Program Files (x86)\GEOVAP\Reliance4\Cert`. After installing **Reliance**, the files that make up a certificate supplied together with this system (issued and signed by GEOVAP, i.e., it is the so-called self-signed certificate) are located in this folder. It is apparent from their contents what format for the certificate files is required by the Web server. So, they can serve as an example to systems integrators. The certificate located in this directory (program files) will be used for all projects running on the same computer (if the certificate is not stored in the project's `Main\Cert` subdirectory – see the previous paragraph).

There are three ways of obtaining/activating the certificate:

- Using the certificate supplied together with **Reliance**
- Generating your own certificate
- Purchasing a certificate issued by a trusted certificate authority

### The certificate supplied together with Reliance

The easiest way is to use the certificate supplied together with **Reliance** (see above). This certificate allows data encryption, which, however, requires confirming a security exception in the browser on the user's computer. Despite confirming it, an untrusted certificate will be indicated by a red padlock, exclamation mark, or crossed-out address in the browser's address bar. This certificate is suitable for testing purposes or for use in an intranet or VPN.

### Your own certificate

Secondly, you can generate your own certificate for a specific address (the computer's hostname or IP address). Its root certificate must be installed in the browser as a trusted certificate authority on all client computers. This time, a green padlock will indicate a secure connection. This certificate is also suitable for testing purposes or for use in an intranet or VPN. To generate the certificate, use the batch files located in the `BatchFiles\Cert` subdirectory of **Reliance's** program files, typically `C:\Program Files (x86)\GEOVAP\Reliance4\BatchFiles\Cert`. The instructions are as follows:

1. Download OpenSSL (e.g., <https://indy.fulgan.com/SSL/>).
2. Extract OpenSSL to `C:\OpenSSL` (or elsewhere, the path can be changed directly in the batch files).
3. Change the `BatchFiles\Cert\OpenSSL.cnf` file as follows:

Change the hostname and/or IP address. If the server will run, for example, on <https://myserver> and <https://192.168.1.100>, the file must be changed as follows:

```
[ req ]
countryName_default = US
stateOrProvinceName_default = New Jersey
localityName_default = Gotham
0.organizationName_default = Crackerjack Limited
organizationalUnitName_default =
commonName_default = myserver
emailAddress_default =

[ alt_names ]
DNS.1 = myserver
IP.1 = 192.168.1.100
```

4. Run the `BatchFiles\Cert\CreateCertAuth.bat` file to create a certificate authority file (root certificate). During the execution of the batch file, you will be required to enter and verify your password for the certificate. Also, you will have to confirm the information entered in the `OpenSSL.cnf` file. The batch job results, among other things, in the `root.pem` file being created.
5. Run the `BatchFiles\Cert\CreateServerCert.bat` file to create a key and a certificate for the server. During the execution of the batch file, you will be required to enter and verify your password for the certificate several times. It is generally a different password than the one for the root certificate. For the sake of simplicity, however, you can enter the same password. Also, you will have to confirm the information entered in the `OpenSSL.cnf` file. The batch job results, among other things, in the `cert.pem` and `key.pem` files being created.
6. Place the `root.pem`, `cert.pem`, and `key.pem` files in a directory intended for certificates (see above). In the directory, create the `pass.txt` file whose first line consists of the password used in step 5.
7. Import the root certificate file `root.pem` into the Web browser as a trusted certificate authority. The import process depends on your Web browser. The file must be imported into the browser on all the client computers on which a Web page is to be used (e.g., <https://myserver>).

### A certificate issued by a trusted certificate authority

This certificate is suitable for running a website (Web pages) on the Internet (e.g., <https://www.myreliapp.com>). It is essential that you purchase a certificate issued by a trusted certificate authority (e.g., Thawte, Symantec, GeoTrust, RapidSSL, DigiCert). This time, a green padlock will indicate a secure connection. The certificate is purchased for a particular domain and must be renewed regularly (once a year or two years).

Certificate authorities issue their certificates in various formats. There are tools for converting them to different formats. One of them is OpenSSL. The **Reliance** system's Web server requires the PEM (X509) format for the certificate. It is a text format that is Base64 encoded. Files in this format usually have the following extensions: `.pem`, `.crt`, `.cer`, `.key`, `.txt`.

#### Example 1:

Converting a certificate in the DER format (a binary format, extensions: `.der`, `.crt`, `.cer`) to the PEM format:

```
openssl x509 -inform der -in certificate.cer -out certificate.pem
```

#### Example 2:

Converting a certificate in the PKCS #12 format (`.p12`, `.pfx`), which contains a private key + an SSL certificate, to the PEM format:

```
openssl pkcs12 -in keyStore.pfx -out keyStore.pem -nodes
```

**Note:** The certificate is sometimes supplied as one file, in which all the three parts are contained – the certificate, the root certificate, and the key. In such a case, it must be divided into individual files.

## 4.2 MIME

The Internet media types provided by **Reliance**'s Web server (e.g., a PDF document) are described in the `MIME.txt` file. The file contains a list, the lines of which are made up of pairs. Each line represents a pair written as follows: `<file extension>=<media type>`. The file is located in the project's `Main\Config` subdirectory. For example: `C:\MyProject\BoilerRoom\Main\Config\MIME.txt`. If it cannot be found in this location, it will be loaded from the **Reliance** program files directory, typically `C:\Program Files (x86)\GEOVAP\Reliance4\Config\MIME.txt`. Based on the media type, the Web browser determines how to work with a particular document.

### 4.3 User-Agent

The User-Agent HTTP header is used by **Reliance**'s Web server, for example, to detect the type of Web browser. The `MobileUserAgents.txt` file contains a list of strings typical of Web browsers in mobile devices (smartphones, tablets). If any of the strings is part of the User-Agent header, the browser is detected as mobile, and the content provided by the Web server can be adapted to this fact. The file is located in the project's `Main\Config` subdirectory. For example: `C:\MyProject\BoilerRoom>Main\Config\MobileUserAgents.txt`. If it cannot be found in this location, it will be loaded from the **Reliance** program files directory, typically `C:\Program Files (x86)\GEOVAP\Reliance4\Config\MobileUserAgents.txt`.

## 4.4 Root Directory

The project's `ThinClients` folder is the Web server's root directory. It has a number of subdirectories and files, some of which are automatically generated when exporting the project for remote users in **Reliance Design** or at project startup (or Web server startup).

The meaning of some files and subdirectories:

### The `.well-known\security.txt` file

Contains information on where and how to report security issues. It is a standard described, for example, at <https://securitytxt.org>.

### The `robots.txt` file

Contains directives for search engines that may prevent specific parts of the Web from being searched. It is a standard described, for example, at <https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt>.

### The `Web` subdirectory

Contains the data server's Web page files.

### The `WebClient` subdirectory

Contains **Reliance Web Client**'s files.

### The `SmartClient` subdirectory

Contains **Reliance Smart Client**'s files.

### The `Private` subdirectory

This subdirectory's files are not provided by the Web server.

### The `WSDL` subdirectory

Contains a file of type WSDL that describes the custom application interface of the Web service (API).

The `ThinClients` folder is intended for custom files and directories. They will be provided by the Web server at an address corresponding to the directory structure. If, for example, the `ThinClients` folder contains the `Custom` subfolder, which contains the `Hello.html` file, this document will be available at `http(s)://<computer_name>:<tcp_port_number>/Custom/Hello.html`.

## 5 Other

### IoT

If the *Enable receiving and handling messages* option in **Reliance Design's Project Options** dialog is active (*Project > Options > Web > IoT*), the Web server will pass received IoT requests to a communication driver for further processing. To find out the address to which the messages should be sent, refer to the **Administration** page's **System Information** tab. See the values *IoT (LoRa)*, *IoT (Sigfox)*, and *IoT Basic Access Authentication*.

### Changing the page appearance

Generating the data server's Web pages is designed so that their appearance can be easily customized. The reason can be, for example, a request from an end customer for placing their logo on the Web pages. The files that make up the content of the Web pages are located in the compressed folder (archive file) *Pages.zip*, typically *C:\Program Files (x86)\GEOVAP\Reliance4\WebServer\Pages.zip*. At data server startup, this folder's content is copied to the Web server's directory. The content of this folder should not be changed so as to change the Web pages. This is due to the fact that when **Reliance** is updated to a newer version, this folder is also updated and the changes made are lost. To customize the data server's Web pages, the *PagesCustom.zi\_* file must be used. If necessary, simply rename its extension *.zi\_* to *.zip* and copy the files you have changed (and/or other files) to this compressed folder. All files contained in the compressed folder (archive file) *PagesCustom.zip* are copied to the Web server's directory only after the content of the compressed folder *Pages.zip* is copied to this directory.

### Aliases

For quick access to selected pages, the following aliases are predefined:

`http(s)://<computer_name>:<tcp_port_number>/m`                      Redirects to  
**Reliance Smart Client's** Web pages

`http(s)://<computer_name>:<tcp_port_number>/w`                      Redirects to  
**Reliance Web Client's** Web pages

## 6 Glossary

### **API**

Stands for *Application Programming Interface*.

### **gzip** (*an acronym for GNU zip*)

A software algorithm used for data compression.

### **HTTP** (*Hypertext Transfer Protocol*)

A frequently used protocol for data communication between Web servers and clients.

### **HTTPS** (*Hypertext Transfer Protocol Secure*)

A protocol used for secure communication with a Web server. HTTPS is an extension of HTTP and its security is based on SSL or TLS.

### **OPC** (*OLE for Process Control*)

A standard for data exchange in industrial automation created by the OPC Foundation.

### **OpenSSL**

A software library implementing the SSL and TLS protocols.

### **REST** (*Representational State Transfer*)

A software architectural style that can be used to easily create, read, update, and delete information from the server using simple HTTP calls.

### **Windows service**

A computer program running in Windows. It has no user interface. The service can operate when a user is not logged on.

### **SSL** (*Secure Sockets Layer*)

Refers to a protocol or layer between the transport layer (e.g., TCP/IP) and the application layer (e.g., HTTP) that uses encryption for secure communication.

### **TLS** (*Transport Layer Security*)

The successor to SSL.

### **URL** (*Uniform Resource Locator*)

It is a string with a defined structure, which specifies the location of information sources (documents, services) on the Internet.

**WSDL**

Stands for *Web Services Description Language*.

**XML** (*Extensible Markup Language*)

A general markup language developed to easily create specific markup languages for different purposes and for a wide range of different data types.

**Data servers**

A common term for the **Reliance Server** and **Reliance Control Server** runtime software.

**HTML** (*Hypertext Markup Language*)

A markup language for creating Web pages.

**Java Web Start**

A framework developed by Sun Microsystems that allows starting and automatically updating programs written in Java directly from Web pages.

**JRE** (*Java Runtime Environment*)

The runtime environment for programs written in Java.

**SOAP** (*Simple Object Access Protocol*)

A protocol for exchanging XML-based messages over the Internet, usually via HTTP.

**TCP port**

A special number used to map data to a particular process running on a computer.

**Thin clients**

A common term for **Reliance Web Client** and **Reliance Smart Client** (designed for use with smartphones and tablets).

**Web server**

A program responsible for processing HTTP requests from clients, usually from Web browsers. Processing a request means, for example, delivering a Web page.

**Web service**

A part of a program that allows data exchange with client applications over the Internet by means of the SOAP protocol. To transfer the data, the Web service uses a Web server.