

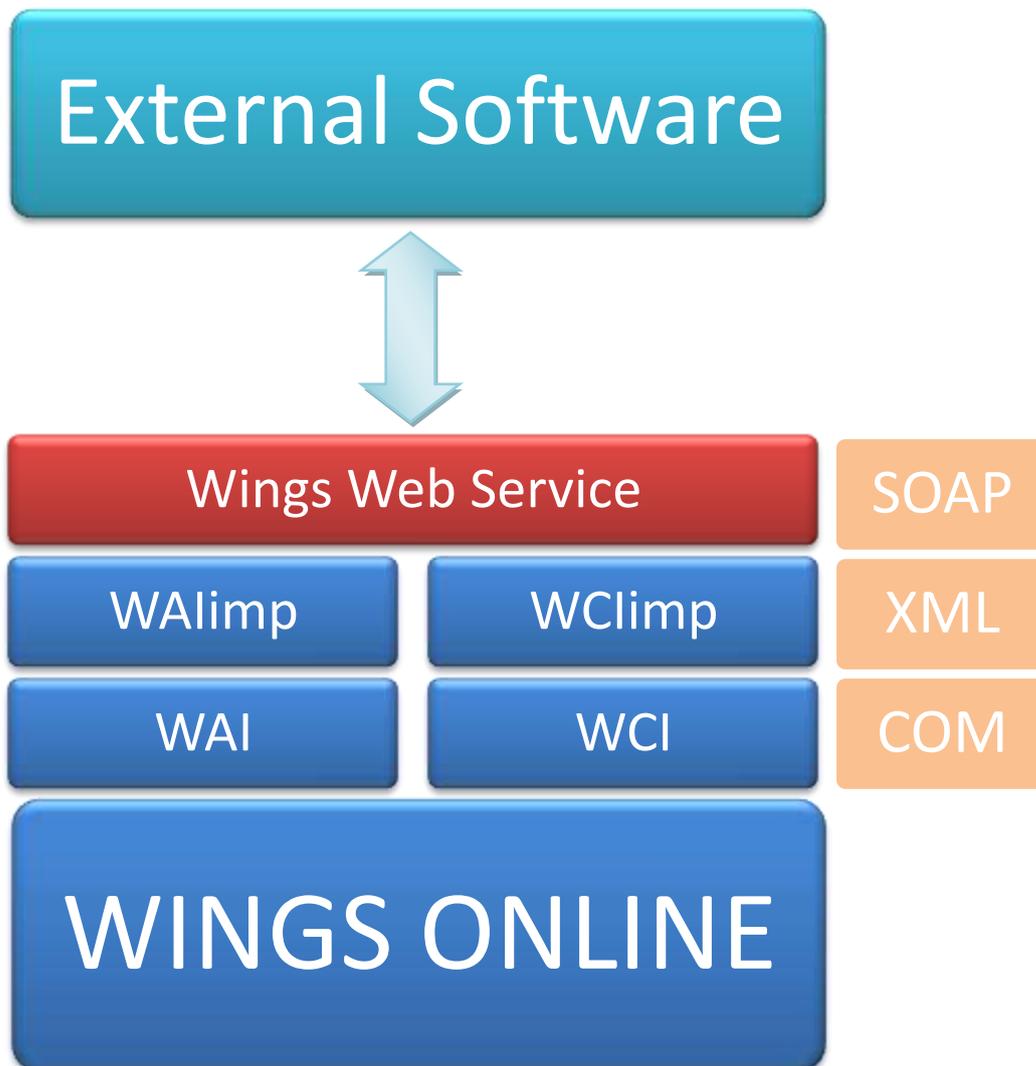
WINGS WEB SERVICE INTERFACE

GENERAL

The Wings Web Service 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 Wings Online.

Schematic:



The interface to the web service can be imported via this link:

<https://online.wingscloud.eu:8080/wsd/IWingsWeb>

Every modern programming language allows to import this wsdl in order to automatically provide the interface functions and make it easy to use the web service.

USAGE

How exactly do you use the Wings Web Service? In short: the Wings Web Service is a web layer to the WALimp (for Wings Accounting) or WClimp (for Wings Logistics) application. When you are able to create a WALimp or WClimp file (see their respective documentations on how to achieve this), this web service will allow you to send this file for processing.

You will immediately get the processing result (true or false) and will be able to get detailed information on the error in case of failure.

Exactly as with the direct WALimp and WClimp import, you are able to pass an SQL query via the interface file and get the result in an XML file. For the purpose of reading the XML result file, you have a separate function.

When you apply for usage of the Wings Web Service, you will receive three things from Wings:

1. A user name: e.g. 'wingscloud\wings1234'
2. A password: e.g. 'Jk9ks5HX'
3. A home directory: e.g. '\\wings-db-01\wingsdata\My Wings'

You have to use this directory as RootPath in your import file and it is advisable to use this path to store your SQL query results (SQL.ResultFile property).

SECURITY

In a cloud application as Wings Online, security is critical. You certainly don't want others to have access to your personal data or company data.

Especially since many different customers share the same environment and because a SOAP web service is publicly accessible, by definition, security is essential for this web service.

The Wings Web Service has several advanced security features to avoid unauthorized access:

1. Any method you call must use a token that you obtain from the Login method. This Login will only return a usable token when a valid username/password combination is sent to the Wings Web Service. This username/password combination is checked on the operating system level. The security settings of the username are applied for the context of the Wings Web Service and will disallow any unauthorized access to other user's data.
2. To avoid users having their regular Wings Online username/password combination used to access the Wings Web Service (or vice versa), the password that is used on the Operation System is encrypted with a lightweight encryption. Hence, the password that you receive from Wings for the Wings Web Service is an encrypted version of the password on the operating system level of the Wings Online Platform and will be decrypted by the Wings Web Service for authentication. You don't need to worry about this, since this whole process is transparent for the user of this web service.
3. It is never a good idea to send plain text username/password combinations over the internet because packet sniffer software could possibly be monitoring the internet traffic between sender and receiver. In order to make the username and password illegible by sniffers, everything is encrypted by an extra SSL layer (Secure Socket Layer).
4. Every call is logged in the operating system's Event Log. For Login/Logout this is done both in the Wings Web Service specific event log as on the operating system's authentication log.

VIRTUAL FTP SERVER

For most purposes, the transfer of data through base64 encoded strings works just fine. However, when encoding or decoding to and from base64 is problematic, or when performance is crucial an alternative is provided through FTP.

All methods that use FTP have identical names to the base64 string based methods, but start with 'Ftp'. In order to use these methods, file transfer to and from the web service will be done through a virtual FTP server.

You can access the virtual FTP server via these parameters:

1. URL: <ftp://online.wingscloud.eu>
2. User name: the same user name that you used for Login() of the web service, but without the domain name part. E.g. 'wings1234'
3. Password: the token handle returned by Login(), converted to a string. E.g. when you receive the numeric value 5544523 as the token handle, the password for the virtual FTP server will be '5544523' (decimal value without leading zeros)
4. It is advisable to use active FTP at the client side

The virtual FTP server will act as a regular FTP server with following restrictions:

- you can only access the FTP server while the Login() access token is active, i.e. after a Login() and before an explicit Logout() or an automatic logout after one hour of inactivity
- you cannot create subdirectories on the FTP server, all files will be uploaded/downloaded from the root drive of the virtual FTP server
- all files on the virtual FTP server are automatically destroyed after logout

The FileName parameter of FtpSendWAFile() and FtpSendWCIFile() may not contain any path information (e.g. 'MyFile.xml') and the file names returned by the other FTP related web service methods will not return any path information either.

IWINGSWEB

METHODS:

	<i>Method Name</i>	<i>Parameters</i>	<i>Result Type</i>	<i>Description</i>	<i>Special considerations</i>
1	Login	Username: String Password: String	Handle: Integer	Checks the user login name/password combination and returns a valid token handle when login is successful. The result value is -1 when Username/Password is not a valid combination for the Wings Web Service.	For Wings Online, the Username must contain the domain name, e.g: "wingscloud\wings1234". In order to preserve resources, the user is automatically logged out after an hour of inactivity, thus invalidating the token handle.
2	Logout	Handle: Integer	Boolean	Logs the user out of the Wings Web Service. Result: True (-1): Success False (0): Error	False means an error condition. Generally, this means that the token handle is not valid (anymore).
3	KeepAlive	Handle: Integer	Boolean	Keeps the web service connection alive for another hour. Normally the web service will automatically be disconnected after one hour of inactivity. Result is True in case of success.	

<i>Method Name</i>	<i>Parameters</i>	<i>Result Type</i>	<i>Description</i>	<i>Special considerations</i>
4 SendWAIFile	Handle: Integer EncodedContents: String	Boolean	Sends a WAImp file to the Wings Web Service for processing. The contents of the file (EncodedContents) must either be TXT, XML or ZIP encoded as a base64 string. See the WAImp documentation for more information on how to create such a file. Result: True (-1): Success False (0): Error	False means an error condition. Call GetErrorAsERR or GetErrorAsXML for a description of the error. See the WAI documentation for a list of possible error codes.
5 SendWCIFile	Handle: Integer EncodedContents: String	Boolean	Sends a WCImp file to the Wings Web Service for processing. The contents of the file (EncodedContents) must either be TXT, XML or ZIP encoded as a base64 string. See the WCImp documentation for more information on how to create such a file. Result: True (-1): Success False (0): Error	False means an error condition. Call GetErrorAsERR or GetErrorAsXML for a description of the error. See the WCI documentation for a list of possible error codes.
6 GetErrorAsERR	Handle: Integer	String	Returns the error as plain text, encoded as a base64 string. When no error is returned, the string will be empty.	Example: <i>ErrorLine = 4</i> <i>ErrorCode = 1</i> <i>ErrorString = Not found</i> <i>ErrorProperty = PathName</i>

<i>Method Name</i>	<i>Parameters</i>	<i>Result Type</i>	<i>Description</i>	<i>Special considerations</i>
7 GetErrorAsXML	Handle: Integer	String	Returns the error as XML text, encoded as a base64 string. When no error is returned, the string will be empty.	Example: <pre><?xml version="1.0" encoding="UTF-8" ?> <ErrorList> <Error> <Position type="Line">4</Position> <Code>1</Code> <Description>Not found</Description> <Property>PathName</Property> </Error> </ErrorList></pre>
8 GetRestFile	Handle: Integer	String	<p>When you look at the WALimp and WClimp documentation, you will see that unprocessed parts (because of an error) remain in the original file in order to be able to retry the import. This call returns the contents of the unprocessed part of the original file in case of an error, encoded as a base64 string.</p> <p>If the import succeeded without error, the resulting string will be empty.</p>	The rest file will be in the original format: ASCII or XML.

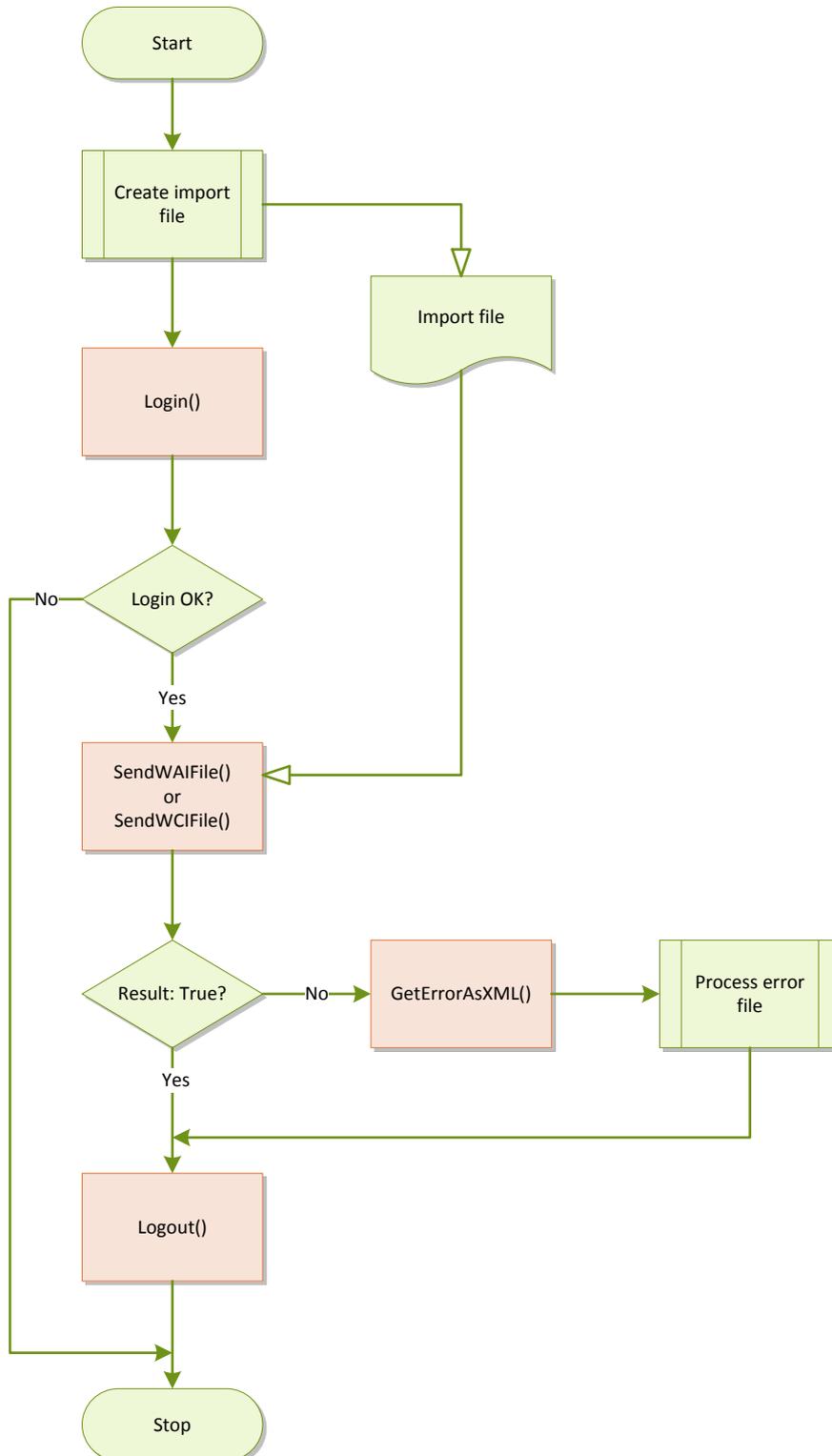
<i>Method Name</i>	<i>Parameters</i>	<i>Result Type</i>	<i>Description</i>	<i>Special considerations</i>
9 GetSQLResult	Handle: Integer FileName: String	String	Returns the SQL result as XML text, encoded as a base64 string. In case of an error, the resulting string will be empty.	The FileName must contain the path to the file on the Wings Online platform. This has to be the same file name that you provided in the SQL.ResultFile property. It is advisable to use the home directory path that you received from Wings to temporarily store your SQL query results on the Wings Online platform.
10 GetRestFileAsZIP	Handle: Integer	String	Returns the same as GetRestFile(), but with the resulting file compressed in a ZIP file.	
11 GetSQLResultAsZIP	Handle: Integer FileName: String	String	Returns the same as GetSQLResult(), but with the resulting file compressed in a ZIP file.	
12 FtpSendWAIFile	Handle: Integer FileName: String	Boolean	Sends a WAImp file to the Wings Web Service for processing. This file (FileName) must either be TXT, XML or ZIP and previously uploaded via the integrated virtual FTP server. Result: True (-1): Success False (0): Error	False means an error condition. Call FtpGetErrorAsERR or FtpGetErrorAsXML for a description of the error. See the WAI documentation for a list of possible error codes.

<i>Method Name</i>	<i>Parameters</i>	<i>Result Type</i>	<i>Description</i>	<i>Special considerations</i>
13	FtpSendWCIFile Handle: Integer FileName: String	Boolean	Sends a WCImp file to the Wings Web Service for processing. This file (FileName) must either be TXT, XML or ZIP and previously uploaded via the integrated virtual FTP server. Result: True (-1): Success False (0): Error	False means an error condition. Call FtpGetErrorAsERR or FtpGetErrorAsXML for a description of the error. See the WCI documentation for a list of possible error codes.
14	FtpGetErrorAsERR Handle: Integer	String	Returns the file name of the error as a plain text file. You need to download this file from the virtual FTP server.	Example: <i>ErrorLine = 4</i> <i>ErrorCode = 1</i> <i>ErrorString = Not found</i> <i>ErrorProperty = PathName</i>
15	FtpGetErrorAsXML Handle: Integer	String	Returns the file name of the error as an XML file. You need to download this file from the virtual FTP server.	Example: <pre><?xml version="1.0" encoding="UTF-8" ?> <ErrorList> <Error> <Position type="Line">4</Position> <Code>1</Code> <Description>Not found</Description> <Property>PathName</Property> </Error> </ErrorList></pre>
16	FtpGetRestFileAsZIP Handle: Integer	String	Returns the file name of a compressed GetRestFile(). You need to download this file from the virtual FTP server.	

<i>Method Name</i>	<i>Parameters</i>	<i>Result Type</i>	<i>Description</i>	<i>Special considerations</i>
17 FtpGetSQLResultAsZIP	Handle: Integer FileName: String	String	Returns the file name of a compressed GetSQLResult(). You need to download this file from the virtual FTP server.	

SCHEMATIC OVERVIEW FOR SENDING IMPORT FILES

EXAMPLE OF SENDING AN IMPORT FILE WITHOUT SQL STATEMENT



EXAMPLE OF SENDING AN SQL QUERY STATEMENT FILE

