

WINGS WEB SERVICE MODULE

GENERAL

The Wings Web Service Module is a SOAP (Simple Object Access Protocol) interface that sits as an extra layer on top of the Wings Accounting Interface file import (WAlimp) and Wings Commercial Interface file import (WClimp). The SOAP protocol is based on simple standards like XML and HTML.

The Wings Web Service allows external programs – whether they are in the cloud or not – to interface with your on-premises Wings installation. The most common application is probably that of web shops integrating with Wings.

The Wings Web Service Module is a Windows service application that should be installed on the server where Wings is installed. If you have both an application server where the Wings programs are installed and a database server where the Wings database is running, you will have to install the Wings Web Service Module on your application server.

Make sure that either WAI.dll or WCI.dll, or both, are properly registered as COM objects on your server. You can do this by executing WingsAcc.exe or WingsCom.exe (with elevated rights if UAC is active) and selecting Help | Support | Register WAI/WCI. Alternatively you can use this command via the command prompt to register the COM objects (but again, with elevated rights when necessary):

```
RegSvr32 C:\Wings\WAI.dll
```

and/or

```
RegSvr32 C:\Wings\WCI.dll  
("C:\Wings" being the path where Wings is installed)
```

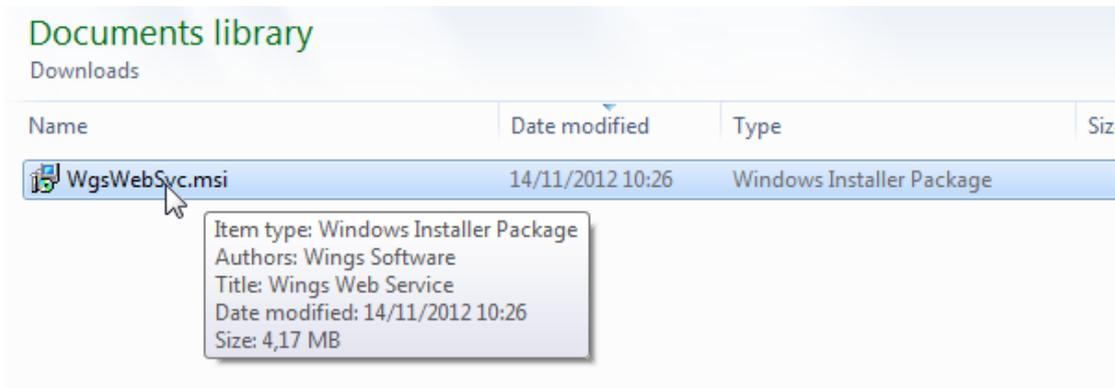
The Wings Web Service allows for a bidirectional link, where XML files with update information can be sent to Wings and information from SQL queries can be retrieved via XML files.

Schematic:

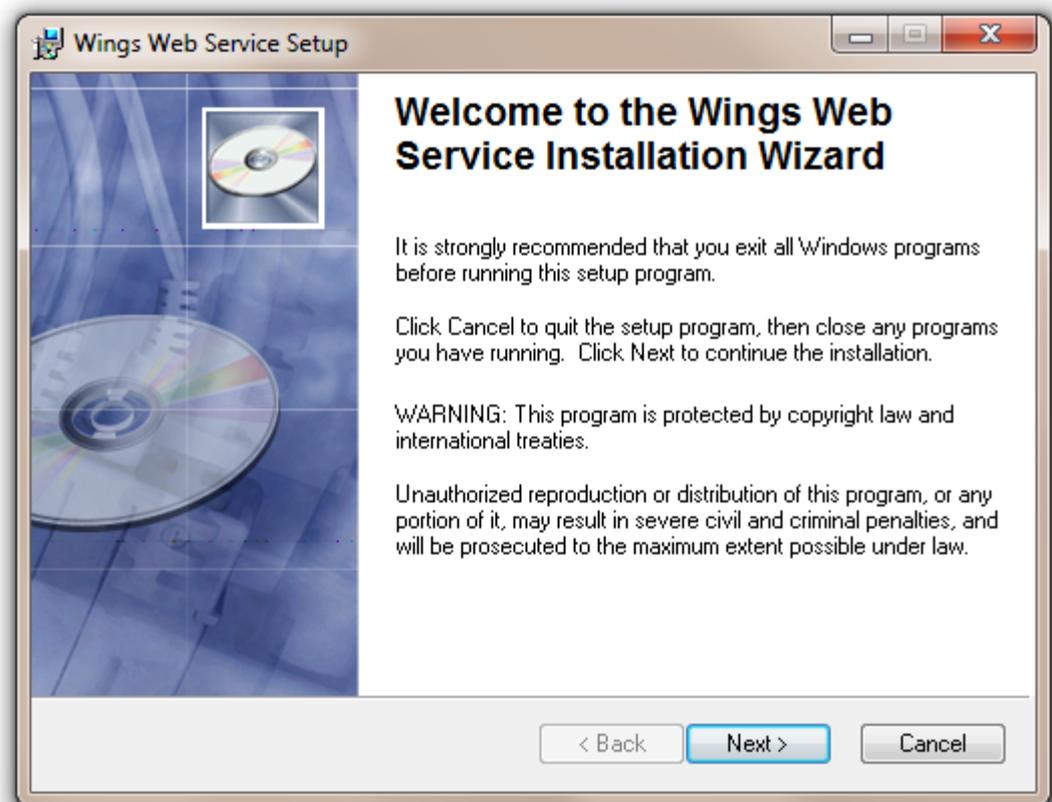


INSTALLATION

1. Log into your server as Administrator and double click on the msi file to launch the installation program:

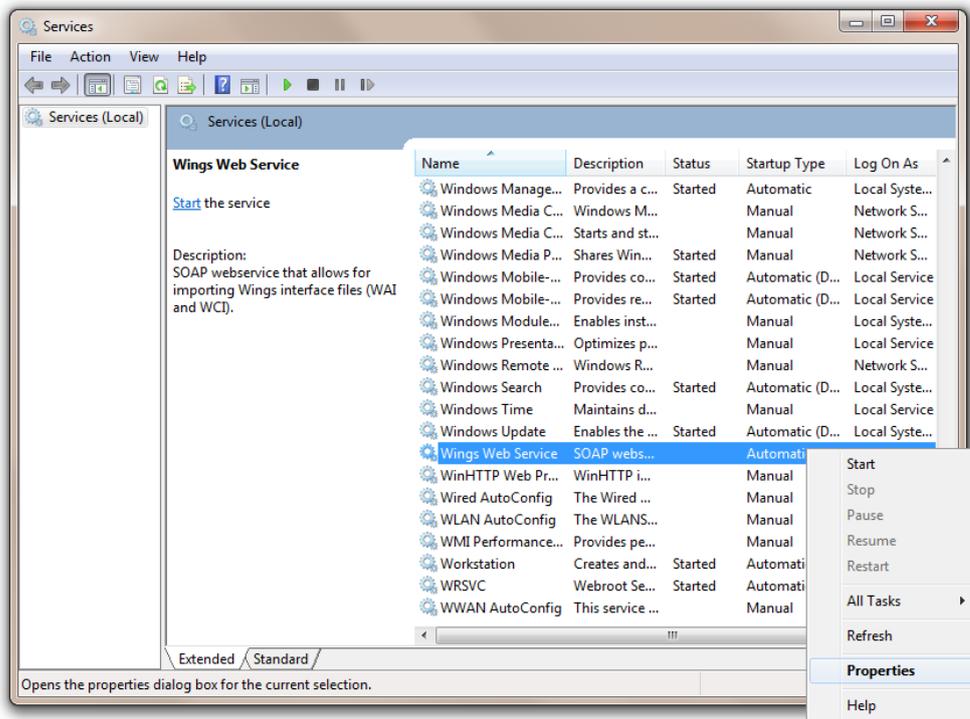


2. Follow all steps of the installation process:

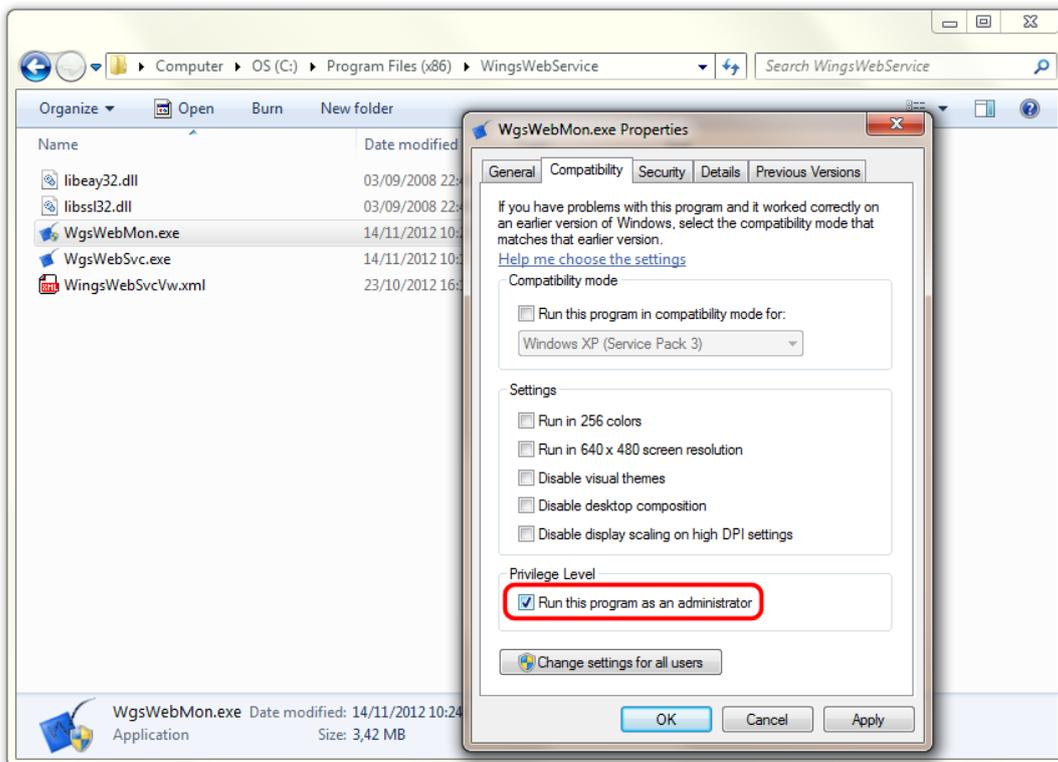


3. Go to Services via the Control Panel and check whether the service is correctly installed (see next screen example)

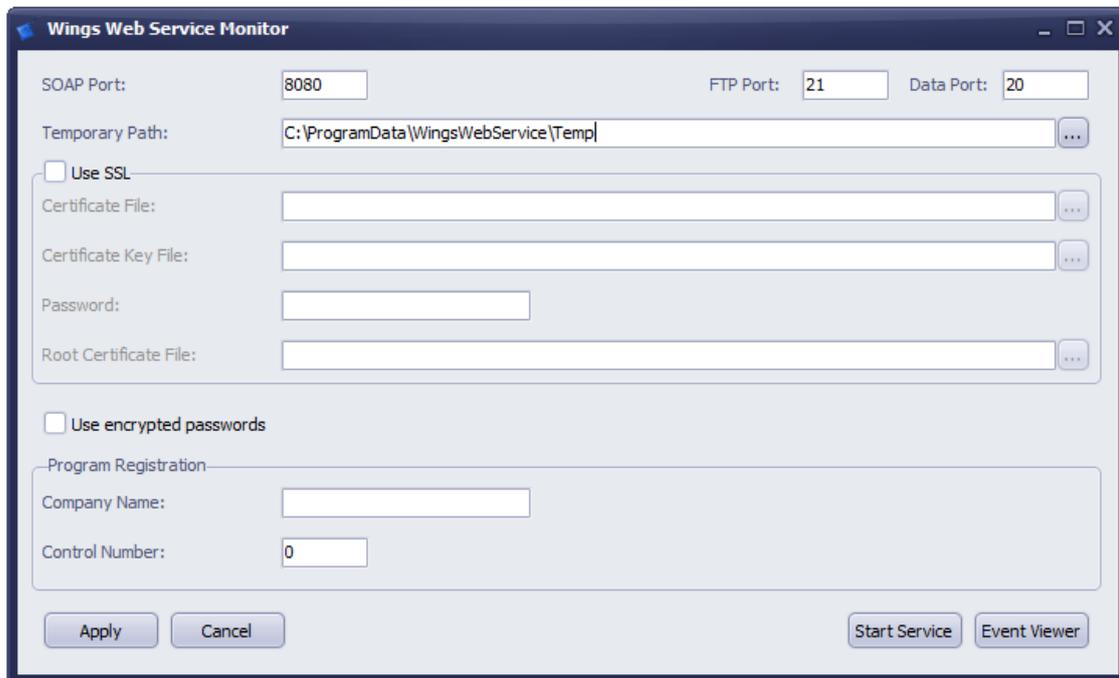
4. If the service has to access non-local resources, you might need to change the “Log On” properties:



5. Find the Wings Web Service Monitor program and check that “Run this program as an administrator” is checked via its properties (when applicable to your OS version):



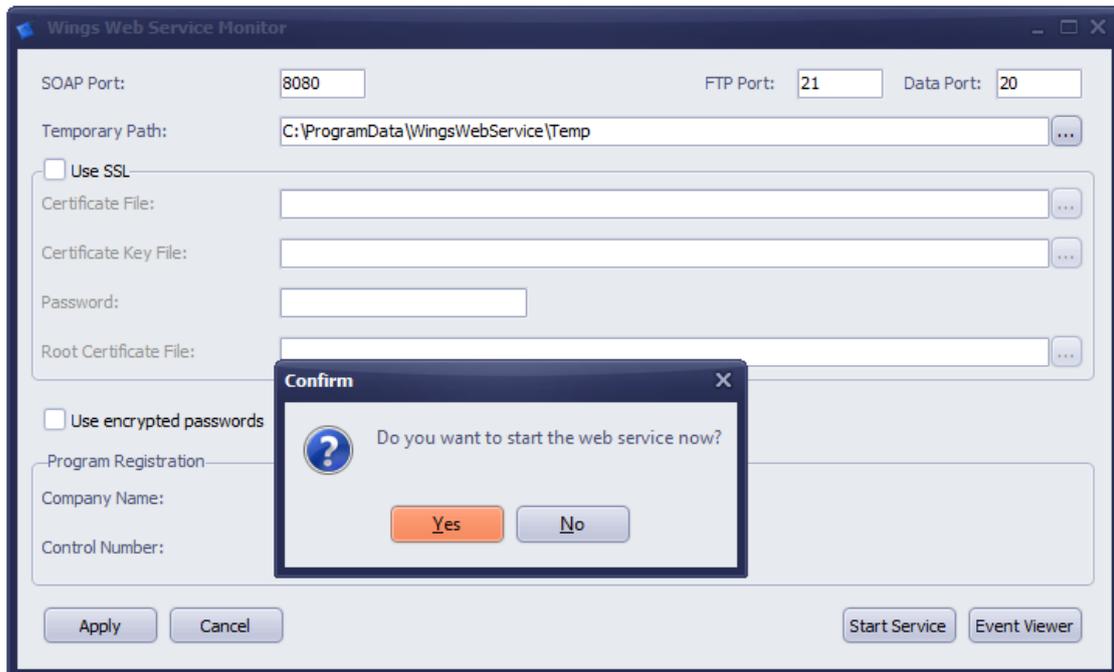
6. Launch the Wings Web Service Monitor (a shortcut has been created in your Programs list) and fill out the various parameters:



The screenshot shows the 'Wings Web Service Monitor' configuration window. It contains several input fields and checkboxes. The 'SOAP Port' is set to 8080, 'FTP Port' to 21, and 'Data Port' to 20. The 'Temporary Path' is set to 'C:\ProgramData\WingsWebService\Temp'. There are checkboxes for 'Use SSL' and 'Use encrypted passwords', both of which are currently unchecked. Below these are fields for 'Certificate File', 'Certificate Key File', 'Password', and 'Root Certificate File'. At the bottom, there is a 'Program Registration' section with fields for 'Company Name' and 'Control Number' (set to 0). Buttons for 'Apply', 'Cancel', 'Start Service', and 'Event Viewer' are located at the bottom of the window.

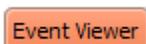
- **SOAP PORT:**
You can choose the SOAP port number that the web service will use (default 8080).
Check that this port is open on your firewall.
- **FTP PORT/DATA PORT:**
You can choose the port numbers for FTP access to the web service (default 21 and 20).
Check that these ports are open on your firewall.
- **TEMPORARY PATH:**
It is advisable to create a separate directory for storing temporary import files.
You can use these files to track down what exactly was imported or why an import failed.
The temporary files will automatically be deleted after a week.
- **USE SSL:**
To increase security, you can add SSL (Secure Socket Layer) to the web service. All you have to do is to select a valid certificate and key file on your server (and set the password when needed).
- **USE ENCRYPTED PASSWORDS:**
When you use this feature, you will have to request an encrypted version of your user password from Wings. This feature is primarily added for our own Wings Online Web Service.
- **PROGRAM REGISTRATION:**
In order to use the Wings Web Service, you will need to provide the Company Name and Control Number.
Do not start the web service until you have correctly registered the program.

7. After you click on “Apply”, you will be asked to start the web service:



Select “Yes”.

8. You can either check that the Wings Web Service is running via the Services of the Control Panel or via the Windows Event Viewer.
All calls and events of the Wings Web Service will be logged in the Windows Application Event Log.
A filtered list of Wings Web Service events is available via this button of the Wings Web Service Monitor:



9. Now check that the web service is accessible from outside the network. The first step is to access the public page via the IP address (or domain name if available) and internet port, e.g.:
<http://248.185.21.95:8080>

10. You should then see something like this:

WgsWebSvc - Service Info Page

WgsWebSvc - PortTypes:

- **IWingsWeb [WSDL]**
 - Login
 - Logout
 - KeepAlive
 - SendWAIFile
 - SendWCIFile
 - GetErrorAsERR
 - GetErrorAsXML
 - GetRestFile
 - GetSQLResult
 - GetRestFileAsZIP
 - GetSQLResultAsZIP
 - FtpSendWAIFile
 - FtpSendWCIFile
 - FtpGetErrorAsERR
 - FtpGetErrorAsXML
 - FtpGetRestFileAsZIP
 - FtpGetSQLResultAsZIP

- **IWSDLPublish [WSDL]**
 - Lists all the PortTypes published by this Service
 - GetPortTypeList
 - GetWSDLForPortType
 - GetTypeSystemsList
 - GetXSDForTypeSystem

WSIL: Link to WS-Inspection document of Services [here](#)

- When you click on the “IWingsWeb” link, you see a list of the complete function descriptions
- The WSDL link is what should be imported in your programming language in order to access the web service

11. Create a Windows user to access the Wings resources via this web service. The user should have similar rights as an ordinary Wings user, but it is not necessary to add the user to the Wings application.

12. You are now ready to use the Wings Web Service. See the general Wings Web Service documentation for more information on how to use this.

A PRACTICAL EXAMPLE

A typical scenario for using the Wings Web Service Module is to allow an external web shop to interface with Wings. In this section, we will briefly discuss the questions that come up in this particular scenario. However, the points of attention remain largely the same in other interface scenarios.

For a starter, the web shop should have access to customers and products. This can be done by sending a WCIimp file containing an SQL query to the Wings Web Service (SendWCIFile). The XML result file(s) can then be retrieved (GetSQLResult). See the Wings Web Service documentation for schema's of this interaction.

But as always with interfaces, you will have to carefully plan how inserts and updates will occur in the two applications. Should products always be created and updated in Wings and synchronized once a day to the web shop, or is the product table of the web shop the master table? The same question can be asked for the customer table. When a new customer is allowed to register himself on the web shop, do you want to automatically create this new customer in Wings, or do you want to place the customer/order on hold for manual checking by an operator before sending the data to Wings?

When you are sure that the customer and the products he ordered are also present in Wings, you can create an XML order file and send it via the Wings Web Service (SendWCIFile). But when necessary, you can also create/update customer and product information in the same file. See the specific WCI documentation on how to correctly create a customer order file, with or without additional customer and product data.

Another area of attention is inventory information. Will you retrieve the product inventory information once a day or rather have the inventory data fetched from Wings as your customer inputs his order? For performance reasons, a good compromise could consist in the combination of both: you could update inventory information once a day, but retrieve actual inventory information for the products being ordered in the background, while the customer is entering the order.

Jointly with the other WAI and WCI documentation, you will find some examples of queries on the Wings database. In order to be able to easily create SQL queries, you can also obtain a description of the Wings data tables from support@wings.be.

INTERFACE MONITORING

In an ideal world, things never go wrong. Unfortunately, – as experience learns – in real live situations you should be prepared to encounter error conditions and handle them appropriately.

Of course, you can handle some of the errors (returned by `GetErrorAsXML`) in your program, but it would be an illusion to think that you could automate the response to all possible error conditions. Instead, you could at least make sure that you are aware of the error condition, for example by automatically letting your software send you an e-mail when an unexpected error is encountered. Consequently, you can do the following to examine the problem:

1. Check that the Wings Web Service is running;
2. Launch the Wings Web Service Monitor and click on Event Viewer to see list of all handled calls and their result (or start the Windows Event Viewer and filter the events yourself);
3. Go to the temporary directory where the Wings Web Service will store all `WAlimp/WClimp` files and get a closer look at the `.err` file to examine the nature of the problem;
4. If you can fix the import file yourself, you can retry the import by manually launching `WAlimp.exe/WClimp.exe` and selecting the file. Another option would be to retry the operation from your own software, after fixing the error condition.