TWS Toolkit

Developer's Manual

v4.3

Algoria

Web Services

Contact and Support

ALGORIA

Address:	31, rue Ernest Renan 92130 Issy-les-Moulineaux - France
Tel.:	France: +33 1 41 90 66 66 Fax: +33 1 41 90 66 56
Email:	support@algoria.fr

TWS Toolkit

Table of contents

Introduction

This document gives details of the functions available on TWS Server via the TWS Toolkit.

TWS Toolkit is a tool intended for applications developers who want to integrate CTI functions (Computer Telephony Integration) functions into their applications.

TWS Toolkit is suitable for standard development tools such as Visual C, C++ Builder, as well as Web development tools (ASP, ASP.net, PHP, etc.).

The principle of TWS Toolkit is to present as Web services a set of functions dedicated to the development of CTI applications, by masking the complexity of PBX management (whether a traditional or IP PBX).

In addition to Web services which correspond to PBX telephony functions, TWS Toolkit presents an over-set of evolved functions which facilitate the developer's tasks even more.

1. Concepts and general information

TWS Toolkit uses the TWS Server v4 so as to be able to manage a PBX. TWS Server v4 runs Windows 2008 Server and Windows 2012 Server.

Like all the products in the TWS ranges, TWS Toolkit requires an unlocking key code to be used. This key code is supplied with TWS Toolkit and is dependent on the key code of the TWS Server that TWS Toolkit is using.

TWS Toolkit provides a set of Web services for integrating telephone management processes within server and client applications.

To be able to run a CTI application developed with TWS Toolkit, or an application in which telephone functions have been integrated using TWS Toolkit, you must:

1. Access a TWS Server.
2. Unlock execution of TWS Toolkit Web services on this server by entering the corresponding key code supplied by your distributor (if TWS Toolkit is not unlocked on TWS Server, it can be accessed by other TWS applications, but not by third-party applications).

Web Services

Web services have a technology permitting applications to communicate remotely using Internet protocols (as part of an intranet, extranet, or via the Web), independently of the platforms and languages that they use. Web services use a set of standard protocols that describe the ways of calling application components.

TWS Server Components

TWS Server consists of a set of programs that make the link between both client application requests and messages generated by the PBX.

TWS Server masks the complexity of communication with the PBX, with Web services supplying a high level standard interface, whatever telephone system (traditional PBX, IP PBX, call centre) you are connected to.

This enables programmers to concentrate on developing their applications without having to worry about communication between the application and the PBX.

TWS Toolkit – Web Services

TWS Toolkit is a set of Web services designed for developers. The Web services provide a simple interface for developers to develop CTI applications very rapidly.

With TWS Toolkit it is possible to develop:

- Call-back applications
- Outgoing call wizards (automatic dialling, "click to call", etc.)
- Incoming call wizards (record display, automatic routing, intelligent routing/filtering, etc.)
- Telephone set supervision applications (line supervision, advanced functions, set programming, attendant console, etc.).

TWS Toolkit supplies:

- a Web services library
- a Web services test application
- examples of codes

TWS Toolkit provides two types of service for an application:

1. Control functions: call emission, call transfer…
2. Monitoring functions: telephone event reception (incoming calls, transferred calls, etc.).

There are three programming modes with TWS Toolkit:

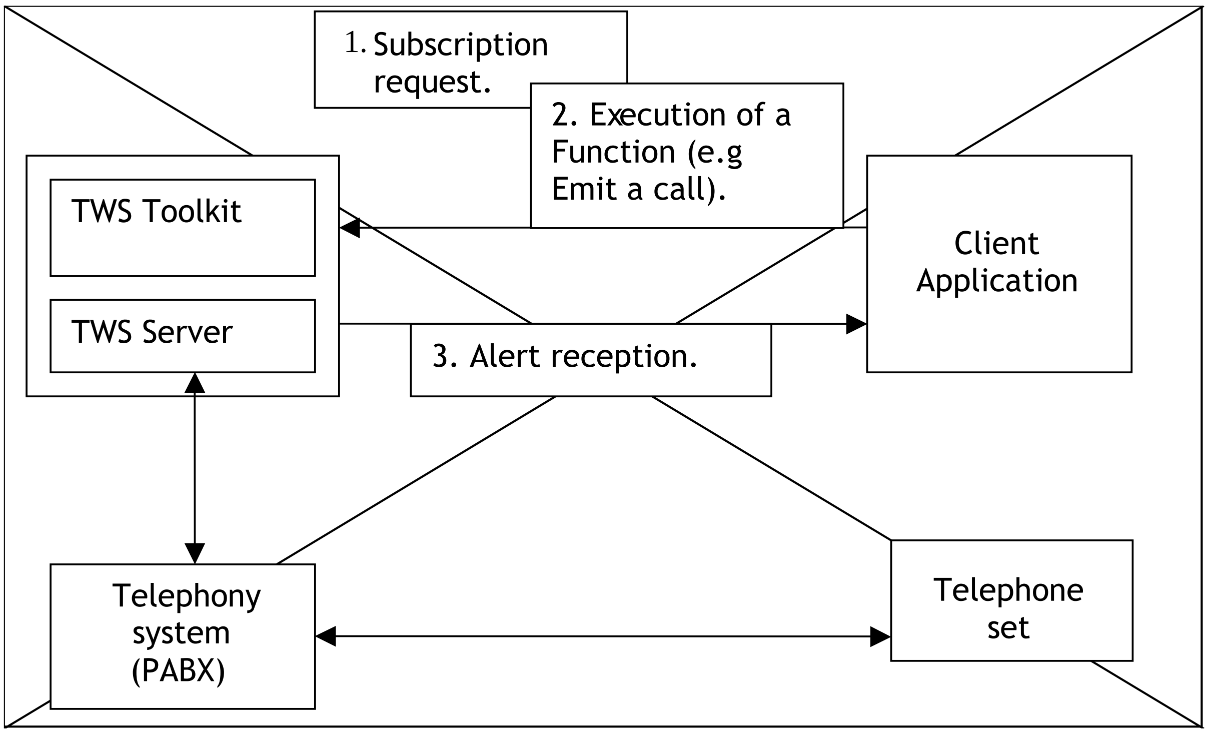
1. The application only needs to emit orders to the PBX.
2. The application only needs to receive events from the PBX.
3. The application uses the two modes described above successively or simultaneously.

Whichever mode is selected, the programming principle remains the same:

1. The application subscribes to the TWS services.
2. The application may then send orders to the telephone system.
3. If the application wants to receive the events, it listens to the address specified during its subscription.

Alerts are via TCP messages.

Architecture



2. Web service call method

As TWS Toolkit functions are a set of Web services, the client application must be able to call this type of interface.

TWS Toolkit is extremely flexible and offers different methods of accessing Web services.

SOAP native access

Calling Web services via its native interface is the best way of using Web services. In this way calling a Web service is the same as calling a standard function. Web services are then seen as local functions on the PC: there is no special processing to carry out.

Most current development tools can call Web services natively:

- all Microsoft tools including Office applications
- all Borland tools
- most Java and PHP tools, etc.

Here your work as programmer involves:

1. referencing the Web service
2. calling the functions you require

After you have referenced a Web service it is usually seen in the development tool as an object in its own right and is used like any other object in the system.

Generic example:

```
Tlk =  new TWS_Toolkit  // creation of TWS Internet object

Tlk.Fonction (parameters. etc.)     // call a function
```

A WSDL of a web service accessible at:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc?wsdl

A SingleWSDL of a web service accessible at:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc?singlewsdl

HTTP access

In certain circumstances you may not be able to use the native interface directly. To get round this problem you can call a Web service via a standard HTTP call. In this case the Web service is executed by calling an URL. All applications capable of sending an HTTP request or setting up a TCP connection can access TWS Server functions.

TWS Toolkit supports the HTTP GET and HTTP POST protocols.

There is 2 data structures for the HTTP urls: JSON and XML.

Generic example in HTTP GET in JSON:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_Function?parameter1=xxx&....

Generic example in HTTP GET in XML:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Xml/TWS_Function?parameter1=xxx&....

With this method you can test the Web services directly using an HTML browser.

For this documentation, you will see example with the JSON and XML structure.

SSL

If you need to access the web services with SSL, it can be configure on TCP port 8001 (see Annexes – SSL Configuration).

Then all web services will be accessible like this:

Generic example in HTTPS GET in JSON:

https://localhost:8001/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_Function?parameter1=xxx&....

Generic example in HTTPS GET in XML:

https://localhost:8001/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Xml/TWS_Function?parameter1=xxx&....

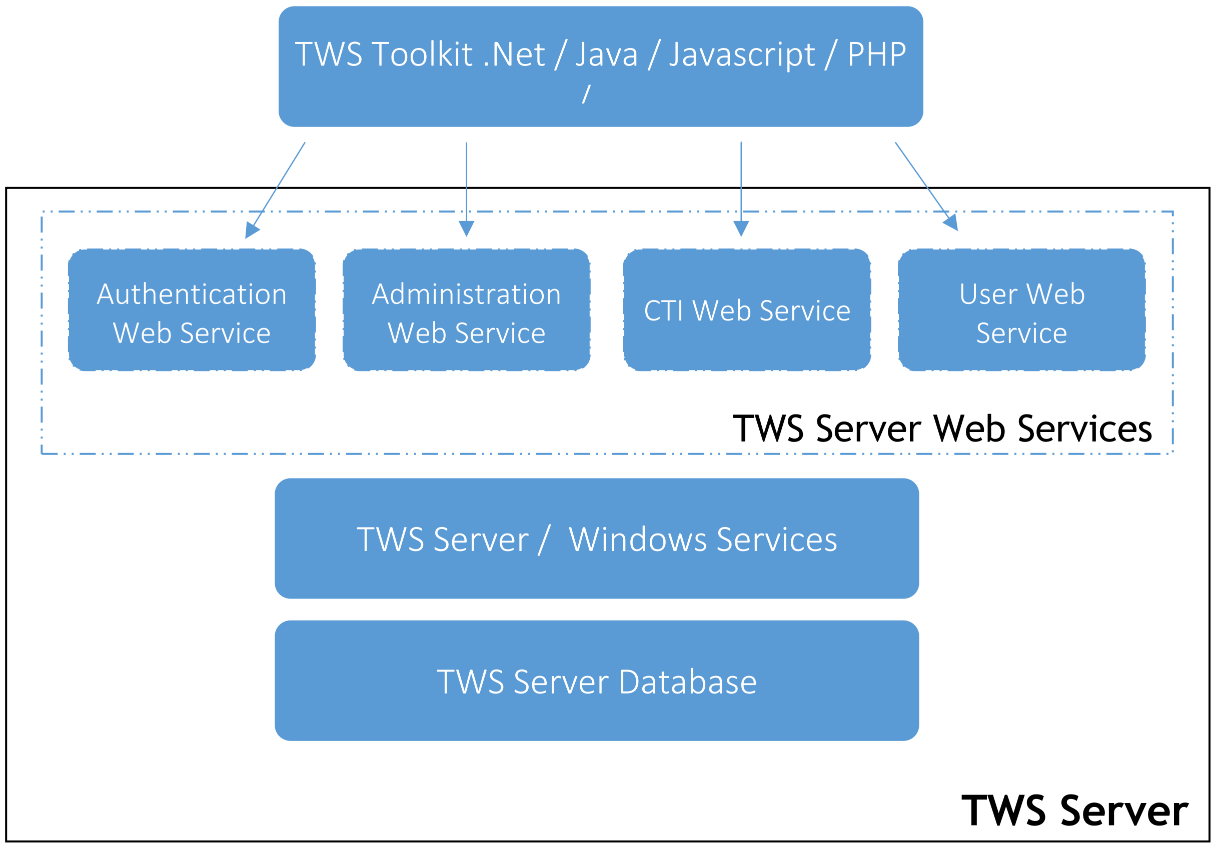
Browser security

If you have some problem to call the Web services, don’t forget to check your firewall and your internet security on the TWS Server and the client computer.

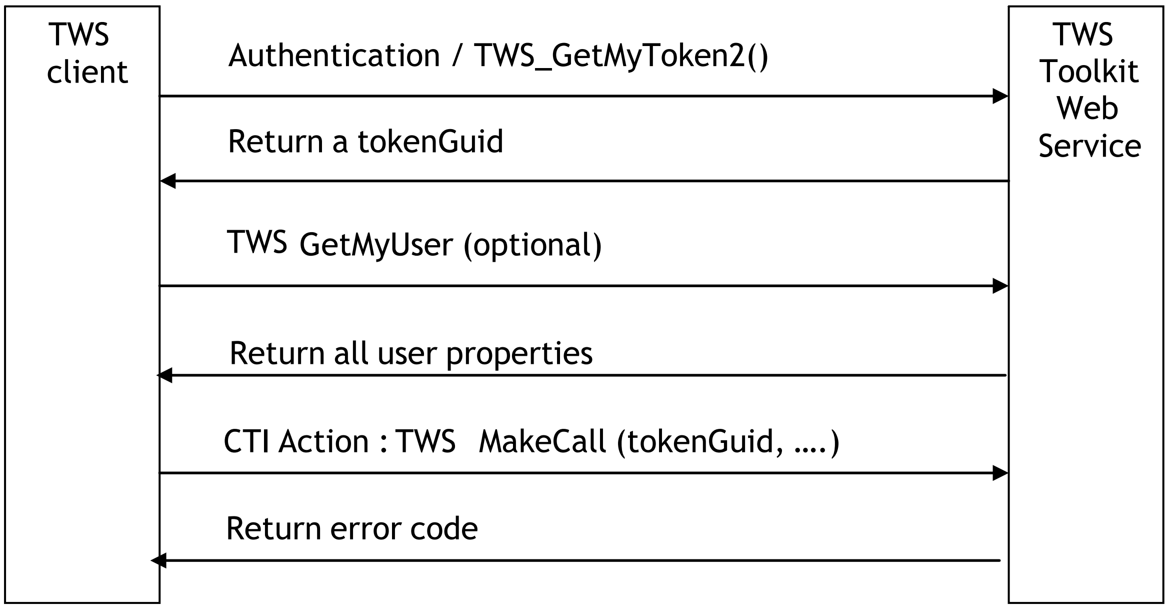
- Firewall: free the (http) port 8000
- Browser security: Disable the Protected Mode by unchecking the option (“Enable Protected Mode”) in the Internet security Properties, for the zones “Internet” and “Local intranet”.

3. TWS Toolkit architecture and protocol

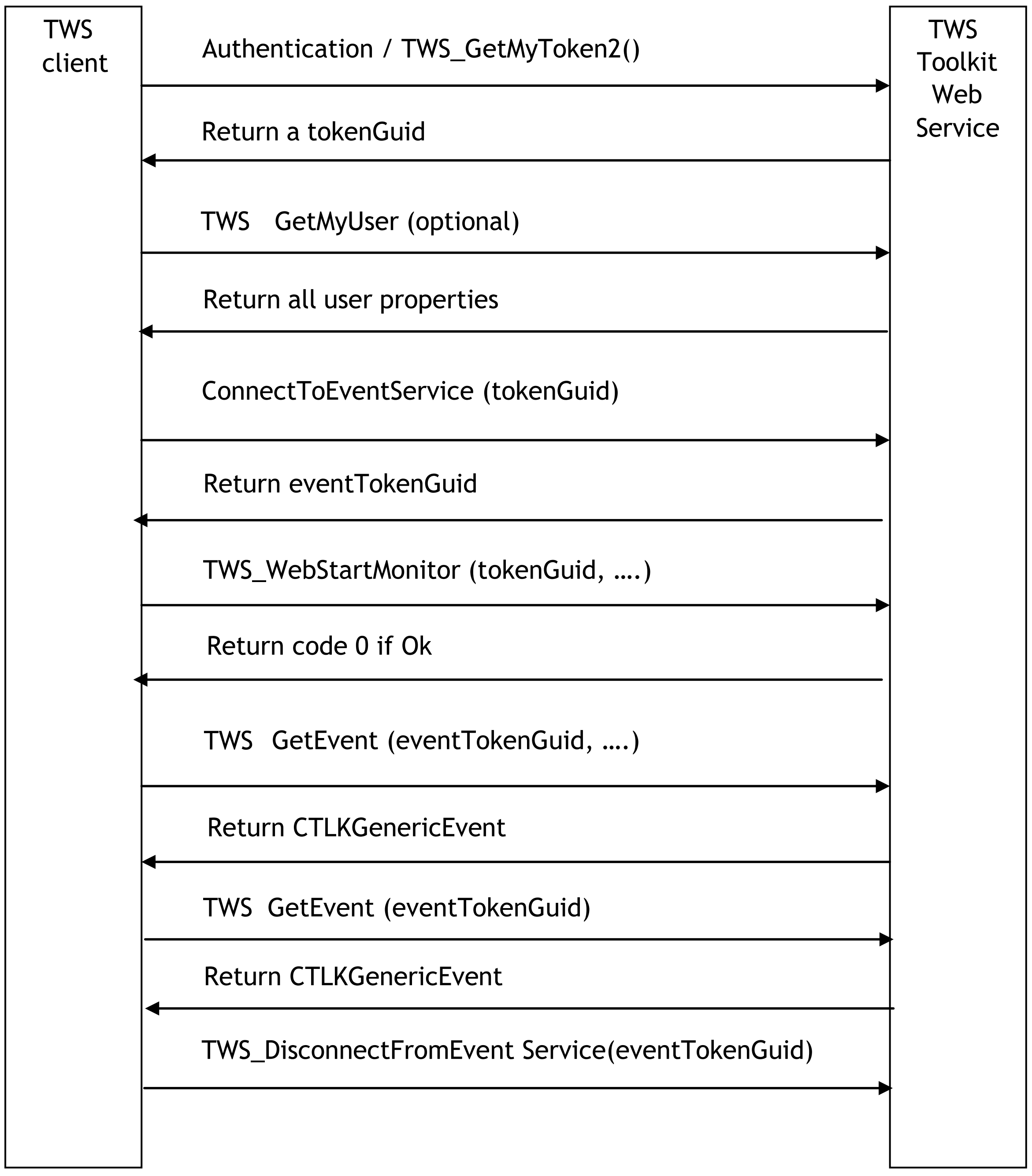
TWS Toolkit schema


TWS Toolkit .Net / Java / Javascript / PHP /
Authentication Web Service
Administration Web Service
CTI Web Service
User Web Service
TWS Server Web Services
TWS Server / Windows Services
TWS Server Database
TWS Server

3.1. TWS Toolkit Protocol without CTI monitoring a device

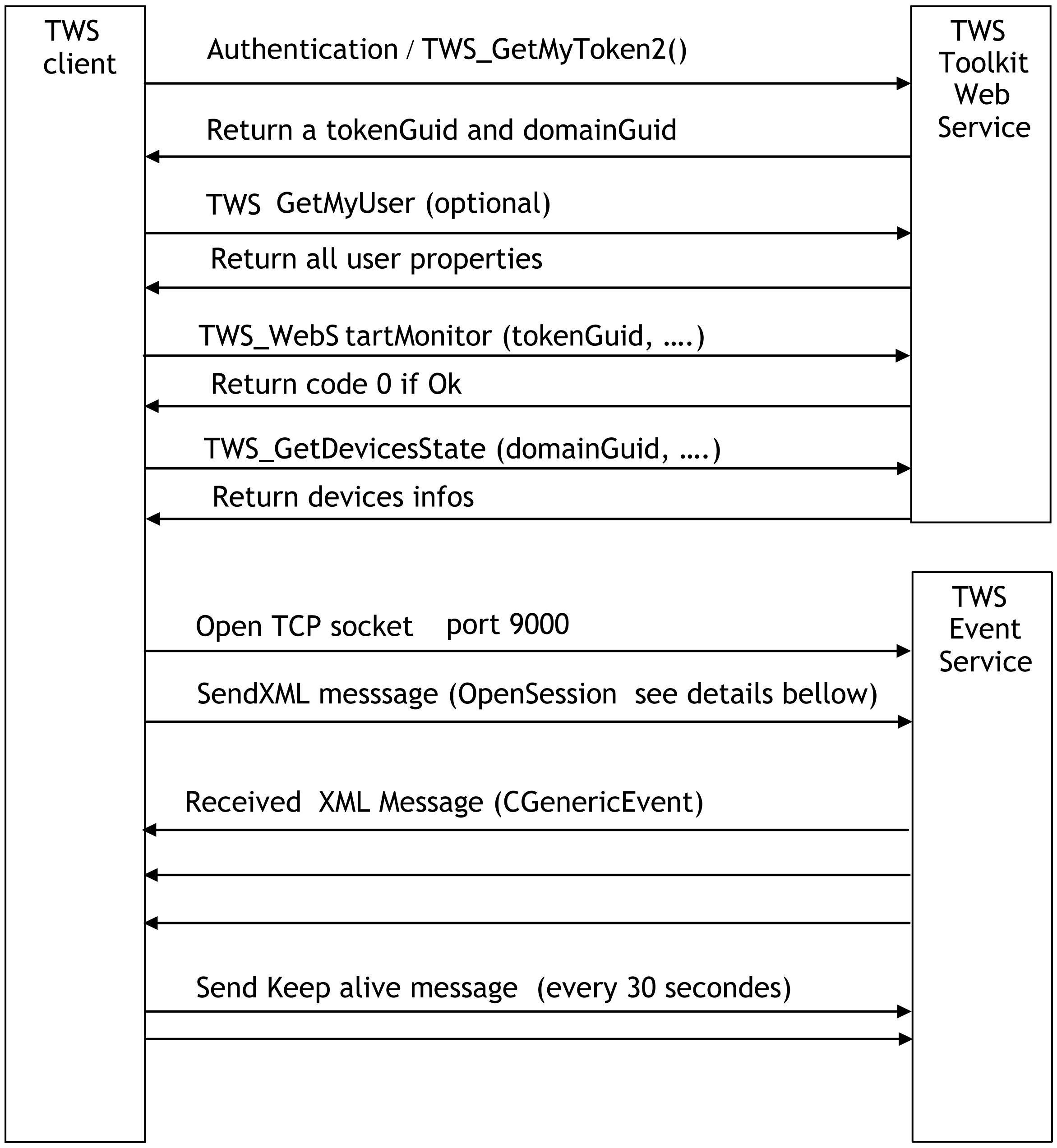

TWS client
TWS Toolkit Web Service
Authentication / TWS_GetMyToken2()
Return a tokenGuid
TWS GetMyUser (optional)
Return all user properties
CTI Action : TWS MakeCall (tokenGuid, ….)
Return error code

3.2. TWS Toolkit Protocol with CTI monitoring using Web Services


TWS client
TWS Toolkit Web Service
Authentication / TWS_GetMyToken2()
Return a tokenGuid
TWS GetMyUser (optional)
Return all user properties
ConnectToEventService (tokenGuid)
Return eventTokenGuid
TWS_WebStartMonitor (tokenGuid, ….)
Return code 0 if Ok
TWS GetEvent (eventTokenGuid, ….)
Return CTLKGenericEvent
TWS GetEvent (eventTokenGuid)
Return CTLKGenericEvent
TWS_DisconnectFromEvent Service(eventTokenGuid)

3.3. TWS Protocol with CTI monitoring by connecting to TWS Server port (listening by Event Service)

Receive all events of a user


TWS client
TWS Toolkit Web Service
Authentication / TWS_GetMyToken2()
Return a tokenGuid and domainGuid
TWS GetMyUser (optional)
Return all user properties
TWS_WebS tartMonitor (tokenGuid, ….)
Return code 0 if Ok
TWS_GetDevicesState (domainGuid, ….)
Return devices infos
TWS Event Service
Open TCP socket port 9000
SendXML messsage (OpenSession see details bellow)
Received XML Message (CGenericEvent)
Send Keep alive message (every 30 secondes)

Open Session message: Send this message immediately after the connection on TWS Server TCP Port 9000.

```
<TWS_WaitForGenericEvent><tokenGuid>{token}</tokenGuid></TWS_WaitFor
GenericEvent>{messageSeparator}
```

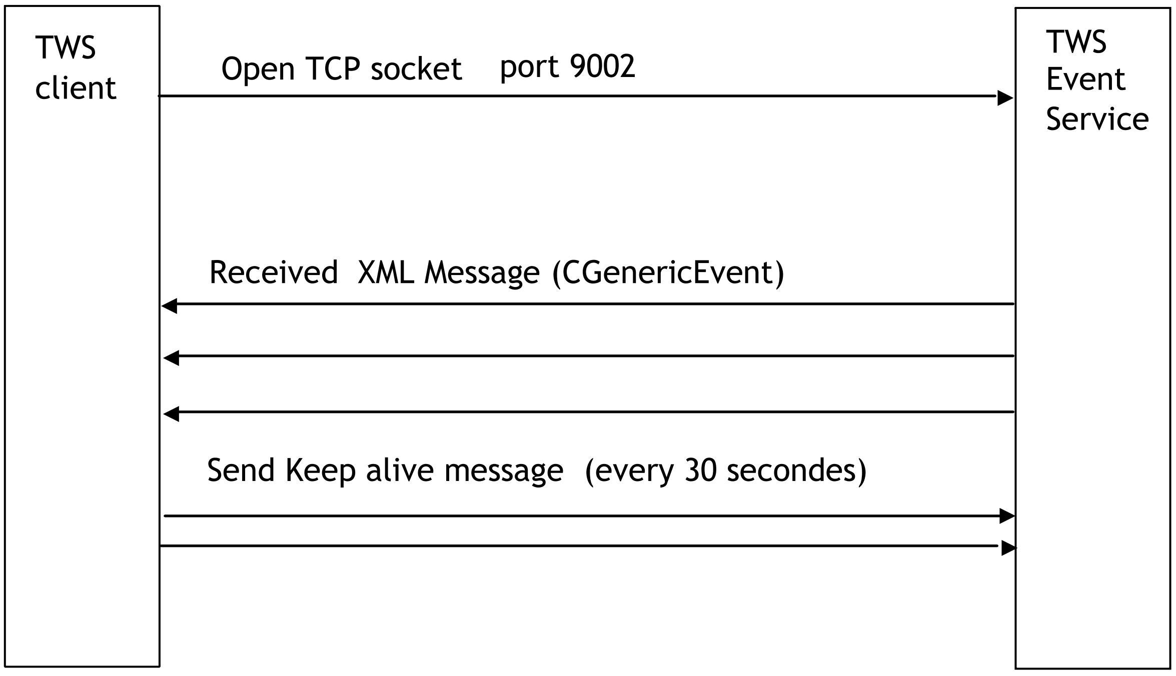
Replace {token} by the tokenGuid retrieved from authentication Web Service.

Keep Alive message: Send this message every 30 sec to keep alive the connection.

```
<CGenericEvent><EventType>AUDIT</EventType></CGenericEvent>{messageS
eparator}
```

`{messageSeparator} = « 82E51812-45C9-4733 » + 0` // the string must be ending by zero byte.

Receive all events of the server


TWS client
Open TCP socket port 9002
TWS Event Service
Received XML Message (CGenericEvent)
Send Keep alive message (every 30 secondes)

Open Session message: No open session message to send to receive all events but the connection should be done on TWS Server TCP Port 9002.

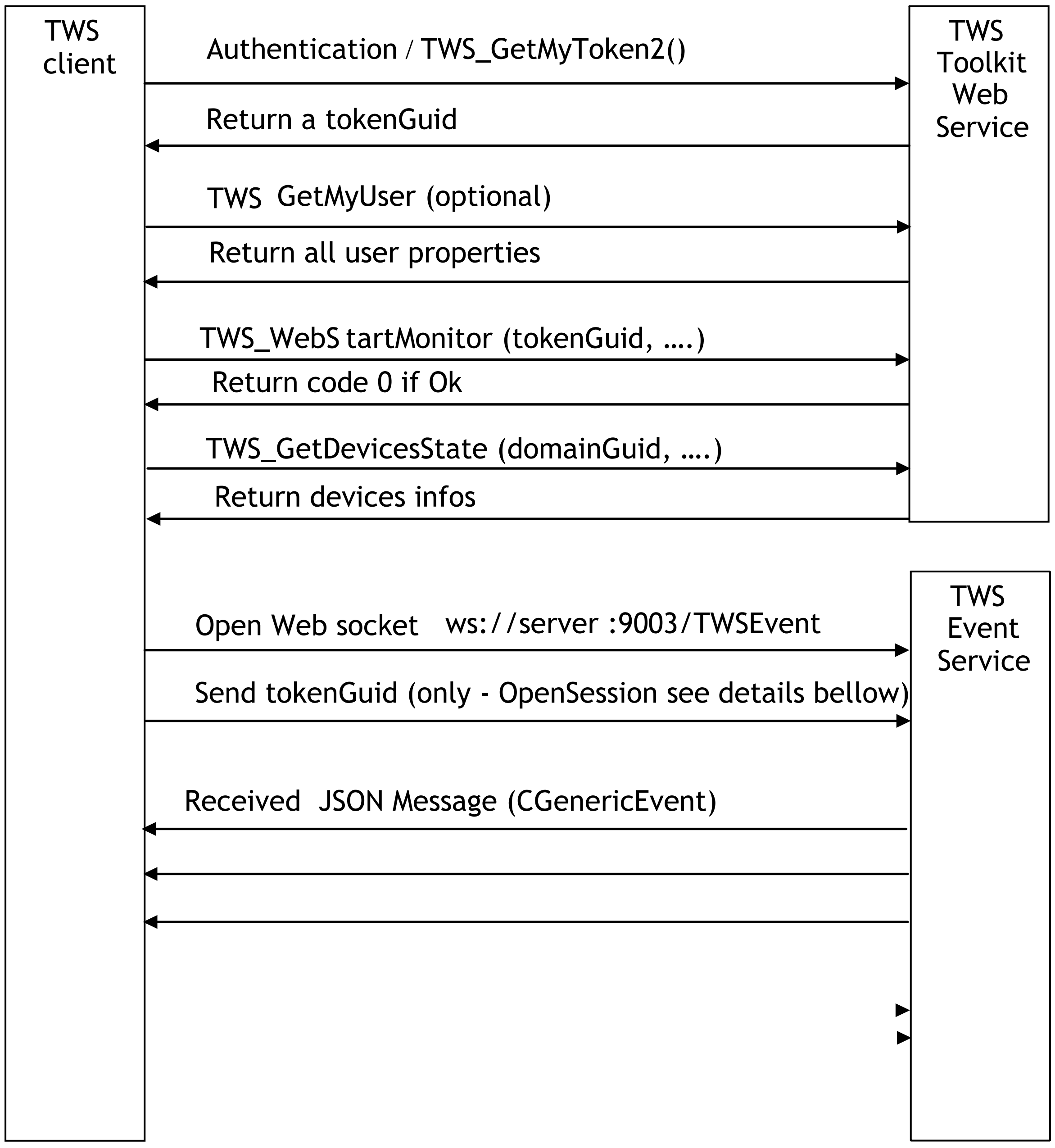
Keep Alive message: Send this message every 30 sec to keep alive the connection with a particular manner to encapsulate data before sending.

```
<CGenericEvent><EventType>AUDIT</EventType></CGenericEvent>
```

Attention: You need a specific formation for this to send or receive messages.

3.4. TWS Protocol with CTI monitoring by connecting to TWS Server WebSockets (listening by Event Service)

Receive all events of a user


TWS client
TWS Toolkit Web Service
Authentication / TWS_GetMyToken2()
Return a tokenGuid
TWS GetMyUser (optional)
Return all user properties
TWS_WebS tartMonitor (tokenGuid, ….)
Return code 0 if Ok
TWS_GetDevicesState (domainGuid, ….)
Return devices infos
TWS Event Service
Open Web socket ws://server :9003/TWSEvent
Send tokenGuid (only - OpenSession see details bellow)
Received JSON Message (CGenericEvent)

Open Session message: Just send the tokenGuid after the connection by a web socket on TWS Server 9003 port.

Web Socket SSL: You can also use the Web Socket with SSL on TWS Server 9004 port.

4. Description of Web services

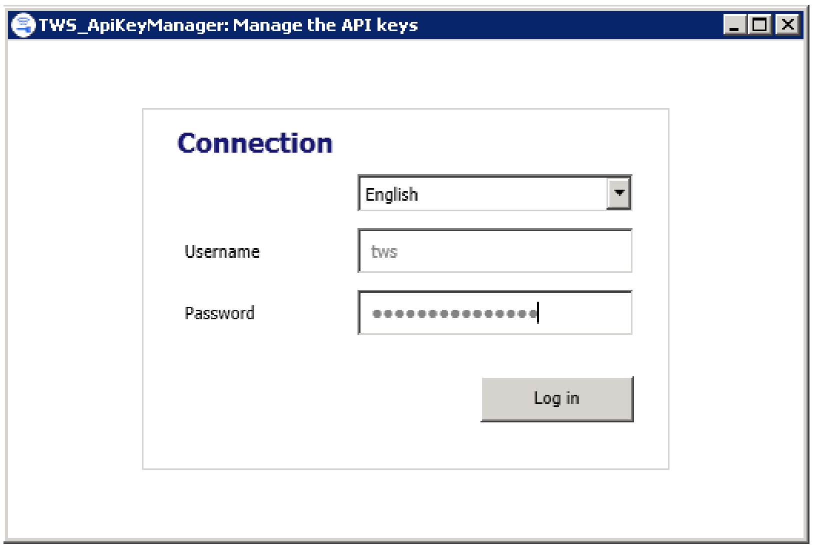
The Web Services are grouped into three categories:

1. User and security functions.
2. Administration functions
3. The telephony functions.
4. The utility functions.

API Keys

From version TWS v4.2.1610, it is advisable to create an API key and use it to get the security token of a user.

The API key can be created only on TWS Server with an application tool called 'TWS_ApiKeyManager'. Launch this tool by browsing the TWS folder and .\TWS_Tools\TWS_APIKeyManager.

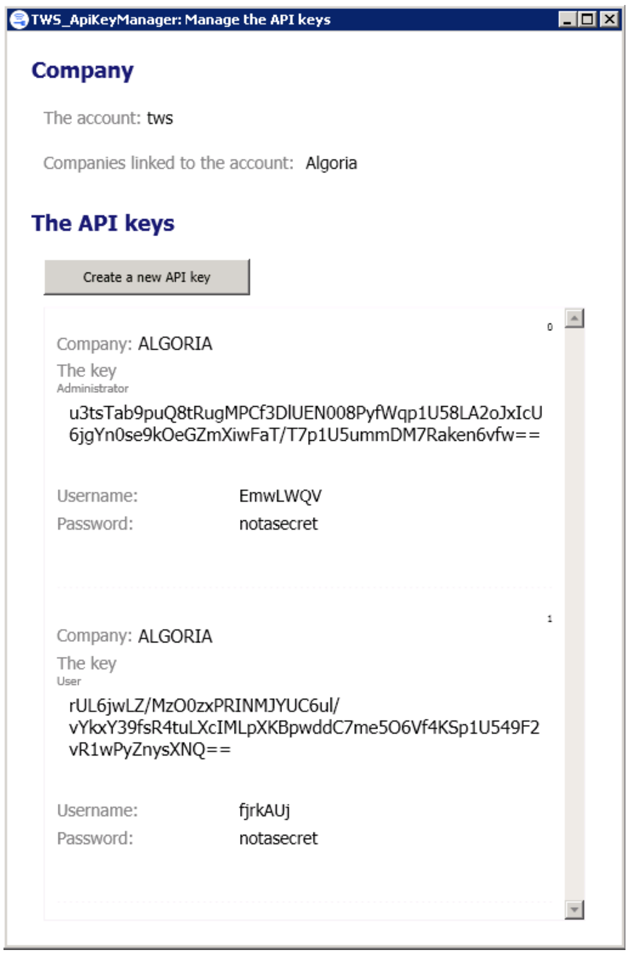


English

Username: tws

Password: •••••••••••••••

Log in with an account created in the administration page, 'Global' menu then 'Admin users'.



Company

The account: tws

Companies linked to the account: Algoria

The API keys

0

Company: ALGORIA
The key
Administrator

u3tsTab9puQ8tRugMPCf3DlUEN008PyfWqp1U58LA2oJxIcU
6jgYn0se9kOeGZmXiwFaT/T7p1U5ummDM7Raken6vfw==

Username:	EmwLWQV
Password:	notasecret

1

Company: ALGORIA
The key
User

rUL6jwLZ/MzO0zxPRINMJYUC6ul/
vYkxY39fsR4tuLXcIMLpXKBpwddC7me5O6Vf4KSp1U549F2
vR1wPyZnysXNQ==

Username:	fjrkAUj
Password:	notasecret

Click on the button 'Create a new API key' then you will get one. Just choose if it is an Administrator or a User apikey. Copy the key, the username and password to use it when needed in the Authentication functions like 'TWS_GetMyToken2' and 'TWS_GetMyTokenAdmin'.

Authentication functions

The WSDL access to this service is via the following address:

- for a windows authentication:

http://localhost:8000/tws_auth/TWS_UserWebSvc/TWS_UserWebSvc.svc?wsdl

- for other authentication than windows (None|TWS|LDAP):

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc?wsdl

The SingleWSDL is accessible at: …/TWS_UserWebSvc/TWS_UserWebSvc.svc?singlewsdl

Please see Administration and Configuration Guide to understand the different types of authenticating a user in TWS.

4.1. TWS_GetMyToken2 (recommended)

Description:

This is the first function to call, this function retrieves a token Id that needed by all other Toolkit function. The username and the password of the TWS user should be known. Fill the apikey parameter with the user apikey value.

Native call:

```
Num  = TWS_GetMyToken2(string apikey, string username (optional), string password (optional))
```
Ex:
```
Num =
TWS_GetMyToken2(“rUL6jwLZ/MzO0zxPRINMJYUC6ul/vYkxY39fsR4tLXcIMLpXKBp
wddC7me5O6Vfd059DBf6Y9eFr8NsjC/wUg==”, “”, “”)
```

If you used windows authentication you can let username and password parameters empty.

This HTTP call is only for Windows authentication or other:

http://localhost:8000/tws_auth/TWS_UserWebSvc/TWS_UserWebSvc.svc/Json/TWS_GetMyToken2?apiKey={APIKEY}&username={USERNAME}&password={PASSWORD}

This HTTP call is only for no-Windows authentication:

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc/XML/TWS_GetMyToken2?apiKey={APIKEY}&username={USERNAME}&password={PASSWORD}

Return: the <a:GUID> is the tokenGuid used in all functions where a token is requested.

```
<?xml version="1.0"?>
<TWS_GetMyToken2Response xmlns="http://tempuri.org/">
  <TWS_GetMyToken2Result xmlns:i="http://www.w3.org/2001/XMLSchema-
instance" xmlns:a=http://schemas.datacontract.org/2004/07/TWS_Database>
     <a:AffectedDocuments>0</a:AffectedDocuments>
     <a:ErrorMessage i:nil="true"/>
     <a:InfoMessage i:nil="true"/>
     <a:Ok>true</a:Ok>
     <a:WasNew>false</a:WasNew>
     <a:Result>
        <a:Devices
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
           <b:string>string</b:string>
        </a:Devices>
        <a:DomainGuid>string</a:DomainGuid>
        <a:Guid>string</a:Guid>
        <a:UserGuid>string</a:UserGuid>
        <a:Username>string</a:Username>
     </a:Result>
  </TWS_GetMyToken2Result>
</TWS_GetMyToken2Response>
```

4.2. TWS_GetMyToken (depreciated)

Description:

This function gives the same result as TWS_GetMyToken2 but it is less secure.

This is the first function to call, this function retrieved a token Id that needed by all other Toolkit function. The username and the password of the TWS user should be known.

Native call:

```
Num  = TWS_GetMyToken(string username (optional), string password
(optional))
```

Ex: `Num = TWS_GetMyToken("", "")`

If you used windows authentication you can let all parameters empty.

HTTP call is only for Windows authentication or other:

http://localhost:8000/tws_auth/TWS_UserWebSvc/TWS_UserWebSvc.svc/XML/TWS_GetMyToken?username={USERNAME}&password={PASSWORD}

HTTP call is only for no-Windows authentication:

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc/XML/TWS_GetMyToken?username={USERNAME}&password={PASSWORD}

Return: the <a:GUID> is the tokenGuid used in all functions where a token is requested.

```
<?xml version="1.0"?>
<TWS_GetMyTokenResponse xmlns="http://tempuri.org/">
  <TWS_GetMyTokenResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance" xmlns:a="http://schemas.datacontract.org/2004/07/TWS_Database">
    <a:DomainGuid>string</a:DomainGuid>
    <a:Guid>string</a:Guid>
    <a:UserGuid>string</a:UserGuid>
    <a:Username>string</a:Username>
  </TWS_GetMyTokenResult>
</TWS_GetMyTokenResponse>
```

4.3. TWS_GetMyTokenAdmin

Description:

Without knowing the password of a user, this function retrieves as an administrator a token Id that needed by all other Toolkit function. See above 'API Keys' chapter for 'apikey', 'adminUsername' and 'adminPassword' parameters, and 'username' must be known.

Native call:

```
Num  = TWS_GetMyTokenAdmin(string apikey, string adminUsername,
string adminPassword, string username)
```
Ex: `Num =`
```
TWS_GetMyTokenAdmin("rUL6jwLZ/MzO0zxPRINMJYUC6ul/vYkxY39fsR4tLXcIMLp
XKBpwddC7me5O6Vfd059DBf6Y9eFr8NsjC/wUg==", "mgplMKIT", "notasecret",
"hpiaple")
```

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc/XML/TWS_GetMyTokenAdmin??apiKey={APIKEY}&adminUsername={ADMINUSERNAME}&adminPassword={ADMINPASSWORD}&username={USERNAME}

Return: the GUID is the tokenGuid used in all functions where a token is requested.

```
<?xml version="1.0"?>
<TWS_GetMyTokenAdminResponse>
  <TWS_GetMyTokenAdminResult>
    <a:AffectedDocuments>0</a:AffectedDocuments>
    <a:ErrorMessage i:nil="true"/>
    <a:InfoMessage i:nil="true"/>
    <a:Ok>true</a:Ok>
    <a:WasNew>false</a:WasNew>
    <a:Result>
      <a:Guid>string</a:Guid>
      <a:UserGuid>string</a:UserGuid>
      <a:Username>string</a:Username>
    </a:Result>
  </TWS_GetMyTokenAdminResult>
</TWS_GetMyTokenAdminResponse>
```

4.4. TWS_GetMyToken2 (for administrator use)

Description:

Without knowing the username and the password of a user, you can also use this function to retrieve as an administrator an ephemeral token Id that needed by all other Toolkit function. See above 'API Keys' chapter for 'apikey' parameter.

Here is the procedure to have this behaviour:

1- Create or use a TWS user with TWS Caller or TWS Toolkit Runtime authorizations.

2- Create an administrator apiKey: See above ‘API Keys’ chapter.

3- With this function, use as parameters the administrator apikey, the username of the created user, and no password.

4- As a result, you will get an ephemeral token Id which could be used only once on all toolkit function. So request again TWS_GetMyToken2 if you have to call any other telephony function.

N.B.: Note that you have to pass the device number of a user as a parameter in the telephony functions you use. The users linked to these devices must have TWS Caller or TWS Toolkit Runtime authorizations.

Native call:

```
Num  = TWS_GetMyToken2(string apikey, string username, string
password (optional))
```
Ex: `Num =`
```
TWS_GetMyToken2(“rUL6jwLZ/MzO0zxPRINMJYUC6ul/vYkxY39fsR4tLXcIMLpXKBp
wddC7me5O6Vfd059DBf6Y9eFr8NsjC/wUg==”, “notusedusername”, “”)
```

With this behaviour no need to call this HTTP URL for Windows authentication:

This HTTP call is only for no-Windows authentication:

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc/XML/TWS_GetMyToken2?apiKey={APIKEY}&username={USERNAME}

Return: the <a:GUID> is the tokenGuid used in all functions where a token is requested.

```
<?xml version="1.0"?>
<TWS_GetMyToken2Response xmlns="http://tempuri.org/">
  <TWS_GetMyToken2Result xmlns:i="http://www.w3.org/2001/XMLSchema-
instance" xmlns:a=http://schemas.datacontract.org/2004/07/TWS_Database>
     <a:AffectedDocuments>0</a:AffectedDocuments>
     <a:ErrorMessage i:nil="true"/>
     <a:InfoMessage i:nil="true"/>
     <a:Ok>true</a:Ok>
     <a:WasNew>false</a:WasNew>
     <a:Result>
        <a:DomainGuid>string</a:DomainGuid>
        <a:Guid>string</a:Guid>
        <a:UserGuid i:nil="true"/>
        <a:Username i:nil="true"/>
     </a:Result>
  </TWS_GetMyToken2Result>
</TWS_GetMyToken2Response>
```

Get User functions

4.5. TWS_GetMyUser

Description:

This function retrieved the Contact object which defines the TWS user. In the CContact object get as results, there is a lot of information collected by the contact directories synchronization. User information's are in the property `UserInfos` :

- UserName
- FirstName
- LastName
- Culture
- …

Native call:

`TWS_GetMyUser(string guidToken)`

guidToken is retrieved from TWS_GetMyToken from Authentication WS.

Ex: `CContact = TWS_GetMyUser ("`**`0ab99936-a653-4b28-9e0f-c187d12b54ad`**`")`

HTTP call:

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc/Json/TWS_GetMyUser?guidToken={GUIDTOKEN}

4.6. TWS_GetMyProfile

Description:

This function retrieved all TWS user profile information like settings linked to the user, presence information or favorite contacts. And you can get user information's in the `User.UserInfos` properties:

- UserName
- FirstName
- LastName
- Culture
- …

Native call:

```
TWS_GetMyProfile(string guidToken)
```

guidToken is retrieved from TWS_GetMyToken from Authentication WS.

Ex: `CUserProfile = TWS_GetMyProfile ("`**`0ab99936-a653-4b28-9e0f-c187d12b54ad`**`")`

HTTP call:

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc/Json/TWS_GetMyProfile

4.7. TWS_NormalizeNumber

Description:

This function returns the normalized number from a character string. The number takes account of the numbering plan rules defined in the TWS administration.

Native call:

```
Num  = TWS_NormalizeNumber(string TokenGuid, string
stringToNormalize)
```
Ex: `Num = TWS_NormalizeNumber(`**`'3911534b-e16c-4427-bc6a-59a78628b2bb'`**`,`
`'fsqfqq 4001 qsfqsf')`

`Returns: 4001`
(the string is cleaned and the external dialing prefix is added).

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_NormalizeNumber?tokenGuid=3911534b-e16c-4427-bc6a-59a78628b2bb&szTo=4031

Return:

```
{"TWS_NormalizeNumberResult":"4031"}
```

4.8. TWS_SearchPeople

Description:

This function requests a resolution of the name according to the telephone number.

This function is used for the TWS incoming call logs.

Native call:

```
Num  = TWS_SearchPeople(string TokenGuid, string stringToSearch)
Ex: Num = TWS_SearchPeople('3911534b-e16c-4427-bc6a-59a78628b2bb',
'franc')
```

HTTP call:

http://localhost:8000/tws/TWS_UserWebSvc/TWS_UserWebSvc.svc/XML/TWS_SearchPeople?guidToken={GUIDTOKEN}&search={SEARCH}

Return:

```
<TWS_SearchPeopleResponse>
  <TWS_SearchPeopleResult>
    <a:Count>2</a:Count>
    <a:Results>
      <a:CContact>…</a:CContact>
      <a:CContact>…</a:CContact>
    </a:Results>
    <a:ResultsGuids i:nil="true"/>
    <a:TimeDatabase>15</a:TimeDatabase>
    <a:TimeIndex>31</a:TimeIndex>
  </TWS_SearchPeopleResult>
</TWS_SearchPeopleResponse>
```

5. Programming telephony functions

Security management

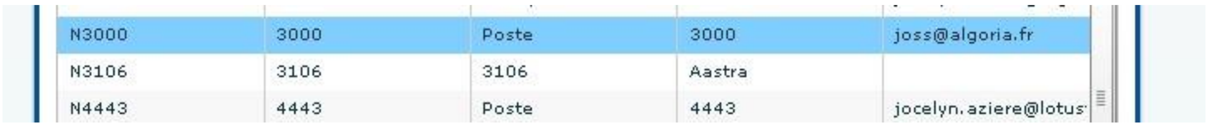
Users who want to execute a Web service must be used to retrieve a token linked to its TWS user. Normally a user may only execute a Web service on the telephone number which corresponds to their login (in the Windows sense). This is to avoid certain users being tempted to control sets other than their own.

Example: To run TWS_WebStartMonitor on set 3000:

1. The user connection must be authenticated at Windows level.
2. This user must be configured with the 3000 set in TWS administration.

In certain cases, applications which are run on the client operating system or on a server as a service may want to supervise several sets (application intended for a secretary responsible for a sales team, for example). A specific Windows user must be declared, who will be used by this application when it connects to the TWS server, and '0000' must be put in the telephone number field corresponding to this user in the TWS administration (see screen below).

If the CTI application is running on a machine different to the TWS server, it is possible to use the authentication of an application.



N3000	3000	Poste	3000	joss@algoria.fr
N3106	3106	3106	Aastra	
N4443	4443	Poste	4443	jocelyn.aziere@lotus

__Rules for using the TWS Toolkit:__

The following sequence must be respected to use the TWS Toolkit functions:

1. The application must retrieved a tokenGuid (see authentication web service)
2. The application subscribes to the TWS services using the TWS_WebStartMonitor function.
3. The application may then send orders to the telephone system (PBX).
4. If the application wants to receive the events, it must use the TCP socket on TWS event services.
5. The application indicates its shutdown using the TWS_WebStopMonitor function.

The logs continue to increment and must disconnect from the Event Server.

Service subscription requests

5.1. TWS_WebStartMonitor

Parameters:

tokenGuid: use the tokenGuid linked to the user you want to monitor.

szApp: use the 'TWS-TLK' value, using another value would reject your request.

szDevice: number of the subscriber to be supervised. This number may be either a telephone system, or an agent, or an automaton or any other part which may be supervised by your telephone system.

Native call:

Example of a supervision request without an alert request:

```
TWS_WebStartMonitor ('XXXX-XXXXX-XXXXX-XXXX', 'TWS-TLK', '3000')
```

This function returns 0 if OK, <> 0 if there is a problem and creates an exception if security is breached.

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_WebStartMonitor?tokenGuid={TOKENGUID}&szApp={SZAPP}&szDevice={SZDEVICE}

Return:

```
{"TWS_WebStartMonitorResult":0}
```

Stopping service subscription

5.2. TWS_WebStopMonitor

Parameters:
tokenGuid: use the tokenGuid linked to the user you want to monitor.

szApp: use the 'TWS-TLK' value, using another value would reject your request.

szDevice: number of the subscriber already supervised. This number may be either a telephone system, or an agent, or an automaton or any other part which may be supervised by your telephone system.

.

Native call:

Example of a supervision request without an alert request:

```
TWS_WebStopMonitor ('XXXX-XXXXX-XXXXX-XXXX' , 'TWS-TLK', '3000')
```

This function returns 0 if OK, <> 0 if there is a problem and creates an exception if security is breached.

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_WebStopMonitor?tokenGuid={TOKENGUID}&szApp={SZAPP}&szDevice={SZDEVICE}

Return:

```
{"TWS_WebStopMonitorResult":0}
```

Reinitializing a supervision

5.3. TWS_WebResetDeviceMonitor

Parameter:

tokenGuid: use the tokenGuid linked to the user you want to monitor.

szDevice: number of the subscriber to be supervised. This number may be either a telephone system, or an agent, or an automaton or any other part which may be supervised by your telephone system.

Native call:

```
TWS_WebResetDeviceMonitor('XXXX-XXXX-XXXX', '2000');
```

This function returns 0 if OK, < 0 if there is a problem and creates an exception if security is breached.

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_WebResetDeviceMonitor?tokenGuid={TOKENGUID}&szDevice={SZDEVICE}

Return:

```
{"TWS_WebResetDeviceMonitorResult":0}
```

Telephony operations

<u>Naming rule:</u>

<u>Functions terminating with DefaultDevice:</u>

These functions determine automatically the set to be controlled according to the user who called it.

<u>Functions not terminating with DefaultDevice</u>:

For these functions, you must specify the set on which the action is to be performed.

In this case, you must ensure that the TWS Web services call security rules are respected.

Example:

`TWS_MakeCall (szFrom, szTo)`	requests specifying the calling set while
`TWS_MakeCallByDefaultDevice (szTo)`	only requests the called number.

<u>Functions containing CallActive:</u> These function are used to manage the current active call.

Example: `TWS_HoldCallActiveByDefaultDevice (string tokenGuid)` // put the current call on hold

<u>Parameters rule:</u>

- *tokenGuid*: id retrieved from TWS_GetMyToken
- *szCall*: call reference
- *szDevice*, *szFrom*: device form witch the action will be done
- *szTo*: phone number of the called contact

<u>Return rule:</u>

- 0 : Ok
- -1 : serverException
- -4 : deviceNotFound

5.4. TWS_MakeCallByDefaultDevice, TWS_MakeCall

Description: These functions are used to launch an outgoing call.

Parameters:

tokenGuid: token guid of the TWS user.

szFrom:	*calling number*
szTo:	*number called*

Native call:

```
X = TWS_MakeCallByDefaultDevice ('2000')

X = TWS_MakeCall ('1000', '2000')

X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_MakeCallByDefaultDevice?tokenGuid=xxx-xxxxx-xxx&szTo=2000

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_MakeCall?tokenGuid=xxxxxxxx-xxx&szFrom=1000&szTo=2000

Return:

```
{"TWS_MakeCallByDefaultDeviceResult":0}
```

5.5. TWS_OpenMakeCallByDefaultDevice, TWS_OpenMakeCall

Description: These functions are used to launch an outgoing call and do not require any supervision in progress. To use this function, the user must be allocated to the OpenCall authorization group (cf administration doc).

Parameters:

tokenGuid: token guid of the TWS user

szFrom: calling number

szTo: number called

Native call:

```
X = TWS_OpenMakeCall ('xxxx-xxxx-xxxx-xxxx', '2000')
```

```
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_OpenMakeCall?tokenGuid=xxx-xxxxx-xxx&szTo=2000

Return:

```
{"TWS_OpenMakeCallResult":0}
```

5.6. TWS_BisByDefaultDevice, TWS_Bis

Description: These functions are used to re-emit the last call made by a set.

Parameters:

tokenGuid: token guid of the TWS user

szDevice: calling set.

Native call:

```
X = TWS_BisByDefaultDevice ('xxxx-xxxxx-xxxx')
X = TWS_Bis ('xxxx-xxxxx-xxxx' ,'1000')
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_BisByDefaultDevice? tokenGuid=xxxx-xxxx-xxxx-xxxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_Bis?tokenGuid=xxxx-xxxx-xxxx-xxxx&szDevice=1000

Return:

```
{"TWS_BisByDefaultDeviceResult":0}
```

5.7. TWS_AnswerCallQueuedbyDefaultDevice, TWS_AnswerCallQueued

Description: This function is used to take the first incoming call in the queue.

Parameter:

tokenGuid: token guid of the TWS user

szDevice: number of the set which takes the call.

Native call:

```
X = TWS_AnswerCallQueuedbyDefaultDevice ('xxxx-xxxxx-xxxx')

X = TWS_AnswerCallQueued ('xxxx-xxxxx-xxxx' ,'1000')

X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_AnswerCallQueuedByDefaultDevice? tokenGuid=xxx-xxxxx-xxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_AnswerCallQueued?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

```
{"TWS_AnswerCallQueuedbyDefaultDeviceResult":0}
```

5.8. TWS_ClearConnection , TWS_ClearConnectionActive, TWS_ClearConnectionActiveByDefaultDevice

Description: These functions are used to terminate a connection (corresponding most often to "on-hooking" to terminate the call).

Parameters:

tokenGuid: token guid of the TWS user

szCall: connection reference.

szDevice: extension number

Native call:

```
X = TWS_ClearConnectionActiveByDefaultDevice ('xxxx-xxxxx-xxxx')

X = TWS_ClearConnectionActive ('xxxx-xxxxx-xxxx' ,'1000')

X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ClearConnexionActiveByDefaultDevice?tokenGuid=xxx-xxxxx-xxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ClearConnexionActive?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

```
{"TWS_ClearConnectionActiveByDefaultDeviceResult":0}
```

5.9. TWS_DivertCall, TWS_DivertCallQueued, TWS_DivertCallQueuedByDefaultDevice

Description: These functions are used to divert an incoming call.

Parameters:

tokenGuid: token guid of the TWS user

szCall: connection reference.

szDevice: extension number

szTo: calling extension number

Native call:

`X = TWS_DivertCallQueuedByDefaultDevice (‘xxxx-xxxxx-xxxx’ ,‘2000’)` - Diverts the incoming call to 2000.

`X = TWS_DivertCallQueued (‘xxxx-xxxxx-xxxx’ ,‘1000’,’3000’)` - Diverts the incoming call from extension 1000 to 3000.

`X = TWS_DivertCall (‘xxxx-xxxxx-xxxx’ , ‘199384872’, ‘1000’, 4000’)` – Diverts the incoming call from extension 1000 referenced by connection 199384872 to 4000.

`X = Request ID No., < 0 if execution impossible (see Web service method error code in Annexes).`

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_DivertCallQueuedByDefaultDevice ?tokenGuid=xxx-xxxxx-xxx&szTo=2000
http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_DivertCallQueued?tokenGuid=xxx-xxxxx-xxx&szDevice=1000&szTo=3000
http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_DivertCall?tokenGuid=xxxxxxxx-xxx&szCall=199384872&szDevice=1000&szTo=4000

Return:

```
{"TWS_DivertCallQueuedByDefaultDeviceResult":0}
```

5.10. TWS_HoldCallActive, TWS_HoldCallActivebyDefaultDevice

Description: This function puts the active connection on hold.

Parameter:

tokenGuid: token guid of the TWS user

szDevice: extension number

Native call:

```
X = TWS_HoldCallActiveByDefaultDevice ('xxxx-xxxxx-xxxx')
X = TWS_HoldCallActive ('xxxx-xxxxx-xxxx' ,'1000')
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_HoldCallActiceByDefaultDevice? tokenGuid=xxx-xxxxx-xxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_HoldCallActice?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

```
{"TWS_HoldCallActiveByDefaultDeviceResult":0}
```

5.11. TWS_RetrieveCall ,TWS_RetrieveCallHeld, TWS_RetrieveCallHeldbyDefaultDevice

Description: This function reactivates the first connection on hold.

Parameters:

tokenGuid: token guid of the TWS user

szCall: connection reference.

szDevice: extension number

Native call:

```
X = TWS_RetrieveCallHeldbyDefaultDevice ('xxxx-xxxxx-xxxx')

X = TWS_RetrieveCallHeld ('xxxx-xxxxx-xxxx' ,'1000')

X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_RetrieveCallHeldByDefaultDevice

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_RetrieveCallHeld?szDevice=1000

Return:

```
{"TWS_RetrieveCallHeldbyDefaultDeviceResult":0}
```

5.12. TWS_ConsultationCallActive, TWS_ConsultationCallActiveByDefaultDevice

Description: This function uses the active connection to send a consultation call ("R1/R2" type call). A consultation call is used to have two calls on the same line, which lets you either switch from one call to another or transfer the call, or move to a conference call.

Parameters:

tokenGuid: token guid of the TWS user

szFrom: extension number

szTo: correspondent's number

Native call:

```
X = TWS_ConsultationCallAvtiveByDefaultDevice ('xxxx-xxxxx-xxxx'
,'5000')
```

```
X = TWS_ConsultationCallAvtive ('xxxx-xxxxx-xxxx' ,'1000', '5000')
```

```
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ConsultationCallActiveByDefaultDevice?tokenGuid=xxx-xxxxx-xxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ConsultationCallActive?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

```
{"TWS_ConsultationCallAvtiveByDefaultDeviceResult":0}
```

5.13. TWS_AlternateCallActive, TWS_AlternateCallActiveByDefaultDevice

Description: These functions are used to alternate between two connections. You must first have made a consultation call (see previous function) to have two correspondents on the same line.

Parameter:

tokenGuid: token guid of the TWS user

szDevice: extension number

Native call:

```
X = TWS_AlternateCallActiveByDefaultDevice ('xxxx-xxxxx-xxxx')

X = TWS_AlternateCallActive ('xxxx-xxxxx-xxxx' ,'1000')

X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_AlternateCallActiveByDefaultDevice?tokenGuid=xxx-xxxxx-xxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_AlternateCallActive?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

```
{"TWS_AlternateCallActiveByDefaultDeviceResult":0}
```

5.14. TWS_ConferenceCallActive, TWS_ConferenceCallActiveByDefaultDevice

Description: These functions are used to move to a 3-way conference. You must first have made a consultation call (see the previous 2 functions) in order to have three correspondents on line on the same line.

Parameter:

tokenGuid: token guid of the TWS user

szDevice: extension number

Native call:

```
X = TWS_ConferenceCallActiveByDefaultDevice ('xxxx-xxxxx-xxxx')

X = TWS_ConferenceCallActive ('xxxx-xxxxx-xxxx' ,'1000')

X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ConferenceCallActiveByDefaultDevice? tokenGuid=xxx-xxxxx-xxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ConferenceCallActive?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

```
{"TWS_ConferenceCallActiveByDefaultDeviceResult":0}
```

5.15. TWS_SingleStepTrans, TWS_SingleStepTransActive, TWS_SingleStepTransActiveByDefaultDevice

Description: These functions are used to transfer a call to another blind correspondent.

Parameters:

tokenGuid: token guid of the TWS user

szCall: connection reference.

szFrom: extension number

szTo: correspondent's number

Native call:

```
X = TWS_SingleStepTransActiveByDefaultDevice ('xxxx-xxxxx-xxxx' ,'5000')
```

Transfers the active call to the 5000.

```
X = TWS_SingleStepTransActive ('xxxx-xxxxx-xxxx' ,'1000', '5000')
```

Transfers the active call from set 1000 to the 5000.

```
X = TWS_SingleStepTrans ('xxxx-xxxxx-xxxx' , '19499582', '1000', '5000')
```

Transfers the call (with reference 19499582), from extension 1000 to extension 5000.

```
X = Request ID No., < 0 if execution impossible (see Web service method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SingleStepTransActiveByDefaultDevice?tokenGuid=xxx-xxxxx-xxx&szTo=5000

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SingleStepTransActive?tokenGuid=xxx-xxxxx-xxx&szFrom=1000&szTo=5000

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SingleStepTrans?tokenGuid=xxx-xxxxx-xxx&szCall=19499582&szFrom=1000&szTo=5000

Return:

{"TWS_SingleStepTransActiveByDefaultDeviceResult":0}

Forward management functions

5.16. TWS_SetForwardOn, TWS_SetForwardOnByDefaultDevice

Description: These functions are used to programme call transfer rules.

Parameters:

tokenGuid: token guid of the TWS user

userGuid: Guid of the user

szTo: correspondent's number

intType: type of forward *Possible values for intType:*

0 :	immediate forward.
2 :	forward on busy.
4 :	forward on no answer.
6 :	forward on internal busy.
8 :	forward on external busy.
10 :	forward on no internal answer.
12 :	forward on no external answer.
14 :	immediate internal forward.
16 :	immediate external forward.

Native call:

```
X = TWS_SetForwardOn ('xxxx-xxxxx-xxxx' , 'yyy-yyyy-yyyyy-yyyy',
'5000', 1)
```

Activates a forward on busy for the user to extension 5000.

```
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SetForwardOn?tokenGuid=xxx-xxxxx-xxx&userGuid=yyyy-yyyyy-yyyyy&szTo=5000&intType=1

Return:

```
{"TWS_SetForwardOnResult":0}
```

5.17. TWS_SetForwardOff, TWS_SetForwardOffByDefaultDevice

Description: These functions are used to cancel the programming of a call transfer rule.

Parameters:

tokenGuid: token guid of the TWS user

userGuid: Guid of the user

szTo: correspondent's number

intType: type of forward *Possible values for intType:*

1 :	immediate forward.
3 :	forward on busy.
5 :	forward on no answer.
7 :	forward on internal busy.
9 :	forward on external busy.
11 :	forward on no internal answer.
13 :	forward on no external answer.
15 :	immediate internal forward.
17 :	immediate external forward.

Native call:

```
X = TWS_SetForwardOff ('xxxx-xxxxx-xxxx' , 'yyy-yyyy-yyyyy-yyyy' , 1)
```

Cancels the forward on busy defined on the TWS user.

```
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SetForwardOff?tokenGuid=xxx-xxxxx-xxx&userGuid=yyyy-yyyyy-yyyyy&intType=1

Return:

```
{"TWS_SetForwardOffResult":0}
```

5.18. TWS_QueryDeviceForwardInfo, TWS_QueryDeviceForwardInfoByDevice

Description: These functions are used to find out the forward programming state.

Parameter:

tokenGuid: token guid of the TWS user

userGuid: Guid of the user

szDevice: extension number

Native call:

```
ForwardInfo = TWS_QueryDeviceForwardInfoDevice ('xxxx-xxxxx-xxxx' ,
'4000');
ForwardInfo = TWS_QueryDeviceForwardInfo ('xxxx-xxxxx-xxxx' , 'yyy-
yyyy-yyyyy-yyyy');
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/XML/TWS_QueryDeviceForwardInfoByDevice?tokenGuid=xxx-xxxxx-xxx&szDevice=4000

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/XML/TWS_QueryDeviceForwardInfo?tokenGuid=xxx-xxxxx-xxx&userGuid=yyyy-yyyyy-yyyyy

Return:

```
<TWS_QueryDeviceForwardInfoResponse>
  <TWS_QueryDeviceForwardInfoResult>
    <a:Busy>4002</a:Busy>
    <a:BusyExt/>
    <a:BusyInt/>
    <a:Imm>4001<a:Imm/>
    <a:ImmExt/>
    <a:ImmInt/>
    <a:IsSoftPhone>false</a:IsSoftPhone>
    <a:IsUpdated>true</a:IsUpdated>
    <a:NoAns/>
    <a:NoAnsExt/>
    <a:NoAnsInt/>
    <a:PersonnalNumber/>
  </TWS_QueryDeviceForwardInfoResult>
</TWS_QueryDeviceForwardInfoResponse>
```

In this return example, there is an immediate forward from extension 4000 to extension 4001 and a forward on busy from extension 4000 to extension 4002.

Extension state query functions

5.19. TWS_QueryCCosByDevice

Description: This function provides the state of the calls in progress per line.

Parameter:

tokenGuid: token guid of the TWS user

szDevice: extension number

Native call:

```
CGenericCCo [] cco = TWS_QueryCCosByDevice ('xxxx-xxxxx-xxxx', 3000)
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_QueryCCosByDevice?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

The information is returned in a table of the following type:

```
    public class  CGenricCco
      {
    public string m_CcoId; //  public
    string m_CcoState; // public
    string  m_CcoType; // CCo type
    public CGenericCnnection[] m_Connections = new Connection[2];
      }
```

The *Connection* class is of this type:

```
    public class Connection   // information about a connection
      {
    public string m_ConnectionId; // connection reference
    public string  m_currentState; // connection state
    public string  m_currentReason; // reserved
    public string m_connectionDirection; // connection direction
    public string m_Device; // device in question
    public CGenricUser [] m_Users;   // correspondent 1
      }
    public class CGenericUser
      {
    public string Guid; // contact Id
    public string  m_IsRedlist; // connection state
    public string  m_publicUserNumber; // contact phone number
    public string m_UserDisplay; // contact display number or Name
    public string m_UserInfoDisplay; // short info to display
      }
```

5.20. TWS_GetDevicesState

Description: This function provides the telephony monitoring state of a device.

Parameter:

domainGuid: domain guid where the TWS user is. It can be the result of TWS_GetMyToken2 .

protocol: the protocol used to start the monitoring of device. CSTA or VTIXML.

firstnumber: an extension number, the first of a range of numbers.

lastnumber: an extension number, the last of a range of numbers. It could be the same than firstnumber.

Native call:

```
CDeviceInfos [] deviceInfos = TWS_GetDevicesState ('xxxx-xxxxx-xxxx',
'CSTA', 1000, 1000)
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/XML/TWS_GetDevicesState?domainGuid=xxx-xxxxx-xxx&protocol=VTIXML&firstnumber=1000&lastnumber=1000

Return:

```
<TWS_GetDevicesStateResponse>
  <TWS_GetDevicesStateResult>
    <a:CDeviceInfos>
      <a:Cluster>2</a:Cluster>
      <a:Device>1000</a:Device>
      <a:Provider>192.1.3.251</a:Provider>
      <a:Site>1</a:Site>
      <a:State>Connected</a:State>
      <a:Type>cti</a:Type>
      <a:vmNumber>9999</a:vmNumber>
    </a:CDeviceInfos>
  </TWS_GetDevicesStateResult>
</TWS_GetDevicesStateResponse>
```

In this return example, the device 1000 is connected: the monitoring is on.

`Cluster`	int	Information linked to the PBX multisite
`Device`	string	Number of the monitored device
`Provider`	string	IP address of the PBX
`Site`	int	Information linked to the PBX multisite
`State`	string	State of the monitoring: • *Connected*: the monitoring is on • *Disconnected*, *ProviderLinkNull*, *OutOfService*: no monitoring. o Verify the connection link you create. o Is the number of the device correct? o Is the device connected or out of service? o Is there enough licenses in the PBX? o Is the type of the device supported? • *InvalidPassword*: the monitoring request should be done with the correct password.
`Type`	string	Type of the monitoring: cti/sipcti/voip/local
`vmNumber`	string	Number of the mailbox linked to the PBX

Data and cache functions

With these functions, you will be able to add, retrieve or delete a string data on a specified key on a 4 hours cache.

5.21. TWS_SetCallData

Description: This function set a string data on a specified key.

Parameter:

key: the unique identifier to store, access data.

data: the string to store.

add: if true the new string data will be concatenated to old data linked to the key.

Native call:

```
CGenericCCo [] cco = TWS_SetCallData ('xxxx', 'xxxx', false)
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SetCallData?key=xxxxx&data=xxxxx&add=false

Return:

0: if the action is done

-29: if not

5.22. TWS_GetCallData

Description: This function get the string data linked on a specified key.

Parameter:

key: the unique identifier to access data.

Native call:

```
CGenericCCo [] cco = TWS_GetCallData ('xxxx')
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_GetCallData?key=xxxxx

Return:

string: the data stored

5.23. TWS_ChangeKeyCallData

Description: This function change the sepcified key with another one and linked the data to it.

Parameter:

oldkey: the unique identifier to change linked to data.

newkey: the new key.

Native call:

```
CGenericCCo [] cco = TWS_ChangeKeyCallData ('xxxx', 'xxxx')
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ChangeKeyCallData?oldkey=xxxxx&newkey=xxxxx

Return:

0: if the action is done

-29: if not

Call parking functions

With these functions, you will be able to park and retrieve a call.

5.24. How to park: TWS_ParkCall, TWS_ParkCallActiveByDefaultDevice

Description: These functions park a call.

Parameter:

tokenGuid: token guid of the TWS user

szCall: connection reference

szDevice: extension number

szTo: extension number of the parking destination. The value could be empty if the call is parked to the user device.

Native call:

```
X = TWS_ParkCallActiveByDefaultDevice ('xxxx-xxxxx-xxxx' ,'5000')

X = TWS_ ParkCall ('xxxx-xxxxx-xxxx', 'c142', '1000', '5000')

X = Request ID No., < 0 if execution impossible (see Web service method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ParkCallActiveByDefaultDevice?tokenGuid=xxx-xxxxx-xxx&szTo=5000

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_ParkCall?tokenGuid=xxx-xxxxx-xxx&szCall=c142&szDevice=1000&szTo=5000

Return: (see Web service method error code in Annexes)

```
{"TWS_ParkCallActiveByDefaultDeviceResult":0}
{"TWS_ParkCallResult":0}
```

5.25. How to unpark: TWS_SendFacility

Description: This function retrieves a call parked by sending a PBX facility code.

Parameter:

tokenGuid: token guid of the TWS user

szPrefix: code prefix related to the PBX to unpark a call (for example: 402 for Alcatel)

szArg: extension number where the call is parked

Native call:

```
X = TWS_SendFacility ('xxxx-xxxxx-xxxx', '402', '5000')
```

```
X = Request ID No., < 0 if execution impossible (see Web service method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SendFacility?tokenGuid=xxx-xxxxx-xxx&szPrefix=402&szTo=5000

Return: (see Web service method error code in Annexes)

```
{"TWS_SendFacilityResult":0}
```

Agent management

With these functions, you will be able to set and query the state of any agent.

Required PBX: Alcatel.

5.26. TWS_AgentLogin

Description: Log on the agent and set the agent state to 'logged on' and then 'ready'.

Parameter:

tokenGuid: token guid of the TWS user agent

szDevice: extension number of the agent's device.

agentDevice: extension number of the device set. This could be different from the device of the TWS user agent.

agentID: extension number of the agent's device.

agentGroup: extension number of the group of the agent.

password: password used to log on the agent.

Native call:

```
X = TWS_AgentLogin('xxxx-xxxxx-xxxx', '1000', '1001', '1000', '1100',
'xxXxxX')
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_AgentLogin?tokenGuid=xx-xxx-xx&szDevice=1000&agentDevice=1001&agentID=1000&agentGroup=1100&password=xxXxxX

Return: (see Web service method error code in Annexes)

```
{"TWS_AgentLoginResult":0}
```

5.27. TWS_AgentLogout

Description: Log out the agent and set the agent state to 'logged off'.

Parameter:

tokenGuid: token guid of the TWS user agent

szDevice: extension number of the agent's device.

agentID: extension number of the agent's device.

password: password used to log on the agent.

Native call:

```
X = TWS_AgentLogout('xxxx-xxxxx-xxxx', '1000', '1000', 'xxXxxX')
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_AgentLogout?tokenGuid=xx-xxx-xx&szDevice=1000&agentID=1000&password=xxXxxX

Return: (see Web service method error code in Annexes)

```
{"TWS_AgentLogoutResult":0}
```

5.28. TWS_SetAgentState

Description: Set the agent state: busy, not ready, ready, working after call.

Attention: 'Logged on' and 'logged off' should not be used here to log on or log off the agent. For that use TWS_AgenLogin or TWS_AgentLogout.

Parameter:

tokenGuid: token guid of the TWS user agent

agentId: extension number of the agent's device. This could be different from the device of the TWS user agent.

state: the state to set. 0 to 5. AgentBusy = 0, ~~LoggedOff = 1, LoggedOn = 2~~, NotReady = 3, Ready = 4, WorkingAfterCall = 5

Native call:

```
X = TWS_SetAgentState('xxxx-xxxxx-xxxx', '1000', 3)
X = Request ID No., < 0 if execution impossible (see Web service
method error code in Annexes).
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_SetAgentState?tokenGuid=xxx-xxxxx-xxx&agentId=1000&state=3

Return: (see Web service method error code in Annexes)

```
{"TWS_SetAgentStateResult":0}
```

5.29. TWS_QueryAgentState, TWS_QueryAgentStateByDefaultDevice

Description: Get the current state of the TWS user agent.

Note that an event (described in chapter 5.30) is sent when this function is called.

Parameter:

tokenGuid: token guid of the TWS user

szDevice: extension number linked to the agent

Native call:

```
X = TWS_QueryAgentStateByDefaultDevice('xxxx-xxxxx-xxxx')
```

```
X = TWS_QueryAgentState('xxxx-xxxxx-xxxx', '1000')
```

HTTP call:

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_QueryAgentStateByDefaultDevice?tokenGuid=xxx-xxxxx-xxx

http://localhost:8000/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc/Json/TWS_QueryAgentState?tokenGuid=xxx-xxxxx-xxx&szDevice=1000

Return:

```
{"TWS_QueryAgentStateByDefaultDeviceResult":3}
{"TWS_QueryAgentStateResult":3}
```

```
EPresenceAgent {AgentBusy = 0, LoggedOff = 1, LoggedOn = 2, NotReady = 3,
Ready = 4, WorkingAfterCall = 5};
```

5.30. Receiving events

Description: When the state is changed, manually on the telephone set or with all above web service functions, an event is sent by TWS server.

If an application is connected to TWS server as described in chapter 3 above, the XML event that should be received is presented like this:

Logged off state

```
<CGenericEvent
xmlns="http://schemas.datacontract.org/2004/07/Algoria.TWS.GenericClass"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <creationDate>2017-06-10T14:11:28.8696341+02:00</creationDate>
  <eventType>CGenericAgentEvent</eventType>
  <genericAgentEvent>
    <AgentDevice i:nil="true"/>
    <AgentGroup>1100</AgentGroup>
    <AgentID>1000</AgentID>
    <AgentState>LoggedOff</AgentState>
    <Cause i:nil="true"/>
  </genericAgentEvent>
  <objectGuid>4133bb7d-c4a7-48e2-9ea3-2d2affe2b395</objectGuid>
  <objectName>1000</objectName>
  <userGuid>d10f0711-03b1-4d7c-ac92-766c5c17b3f0</userGuid>
</CGenericEvent>
```

Ready state

```xml
<CGenericEvent xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.datacontract.org/2004/07/Algoria.TWS.GenericClass">
  <creationDate>2017-06-10T14:08:51.7876495+02:00</creationDate>
  <eventType>CGenericAgentEvent</eventType>
  <genericAgentEvent>
    <AgentDevice i:nil="true" />
    <AgentGroup>1100</AgentGroup>
    <AgentID>1000</AgentID>
    <AgentState>Ready</AgentState>
    <Cause i:nil="true" />
  </genericAgentEvent>
  <objectGuid>4133bb7d-c4a7-48e2-9ea3-2d2affe2b395</objectGuid>
  <objectName>1000</objectName>
  <userGuid>d10f0711-03b1-4d7c-ac92-766c5c17b3f0</userGuid>
</CGenericEvent>
```

The important properties are:

- `creationDate`: at what time TWS server creates this event with the information received from the PBX
- `eventType`: it should be `CGenericAgentEvent`
- `AgentDevice`: extension number of the device set. This property will always be empty because it is never sent by the PBX.
- `AgentGroup`: the extension number of the group of the user agent.
- `AgentID`: the extension number of the user agent’s device.
- `AgentState`: the agent state is the string representation of the enum object: `EPresenceAgent { AgentBusy = 0 , LoggedOff = 1, LoggedOn = 2, NotReady = 3, Ready = 4, WorkingAfterCall = 5}`

6. Get call logs

How to get call logs with web service

There is many manner to export or get call logs from TWS Server but hereafter you will find how to get it with web services.

The web service to use is:

http://localhost:8000/tws/TWS_SocialCollaborationWebSvc/TWS_SocialCollaborationWebSvc.svc?wsdl

The SingleWSDL is accessible at:

http://localhost:8000/tws/TWS_SocialCollaborationWebSvc/TWS_SocialCollaborationWebSvc.svc?singlewsdl

6.1. GetMyDocumentsByTypeWithAcls

Description: This function get any documents of a user considering the parameters passed.

Parameter:

userGuid: guid of the TWS user

personGuid: guid of the TWS user or null *(not used)*

page: the number of the page to get (start from 1)

length: the number of documents per page

search: which document to get (CALL=3, CALLIN=1001, CALLOUT=1002, CALLMISSED=1003)

Native call:

```
CDocument[] documents = GetMyDocumentsByTypeWithAcls ('97a02200-fa08-4cdd-96ab-e44eb7f577dd', …)
```

HTTP call:

http://localhost:8000/tws/TWS_SocialCollaborationWebSvc/TWS_SocialCollaborationWebSvc.svc/XML/GetMyDocumentsByTypeWithAcls?userGuid=97a02200-fa08-4cdd-96ab-e44eb7f577dd&personGuid=&page=1&lenght=2&search=1000

Return:

```
<GetMyDocumentsByTypeWithAclsResponse>
  <GetMyDocumentsByTypeWithAclsResult>
    <a:CDocument>
      <a:Acls>
        <b:string>97a02200-fa08-4cdd-96ab-e44eb7f577dd</b:string>
      </a:Acls>
      <a:AclsDeny i:nil="true">
      <a:ActionType>CALL</a:ActionType>
      <a:Call>
        <a:Device>1000</a:Device>
        <a:Guid>4b29d5f2-dfd5-49f4-9ee2-af6698a07517</a:Guid>
        <a:IncomingCall>1</a:IncomingCall>
        <a:Speakers>
          <a:CSpeaker>
            <a:Cause>RINGING</a:Cause>
            <a:Company/>
            <a:DomainGuid i:nil="true"/>
            <a:Firstname/>
            <a:Lastname/>
            <a:On>2016-11-14T10:40:12.736Z</a:On>
            <a:PersonGuid>58d83796-28d1-43ad-b1e3-ac88a5ee0b86</a:PersonGuid>
            <a:Phone>0141906666</a:Phone>
            <a:PhoneContactInfoKey i:nil="true"/>
            <a:PhoneRedList>false</a:PhoneRedList>
            <a:Picture i:nil="true"/>
            <a:Type>UNKNOWN</a:Type>
            <a:UserGuid i:nil="true"/>
          </a:CSpeaker>
        </a:Speakers>
        <a:Success>1</a:Success>
        <a:TimeEnd>2016-11-14T10:40:54.756Z</a:TimeEnd>
        <a:TimeStart>2016-11-14T10:40:12.712Z</a:TimeStart>
        <a:TimeStart1>2016-11-14T10:40:42.749Z</a:TimeStart1>
        <a:UserGuid>97a02200-fa08-4cdd-96ab-e44eb7f577dd</a:UserGuid>
        <a:UserName/>
      </a:Call>
      <a:ChildDocuments i:nil="true"/>
      <a:CreateOn>2016-11-14T10:40:12.712Z</a:CreateOn>
      <a:Guid>7a3873e9-e6d5-4fc9-8d2b-0586e52f04cf</a:Guid>
      <a:IsOpen>false</a:IsOpen>
      <a:MediaFile i:nil="true"/>
      <a:Messages i:nil="true"/>
      <a:OwnerFirstName i:nil="true"/>
      <a:OwnerGuid>97a02200-fa08-4cdd-96ab-e44eb7f577dd</a:OwnerGuid>
      <a:OwnerLastName i:nil="true"/>
      <a:OwnerPicture i:nil="true"/>
      <a:RootGuid i:nil="true"/>
      <a:Sms i:nil="true"/>
      <a:Subject/>
      <a:Text i:nil="true"/>
      <a:ToFirstName/>
      <a:ToGuid>58d83796-28d1-43ad-b1e3-ac88a5ee0b86</a:ToGuid>
      <a:ToLastName/>
      <a:ToPhone>0141906666</a:ToPhone>
      <a:ToPicture i:nil="true"/>
    </a:CDocument>
  </GetMyDocumentsByTypeWithAclsResult>
</GetMyDocumentsByTypeWithAclsResponse>
```

`CDocument`	object	A Document object containing the call information
`ActionType`	EActionType	CALL=3, CALLIN=1001, CALLOUT=1002, CALLMISSED=1003
`Call`	object	A Call object containing all the information needed
`-Device`	string	The device number of the user who receives or makes the call
`-Guid`	string	Id of a call object
`-IncomingCall`	int	0 for incoming calls, 1 for outgoing calls
`-Success`	int	0 if not succeeded, 1 if succeeded
`-TimeEnd`	datetime	Date of the end of the call
`-TimeStart`	datetime	Date when the call arrives or is proceeded on the device
`-TimeStart1`	datetime	Date of the end of the ringing state
`-UserGuid`	string	Id of the user who receives or makes the call
`-Speakers`	Cspeaker[]	Array of correspondent(s) linked to the call
`--Company`	string	Company name of the correspondent
`--Firstname`	string	Firstname of the correspondent
`--Lastname`	string	Lastname of the correspondent
`--PersonGuid`	string	Id of the contact object of the correspondent
`--Phone`	string	Phone number of the correspondent
`--PhoneRedList`	bool	Is the contact in red list? “true” or “false”
`--UserGuid`	string	If the contact is a user then the id of this user

6.2. GetPersonDocumentsByTypeWithAcls

Description: This function get any documents made between a user and a contact considering the parameters passed.

Parameter:

userGuid: guid of the TWS user

personGuid: guid of the contact

page: the number of the page to get (start from 1)

length: the number of documents per page *search*: which document to get (CALL=3, CALLIN=1001, CALLOUT=1002, CALLMISSED=1003)

Native call:

```
CDocument[] documents = GetPersonDocumentsByTypeWithAcls ('97a02200-fa08-4cdd-96ab-e44eb7f577dd', …)
```

HTTP call:

http://localhost:8000/tws/TWS_SocialCollaborationWebSvc/TWS_SocialCollaborationWebSvc.svc/XML/GetPersonDocumentsByTypeWithAcls?userGuid=97a02200-fa08-4cdd-96ab-e44eb7f577dd&personGuid=58d83796-28d1-43ad-b1e3-ac88a5ee0b86&page=1&lenght=2&search=1000

Return: see 6.1 for the return result.

7. Management of events sent by TWS Server

Managing events with web service

Every application can subscribe to web service event. The application will received XML messages. To receive these events the application must call several web services:

1. Call TWS_ConnectToEventService
2. Call TWS_GetEvent in a loop to receive CallEvent / CallSupervisionEvent / …
3. Call TWS_DisconnectFromEventService to stop event reception

7.1. TWS_ConnectToEventService

Description: This function initializes the event service process. The best way to use this function is in SOAP.

Parameter:

tokenGuid: token guid of the TWS user

Native call:

```
String eventTokenGuid = TWS_ConnectToEventService ('xxxx-xxxxx-xxxx')
```

Return:

`eventTokenGuid` is the event id assigned to one session of getting events. This id would be used in other web services. Note that if no web services for getting events are called after a certain time, the id will be useless.

7.2. TWS_GetEvent

Description: This function is called to wait the next CTI event for a user device. This function should be called indefinitely in a loop code until you disconnect the web services to receive events. This function blocks 30 seconds if no events are received. The best way to use this function is in SOAP.

Parameter:

eventTokenGuid: the event id assigned to one session of getting events, returned by TWS_ConnectToEventService.

Native call:

```
CTLKGenericEvent genericEvent = TWS_GetEvent ('xxxx-xxxxx-xxxx')
```

Return:

`CTLKGenericEvent` could be one of the following event:

- `CTLKCallEvent`: Call event of a TWS user device
- `CTLKConnectionSupervisionEvent`: Supervision event of a TWS user device (it is a simple telephony event of user who are in your intercom group)
- `CTLKPresenceEvent`: Prevence event, TWS / Calendar event (Lotus / Exchange).

7.3. TWS_DisconnectFromEventService

Description: This function is called to disconnect web services from the event services on the TWS Server. Its stops events to be sent. While this function is called, the loop code of the TWS_GetEvent function must be stopped. The best way to use this function is in SOAP.

Parameter:

eventTokenGuid: the event id assigned to one session of getting events, returned by TWS_ConnectToEventService.

Native call:

```
int result = TWS_ DisconnectFromEventService ('xxxx-xxxxx-xxxx')
```

Annexes

Enumeration Object Description

```
public enum ECCoState
{
free = 0,
active = 1,
held = 2,
delivered = 3,
predelivered = 4,
undef = 5,
}
```

```
public enum EConnectionState
{
idle = 0,
initiated = 1,
originated = 2,
delivered = 3,
established = 4,
cleared = 5,
held = 6,
conferenced = 7,
transfered = 8,
diverted = 9,
queued = 10,
retrieved = 11,
blocked = 12,
predelivered = 13,
disconnected = 14,
notified = 15,
failed = 16,
all = 17,
}
```

```
public enum EPresenceType
{
NONE = 0,
CALENDAR = 1,
TWS = 2,
AGENT = 3,
}
```

```
public enum EPresenceAgent
{
AgentBusy,
LoggedOff,
LoggedOn,
NotReady,
Ready,
WorkingAfterCall
}
```

Event object example

CTLKCallEvent reveived from Web Service method

Delivered state

```xml
<?xml version="1.0" encoding="utf-16"?>
<CTLKCallEvent>
  <ExtensionData />
  <creationDate>2013-07-01T16:59:05.9376495+02:00</creationDate>
  <eventType>CallEvent</eventType>
  <objectGuid>62c98c72-4863-4626-8078-fe65c321b22b</objectGuid>
  <objectName>4694</objectName>
  <userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</userGuid>
  <genericCCOs>
    <CGenericCCo>
      <ExtensionData />
      <m_Connections>
        <CGenericConnection>
          <ExtensionData />
          <m_Users>
            <CGenericUser>
              <ExtensionData />
              <Addresses />
              <Company>Algo</Company>
              <CreatedOn>2013-04-26T08:17:08.71Z</CreatedOn>
              <Customs />
              <Devices />
              <DirectoryInfos>
                <ExtensionData />
                <Priority>0</Priority>
              </DirectoryInfos>
              <DomainGuid>2047d7f2-428b-49f6-beaf-7b2e102f954d</DomainGuid>
              <Emails />
              <ExternalUserInfos />
              <Firstname>test </Firstname>
              <IsUser>false</IsUser>
              <Lastname>Me</Lastname>
              <MergedInfos>
                <ExtensionData />
                <ContactsGuids>
                  <string>c3bb0324-a8f5-444c-8059-9797445a73de</string>
                </ContactsGuids>
                <DirectoriesGuids>
                  <string>479934ca-c683-4845-90b5-e8e8c01311bf</string>
                </DirectoriesGuids>
                <DirectoriesNames>
                  <string>[TwsPrivate]</string>
                </DirectoriesNames>
                <IsDeleted>false</IsDeleted>
                <Priority>0</Priority>
              </MergedInfos>
              <PersonGuid>c3bb0324-a8f5-444c-8059-9797445a73de</PersonGuid>
              <PersonalInfos>
                <ExtensionData />
                <ContactsListsGuids />
              </PersonalInfos>
              <Phones>
                <CContactInfo>
                  <ExtensionData />
                  <Key>work0</Key>
                  <Label>[Work]</Label>
```

```
                    <Type>PHONE</Type>
                    <Value>4594</Value>
                  </CContactInfo>
                </Phones>
                <Presences />
                <UpdatedOn>2013-04-26T08:17:08.71Z</UpdatedOn>
                <Urls />
                <UserInfos>
                  <ExtensionData />
                  <Enabled>true</Enabled>
                  <Init>false</Init>
                  <Type>USER</Type>
                </UserInfos>
                <m_IsRedList>false</m_IsRedList>
                <m_UserDisplay>4594</m_UserDisplay>
                <m_UserInfoDisplay>4594</m_UserInfoDisplay>
                <m_connectionDirection>incoming</m_connectionDirection>
                <m_publicUserNumber>4594</m_publicUserNumber>
              </CGenericUser>
              <CGenericUser d7p1:nil="true"
xmlns:d7p1="http://www.w3.org/2001/XMLSchema-instance" />
            </m_Users>
            <m_callAction>
              <ExtensionData />
              <Answer>true</Answer>
              <AnswerWithVideo>false</AnswerWithVideo>
              <Callback>false</Callback>
              <Conference>false</Conference>
              <DivertCall>false</DivertCall>
              <DropConference>false</DropConference>
              <Hold>false</Hold>
              <HoldConference>false</HoldConference>
              <Recover>false</Recover>
              <Release>false</Release>
              <SingleStepTransfer>false</SingleStepTransfer>
              <Swaphold>false</Swaphold>
              <Transfer>false</Transfer>
              <UnHold>false</UnHold>
              <UnholdConference>false</UnholdConference>
            </m_callAction>
            <m_callLogId>b0b40e79-b51a-4528-8d3c-0794cec8eabd</m_callLogId>
            <m_connectionDirection>incoming</m_connectionDirection>
            <m_connectionId>c1102</m_connectionId>
            <m_currentReason>Connected</m_currentReason>
            <m_currentState>delivered</m_currentState>
            <m_deviceNumber>4694</m_deviceNumber>
            <m_domainGuid>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainGuid>
            <m_domainName>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainName>
            <m_eventReason>NameResolved</m_eventReason>
            <m_mediaType>None</m_mediaType>
            <m_timeEnd>2013-07-01T16:59:05.9306999+02:00</m_timeEnd>
            <m_timeStart>2013-07-01T16:59:05.9306999+02:00</m_timeStart>
            <m_timeStart1>2013-07-01T16:59:05.9306999+02:00</m_timeStart1>
            <m_userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</m_userGuid>
          </CGenericConnection>
        </m_Connections>
        <m_ccoId>102</m_ccoId>
        <m_ccoState>delivered</m_ccoState>
        <m_ccoType>simplecall</m_ccoType>
      </CGenericCCo>
    </genericCCOs>
</CTLKCallEvent>
```

Established state

```xml
<?xml version="1.0" encoding="utf-16"?>
<CTLKCallEvent>
  <ExtensionData />
  <creationDate>2013-07-01T16:59:07.0576495+02:00</creationDate>
  <eventType>CallEvent</eventType>
  <objectGuid>62c98c72-4863-4626-8078-fe65c321b22b</objectGuid>
  <objectName>4694</objectName>
  <userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</userGuid>
  <genericCCOs>
    <CGenericCCo>
      <ExtensionData />
      <m_Connections>
        <CGenericConnection>
          <ExtensionData />
          <m_Users>
            <CGenericUser>
              <ExtensionData />
              <Addresses />
              <Company>Algo</Company>
              <CreatedOn>2013-04-26T08:17:08.71Z</CreatedOn>
              <Customs />
              <Devices />
              <DirectoryInfos>
                <ExtensionData />
                <Priority>0</Priority>
              </DirectoryInfos>
              <DomainGuid>2047d7f2-428b-49f6-beaf-7b2e102f954d</DomainGuid>
              <Emails />
              <ExternalUserInfos />
              <Firstname>test </Firstname>
              <IsUser>false</IsUser>
              <Lastname>Me</Lastname>
              <MergedInfos>
                <ExtensionData />
                <ContactsGuids>
                  <string>c3bb0324-a8f5-444c-8059-9797445a73de</string>
                </ContactsGuids>
                <DirectoriesGuids>
                  <string>479934ca-c683-4845-90b5-e8e8c01311bf</string>
                </DirectoriesGuids>
                <DirectoriesNames>
                  <string>[TwsPrivate]</string>
                </DirectoriesNames>
                <IsDeleted>false</IsDeleted>
                <Priority>0</Priority>
              </MergedInfos>
              <PersonGuid>c3bb0324-a8f5-444c-8059-9797445a73de</PersonGuid>
              <PersonalInfos>
                <ExtensionData />
                <ContactsListsGuids />
              </PersonalInfos>
              <Phones>
                <CContactInfo>
                  <ExtensionData />
                  <Key>work0</Key>
                  <Label>[Work]</Label>
                  <Type>PHONE</Type>
                  <Value>4594</Value>
                </CContactInfo>
              </Phones>
              <Presences />
              <UpdatedOn>2013-04-26T08:17:08.71Z</UpdatedOn>
              <Urls />
              <UserInfos>
                <ExtensionData />
                <Enabled>true</Enabled>
                <Init>false</Init>
                <Type>USER</Type>
              </UserInfos>
              <m_IsRedList>false</m_IsRedList>
              <m_UserDisplay>4594</m_UserDisplay>
              <m_UserInfoDisplay>4594</m_UserInfoDisplay>
              <m_connectionDirection>incoming</m_connectionDirection>
              <m_publicUserNumber>4594</m_publicUserNumber>
            </CGenericUser>
            <CGenericUser d7p1:nil="true"
xmlns:d7p1="http://www.w3.org/2001/XMLSchema-instance" />
          </m_Users>
          <m_callAction>
            <ExtensionData />
            <Answer>false</Answer>
            <AnswerWithVideo>false</AnswerWithVideo>
            <Callback>false</Callback>
            <Conference>false</Conference>
            <DivertCall>false</DivertCall>
            <DropConference>false</DropConference>
            <Hold>true</Hold>
            <HoldConference>false</HoldConference>
            <Recover>false</Recover>
            <Release>true</Release>
            <SingleStepTransfer>false</SingleStepTransfer>
            <Swaphold>false</Swaphold>
            <Transfer>false</Transfer>
            <UnHold>false</UnHold>
            <UnholdConference>false</UnholdConference>
          </m_callAction>
          <m_callLogId>b0b40e79-b51a-4528-8d3c-0794cec8eabd</m_callLogId>
          <m_connectionDirection>incoming</m_connectionDirection>
          <m_connectionId>c1102</m_connectionId>
          <m_currentReason>Connected</m_currentReason>
          <m_currentState>established</m_currentState>
          <m_deviceNumber>4694</m_deviceNumber>
          <m_domainGuid>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainGuid>
          <m_domainName>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainName>
          <m_eventReason>Normal</m_eventReason>
          <m_mediaType>None</m_mediaType>
          <m_timeEnd>2013-07-01T16:59:07.0562916+02:00</m_timeEnd>
          <m_timeStart>2013-07-01T16:59:05.9306999+02:00</m_timeStart>
          <m_timeStart1>2013-07-01T16:59:07.0562916+02:00</m_timeStart1>
          <m_userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</m_userGuid>
        </CGenericConnection>
      </m_Connections>
      <m_ccoId>102</m_ccoId>
      <m_ccoState>active</m_ccoState>
      <m_ccoType>simplecall</m_ccoType>
    </CGenericCCo>
  </genericCCOs>
</CTLKCallEvent>
```

Cleared state

```
<?xml version="1.0" encoding="utf-16"?>
<CTLKCallEvent>
  <ExtensionData />
  <creationDate>2013-07-01T16:59:08.7876495+02:00</creationDate>
  <eventType>CallEvent</eventType>
  <objectGuid>62c98c72-4863-4626-8078-fe65c321b22b</objectGuid>
  <objectName>4694</objectName>
  <userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</userGuid>
  <genericCCOs>
    <CGenericCCo>
      <ExtensionData />
      <m_Connections>
        <CGenericConnection>
          <ExtensionData />
          <m_Users>
            <CGenericUser>
              <ExtensionData />
              <Addresses />
              <Company>Algo</Company>
              <CreatedOn>2013-04-26T08:17:08.71Z</CreatedOn>
              <Customs />
              <Devices />
              <DirectoryInfos>
                <ExtensionData />
                <Priority>0</Priority>
              </DirectoryInfos>
              <DomainGuid>2047d7f2-428b-49f6-beaf-7b2e102f954d</DomainGuid>
              <Emails />
              <ExternalUserInfos />
              <Firstname>test </Firstname>
              <IsUser>false</IsUser>
              <Lastname>Me</Lastname>
              <MergedInfos>
                <ExtensionData />
                <ContactsGuids>
                  <string>c3bb0324-a8f5-444c-8059-9797445a73de</string>
                </ContactsGuids>
                <DirectoriesGuids>
                  <string>479934ca-c683-4845-90b5-e8e8c01311bf</string>
                </DirectoriesGuids>
                <DirectoriesNames>
                  <string>[TwsPrivate]</string>
                </DirectoriesNames>
                <IsDeleted>false</IsDeleted>
                <Priority>0</Priority>
              </MergedInfos>
              <PersonGuid>c3bb0324-a8f5-444c-8059-9797445a73de</PersonGuid>
              <PersonalInfos>
                <ExtensionData />
                <ContactsListsGuids />
              </PersonalInfos>
              <Phones>
                <CContactInfo>
                  <ExtensionData />
                  <Key>work0</Key>
                  <Label>[Work]</Label>
                  <Type>PHONE</Type>
                  <Value>4594</Value>
                </CContactInfo>
              </Phones>
              <Presences />
              <UpdatedOn>2013-04-26T08:17:08.71Z</UpdatedOn>
              <Urls />
              <UserInfos>
                <ExtensionData />
                <Enabled>true</Enabled>
                <Init>false</Init>
                <Type>USER</Type>
              </UserInfos>
              <m_IsRedList>false</m_IsRedList>
              <m_UserDisplay>4594</m_UserDisplay>
              <m_UserInfoDisplay>4594</m_UserInfoDisplay>
              <m_connectionDirection>incoming</m_connectionDirection>
              <m_publicUserNumber>4594</m_publicUserNumber>
            </CGenericUser>
            <CGenericUser d7p1:nil="true"
xmlns:d7p1="http://www.w3.org/2001/XMLSchema-instance" />
          </m_Users>
          <m_callAction>
            <ExtensionData />
            <Answer>false</Answer>
            <AnswerWithVideo>false</AnswerWithVideo>
            <Callback>false</Callback>
            <Conference>false</Conference>
            <DivertCall>false</DivertCall>
            <DropConference>false</DropConference>
            <Hold>false</Hold>
            <HoldConference>false</HoldConference>
            <Recover>false</Recover>
            <Release>false</Release>
            <SingleStepTransfer>false</SingleStepTransfer>
            <Swaphold>false</Swaphold>
            <Transfer>false</Transfer>
            <UnHold>false</UnHold>
            <UnholdConference>false</UnholdConference>
          </m_callAction>
          <m_callLogId>b0b40e79-b51a-4528-8d3c-0794cec8eabd</m_callLogId>
          <m_connectionDirection>incoming</m_connectionDirection>
          <m_connectionId>c1102</m_connectionId>
          <m_currentReason>Idle</m_currentReason>
          <m_currentState>cleared</m_currentState>
          <m_deviceNumber>4694</m_deviceNumber>
          <m_domainGuid>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainGuid>
          <m_domainName>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainName>
          <m_eventReason>Normal</m_eventReason>
          <m_mediaType>None</m_mediaType>
          <m_timeEnd>2013-07-01T16:59:08.7547566+02:00</m_timeEnd>
          <m_timeStart>2013-07-01T16:59:05.9306999+02:00</m_timeStart>
          <m_timeStart1>2013-07-01T16:59:07.0562916+02:00</m_timeStart1>
          <m_userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</m_userGuid>
        </CGenericConnection>
      </m_Connections>
      <m_ccoId>102</m_ccoId>
      <m_ccoState>free</m_ccoState>
      <m_ccoType>free</m_ccoType>
    </CGenericCCo>
  </genericCCOs>
</CTLKCallEvent>
```

Events reveived from TWS Server Event Services

GenericEvent of type CallEvent in a Cleared state

```xml
<CGenericEvent
xmlns="http://schemas.datacontract.org/2004/07/Algoria.TWS.GenericClass"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <creationDate>2013-07-01T16:59:08.7876495+02:00</creationDate>
  <eventType>CallEvent</eventType>
  <genericCallEvent>
    <CGenericCCo>
      <m_Connections>
        <CGenericConnection>
          <genericForwardedUser i:nil="true"/>
          <genericLastRedirectedUser i:nil="true"/>
          <genericCalledUser>
            <Company
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">Algo</Company>
            <CreatedOn
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">2012-05-
16T17:17:49.754Z</CreatedOn>
            <Customs xmlns="http://schemas.datacontract.org/2004/07/TWS_Database" />
            <Devices xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
              <CDevice>
                <Guid>9c804c52-b808-4e71-9624-962deff31911</Guid>
                <Number>4694</Number>
              </CDevice>
            </Devices>
            <DomainGuid
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">2047d7f2-428b-49f6-beaf-
7b2e102f954d</DomainGuid>
            <Emails xmlns="http://schemas.datacontract.org/2004/07/TWS_Database" />
            <Firstname
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">Hector</Firstname>
            <Guid
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">5e3795ab-d8a1-464c-b9e8-
e5cff19ebc93</Guid>
            <IsUser
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">true</IsUser>
            <Lastname
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">Piaple</Lastname>
            <MergedInfos xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
              <ContactsGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>5e3795ab-d8a1-464c-b9e8-e5cff19ebc93</a:string>
              </ContactsGuids>
              <DirectoriesGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>d2fcd657-8df1-4e62-b9af-04d8f808e419</a:string>
              </DirectoriesGuids>
              <DirectoriesNames
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>[TwsUsers]</a:string>
              </DirectoriesNames>
              <InternalKeys
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                <a:string>5e3795ab-d8a1-464c-b9e8-e5cff19ebc93</a:string>
              </InternalKeys>
              <Priority>1</Priority>
            </MergedInfos>
            <PersonGuid
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">5e3795ab-d8a1-464c-b9e8-
e5cff19ebc93</PersonGuid>
            <PersonalInfos
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database" />
            <Phones xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
              <CContactInfo>
                <Key>work0_d2fcd758-8df1-4e62-b9af-04d8f808e419</Key>
                <Label>[Work]</Label>
                <Type>TWS</Type>
                <Value>4594</Value>
              </CContactInfo>
            </Phones>
            <Picture xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
            <Presences xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
            <UpdatedOn
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">2013-07-
01T13:00:41.272Z</UpdatedOn>
            <Urls xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
            <UserInfos xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
              <Enabled>true</Enabled>
              <Username>hpiaple</Username>
            </UserInfos>
            <m_IsRedList>false</m_IsRedList>
            <m_UserDisplay>4694</m_UserDisplay>
            <m_UserInfoDisplay>Hector Piaple</m_UserInfoDisplay>
            <m_connectionDirection>outgoing</m_connectionDirection>
            <m_publicUserNumber>4694</m_publicUserNumber>
          </genericCalledUser>
          <m_Users>
            <CGenericUser>
              <Addresses
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <Company
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">Algo</Company>
              <CreatedOn
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">2013-04-
26T08:17:08.71Z</CreatedOn>
              <Customs xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <Devices xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <DirectoryInfos
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
                <DirectoryGuid i:nil="true"/>
                <ExternalKey i:nil="true"/>
                <InternalKeys i:nil="true"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                <OwnerGuid i:nil="true"/>
                <Priority>0</Priority>
                <Reverses i:nil="true"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                <Searches i:nil="true"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
              </DirectoryInfos>
              <DomainGuid
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">2047d7f2-428b-49f6-beaf-
7b2e102f954d</DomainGuid>
              <Emails xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <ExternalUserInfos
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <Firstname
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">test </Firstname>
              <Guid i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <IsUser
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">false</IsUser>
```

```
              <Lastname
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">Me</Lastname>
              <MergedInfos
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
                <ContactsGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                  <a:string>c3bb0324-a8f5-444c-8059-9797445a73de</a:string>
                </ContactsGuids>
                <DirectoriesGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                  <a:string>479934ca-c683-4845-90b5-e8e8c01311bf</a:string>
                </DirectoriesGuids>
                <DirectoriesNames
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
                  <a:string>[TwsPrivate]</a:string>
                </DirectoriesNames>
                <IsDeleted>false</IsDeleted>
                <Priority>0</Priority>
              </MergedInfos>
              <PersonGuid
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">c3bb0324-a8f5-444c-8059-
9797445a73de</PersonGuid>
              <PersonalInfos
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
                <ContactsListsGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
                <DefaultPhone i:nil="true"/>
                <DisplayName i:nil="true"/>
              </PersonalInfos>
              <Phones xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
                <CContactInfo>
                  <Key>work0</Key>
                  <Label>[Work]</Label>
                  <RawValue i:nil="true"/>
                  <RedList i:nil="true"/>
                  <Type>PHONE</Type>
                  <Value>4594</Value>
                </CContactInfo>
              </Phones>
              <Picture i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <Presences
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <UpdatedOn
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">2013-04-
26T08:17:08.71Z</UpdatedOn>
              <Urls xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
              <UserInfos xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
                <ActiveProfileGuid i:nil="true"/>
                <Culture i:nil="true"/>
                <Enabled>true</Enabled>
                <Init>false</Init>
                <Ip i:nil="true"/>
                <Password i:nil="true"/>
                <Type>USER</Type>
                <Username i:nil="true"/>
                <VoiceMailPassword i:nil="true"/>
              </UserInfos>
              <m_IsRedList>false</m_IsRedList>
              <m_UserDisplay>4594</m_UserDisplay>
              <m_UserInfoDisplay>test  Me</m_UserInfoDisplay>
              <m_connectionDirection>incoming</m_connectionDirection>
              <m_publicUserNumber>4594</m_publicUserNumber>
            </CGenericUser>
            <CGenericUser i:nil="true"/>
```

```
          </m_Users>
          <m_callAction>
            <Answer>false</Answer>
            <AnswerWithVideo>false</AnswerWithVideo>
            <Callback>false</Callback>
            <Conference>false</Conference>
            <DivertCall>false</DivertCall>
            <DropConference>false</DropConference>
            <Hold>false</Hold>
            <HoldConference>false</HoldConference>
            <Recover>false</Recover>
            <Release>false</Release>
            <SingleStepTransfer>false</SingleStepTransfer>
            <Swaphold>false</Swaphold>
            <Transfer>false</Transfer>
            <UnHold>false</UnHold>
            <UnholdConference>false</UnholdConference>
          </m_callAction>
          <m_callLogId>b0b40e79-b51a-4528-8d3c-0794cec8eabd</m_callLogId>
          <m_callLogIdRoot2 i:nil="true"/>
          <m_ccoId i:nil="true"/>
          <m_connectionDirection>incoming</m_connectionDirection>
          <m_connectionId>c1102</m_connectionId>
          <m_currentReason>Idle</m_currentReason>
          <m_currentState>cleared</m_currentState>
          <m_deviceNumber>4694</m_deviceNumber>
          <m_domainGuid>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainGuid>
          <m_domainName>2047d7f2-428b-49f6-beaf-7b2e102f954d</m_domainName>
          <m_eventReason>Normal</m_eventReason>
          <m_mediaType>None</m_mediaType>
          <m_queueGuid i:nil="true"/>
          <m_timeEnd>2013-07-01T16:59:08.7547566+02:00</m_timeEnd>
          <m_timeStart>2013-07-01T16:59:05.9306999+02:00</m_timeStart>
          <m_timeStart1>2013-07-01T16:59:07.0562916+02:00</m_timeStart1>
          <m_userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</m_userGuid>
        </CGenericConnection>
      </m_Connections>
      <m_ccoId>102</m_ccoId>
      <m_ccoState>free</m_ccoState>
      <m_ccoType>free</m_ccoType>
    </CGenericCCo>
  </genericCallEvent>
  <genericCallQueueEvent i:nil="true"/>
  <genericCallSupervisionEvent i:nil="true"/>
  <genericDTMFEvent i:nil="true"/>
  <genericDTMFToPlayEvent i:nil="true"/>
  <genericForwardEvent i:nil="true"/>
  <genericMediaConferenceEvent i:nil="true"/>
  <genericMediaDeviceEvent i:nil="true"/>
  <genericMediaServeFileEvent i:nil="true"/>
  <genericMediaServerEvent i:nil="true"/>
  <genericMessageEvent i:nil="true"/>
  <genericNoteEvent i:nil="true"/>
  <genericNotificationEvent i:nil="true"
xmlns:a="http://schemas.datacontract.org/2004/07/TWS_Database"/>
  <genericPresenceEvent i:nil="true"/>
  <genericRTPEvent i:nil="true"/>
  <genericRuleEvent i:nil="true"/>
  <genericServerEvent i:nil="true"/>
  <genericSoundEvent i:nil="true"/>
  <objectGuid>62c98c72-4863-4626-8078-fe65c321b22b</objectGuid>
  <objectName>4694</objectName>
  <userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</userGuid>
</CGenericEvent>
```

GenericEvent of type CallSupervisionEvent in a Cleared state

```
<CGenericEvent
xmlns="http://schemas.datacontract.org/2004/07/Algoria.TWS.GenericClass"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <creationDate>2017-05-20T14:08:51.7876495+02:00</creationDate>
  <eventType>CallSupervisionEvent</eventType>
  <genericCallEvent i:nil="true"/>
  <genericCallQueueEvent i:nil="true"/>
  <genericCallSupervisionEvent>
    <m_Users>
      <CGenericUser>
        <Addresses xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <Company i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <CreatedOn xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">0001-
01-01T00:00:00</CreatedOn>
        <Customs xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <Devices xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <DirectoryInfos xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
          <DirectoryGuid i:nil="true"/>
          <ExternalKey i:nil="true"/>
          <InternalKeys i:nil="true"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <OwnerGuid i:nil="true"/>
          <Priority>0</Priority>
          <Reverses i:nil="true"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <Searches i:nil="true"
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
        </DirectoryInfos>
        <DomainGuid
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">2047d7f2-428b-49f6-beaf-
7b2e102f954d</DomainGuid>
        <Emails xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <ExternalUserInfos
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <Firstname i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <Guid i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <IsUser
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">false</IsUser>
        <Lastname i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <MergedInfos xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
          <ContactsGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <DirectoriesGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <DirectoriesNames
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <IsDeleted>false</IsDeleted>
          <Priority>0</Priority>
        </MergedInfos>
        <PersonGuid i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <PersonalInfos xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
          <ContactsListsGuids
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
          <DefaultPhone i:nil="true"/>
          <DisplayName i:nil="true"/>
        </PersonalInfos>
        <Phones xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <Picture i:nil="true"
xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <Presences xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <UpdatedOn xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">0001-
01-01T00:00:00</UpdatedOn>
        <Urls xmlns="http://schemas.datacontract.org/2004/07/TWS_Database"/>
        <UserInfos xmlns="http://schemas.datacontract.org/2004/07/TWS_Database">
          <ActiveProfileGuid i:nil="true"/>
          <Culture i:nil="true"/>
          <Enabled>true</Enabled>
          <Init>false</Init>
          <Ip i:nil="true"/>
          <Password i:nil="true"/>
          <Type>USER</Type>
          <Username i:nil="true"/>
          <VoiceMailPassword i:nil="true"/>
        </UserInfos>
        <m_IsRedList>false</m_IsRedList>
        <m_UserDisplay>****</m_UserDisplay>
        <m_UserInfoDisplay>****</m_UserInfoDisplay>
        <m_connectionDirection>outgoing</m_connectionDirection>
        <m_publicUserNumber/>
      </CGenericUser>
      <CGenericUser i:nil="true"/>
    </m_Users>
    <m_connectionId>c165535</m_connectionId>
    <m_currentReason>Connected</m_currentReason>
    <m_currentState>cleared</m_currentState>
    <m_deviceNumber>4594</m_deviceNumber>
    <m_deviceType>PROFESSIONAL</m_deviceType>
  </genericCallSupervisionEvent>
  <genericDTMFEvent i:nil="true"/>
  <genericDTMFToPlayEvent i:nil="true"/>
  <genericForwardEvent i:nil="true"/>
  <genericMediaConferenceEvent i:nil="true"/>
  <genericMediaDeviceEvent i:nil="true"/>
  <genericMediaServeFileEvent i:nil="true"/>
  <genericMediaServerEvent i:nil="true"/>
  <genericMessageEvent i:nil="true"/>
  <genericNoteEvent i:nil="true"/>
  <genericNotificationEvent i:nil="true"
xmlns:a="http://schemas.datacontract.org/2004/07/TWS_Database"/>
  <genericPresenceEvent i:nil="true"/>
  <genericRTPEvent i:nil="true"/>
  <genericRuleEvent i:nil="true"/>
  <genericServerEvent i:nil="true"/>
  <genericSoundEvent i:nil="true"/>
  <objectGuid>62c98c72-4863-4626-8078-fe65c321b22b</objectGuid>
  <objectName>4694</objectName>
  <userGuid>97a02200-fa08-4cdd-96ab-e44eb7f577ad</userGuid>
</CGenericEvent>
```

Web service method error code

```
        fileConversionException = -2010,
        voiceMediaException = -2000,
        callDisconnectedCauseOfMediaDeviceDialog = -1001,
        callDisconnected = -1000,
        ruleActivationFailed = -501,
        invalidParameter = -251,
        nullException = -250,
        fileNotFound = -200,
        smsMessageTooLong = -152,
        smsSentFromServerFailed = -151,
        smsConnectionToServerFailed = -150,
        methodNotImplemented = -100,
        tokenNotFound = -44,
        personalContactNotFound = -41,
        contactNotFound = -40,
        sendEmailException = -30,
        actionFailed = -29,
        timeout = -28,
        actionAlreadyDone = -27,
        actionAlreadyStarted = -26,
        unauthorizedAction = -25,
        providerResponseTimeout = -24,
        webServiceException = -23,
        deviceForbidden = -22,
        cantStartMonitorDuringStarting = -21,
        twstlkLicenceLimitOverflow = -20,
        twsvisLicenceLimitOverflow = -19,
        twsrecLicenceLimitOverflow = -18,
        twsmalLicenceLimitOverflow = -17,
        twsvipLicenceLimitOverflow = -16,
        twsphpLicenceLimitOverflow = -15,
        twsaltLicenceLimitOverflow = -14,
        twscalLicenceLimitOverflow = -13,
        twssrvLicenceLimitOverflow = -12,
        licenceLimitOverflow = -11,
        voiceMailNumberNotFound = -10,
        pretelComNotFound = -9,
        telComNotFound = -8,
        applicationNotFound = -7,
        providerNotFound = -6,
        actionNotFound = -5,
        deviceNotFound = -4,
        connectionNotFound = -3,
        serverNotRunning = -2,
        serverException = -1,
        ok = 0,
        makeCallMobile = 20,
```

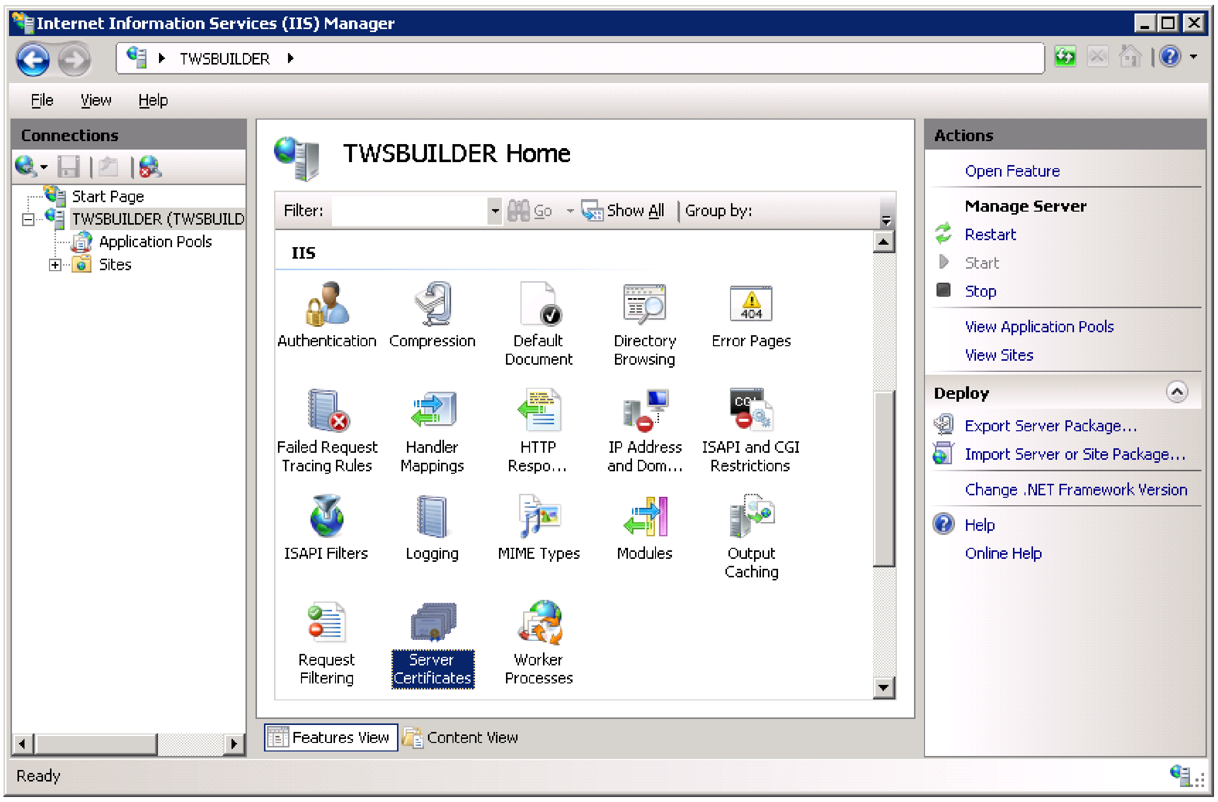
SSL configuration

Use TWS Web Services in SSL

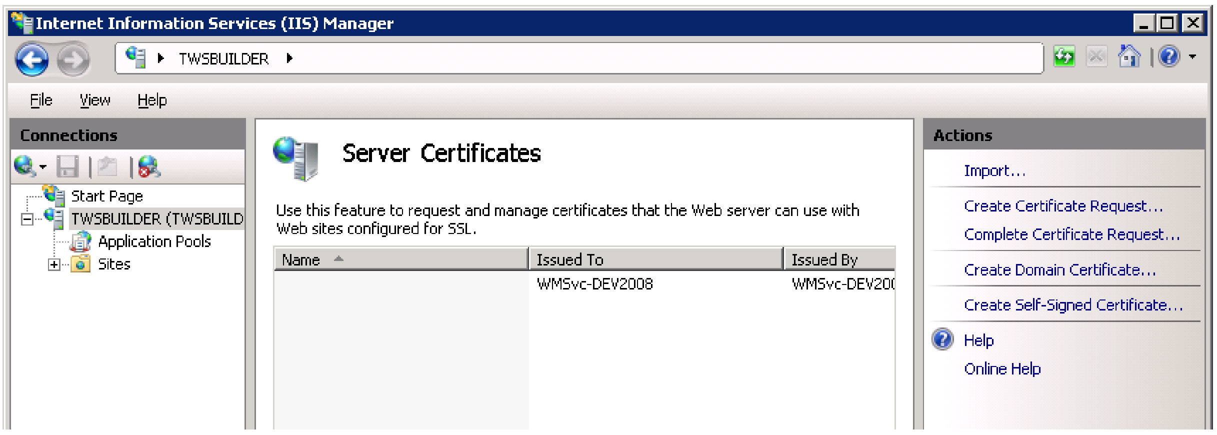
Use or Create an SSL certificate

If you already have an SSL certificate, you can skip this chapter.

Otherwise, open IIS Manager, select your server, and then double click on "*Server Certificates*".



On the right panel, click on "*Create Self-Signed Certificate…*".

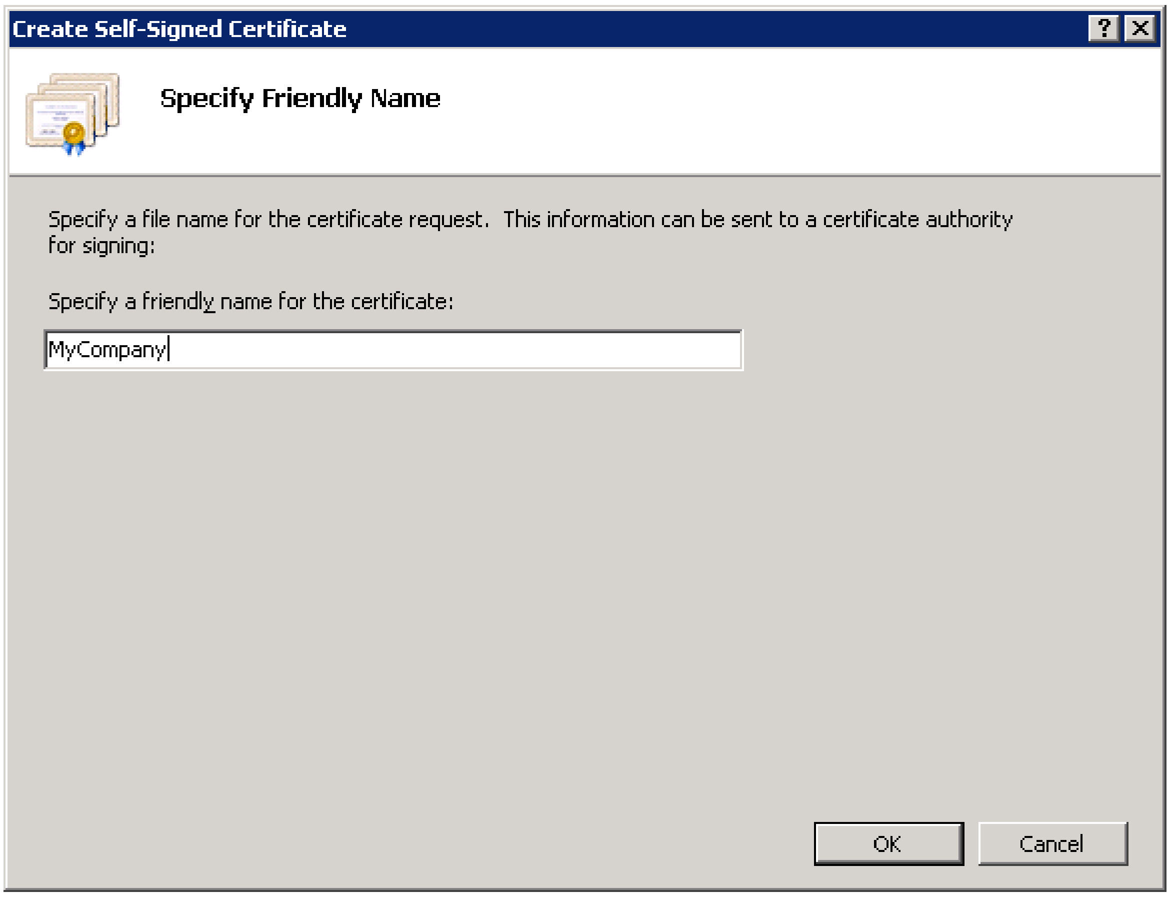


Internet Information Services (IIS) Manager
TWSBUILDER
File View Help
Connections
Start Page
TWSBUILDER (TWSBUILD
Application Pools
Sites
Server Certificates
Actions
Import...
Create Certificate Request...
Complete Certificate Request...
Create Domain Certificate...
Create Self-Signed Certificate...
Help
Online Help

Use this feature to request and manage certificates that the Web server can use with Web sites configured for SSL.

Name	Issued To	Issued By
	WMSvc-DEV2008	WMSvc-DEV20

Enter the name of your certificate and click *OK*.



Specify a file name for the certificate request. This information can be sent to a certificate authority for signing:

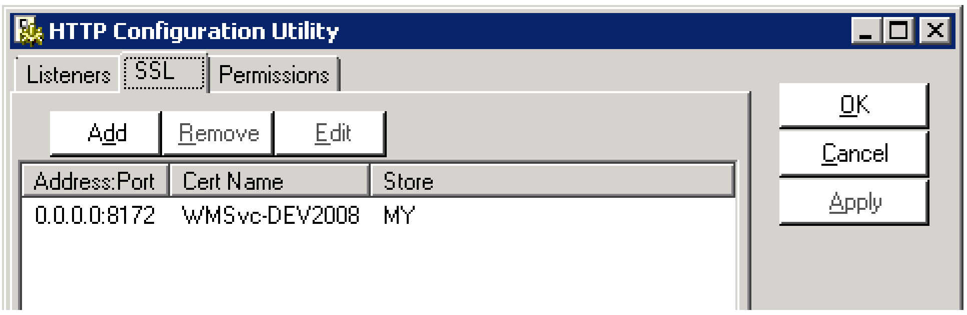
Specify a friendly name for the certificate:

MyCompany

Now you should see your newly created certificate in the list.

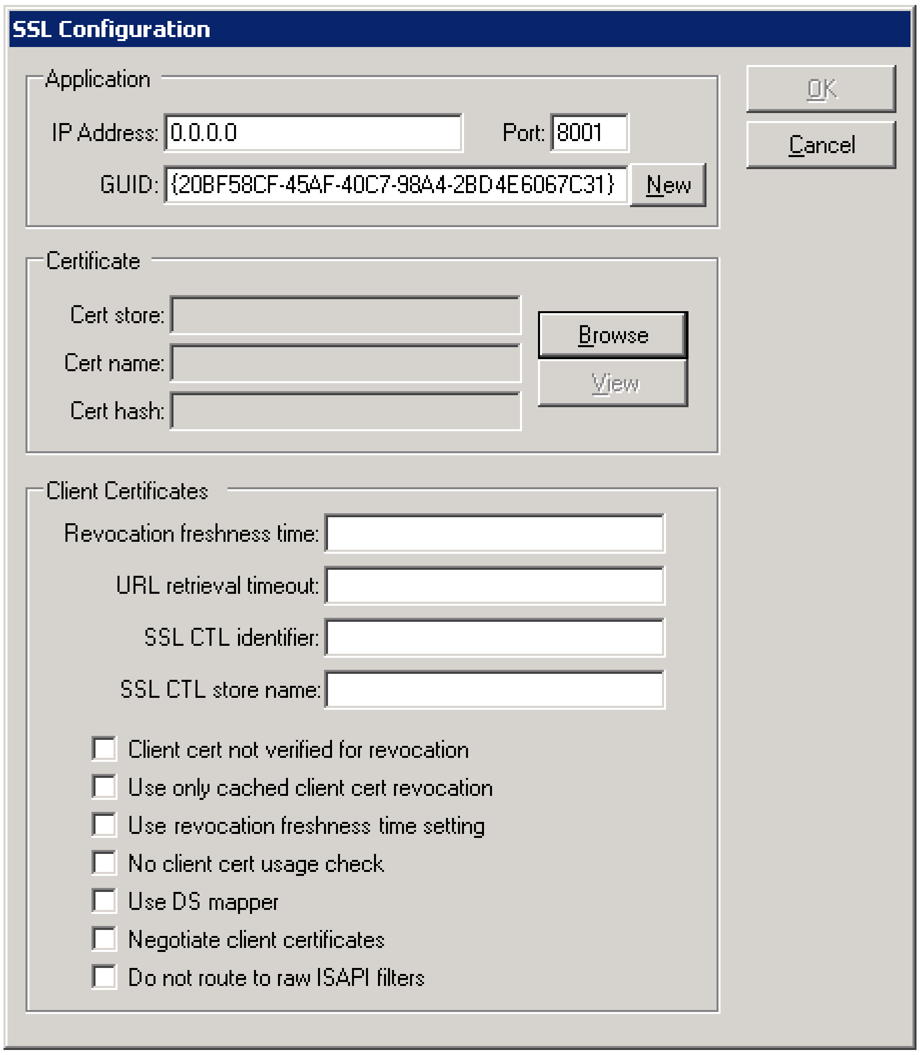
Install your SSL certificate

Open "*\TWS4\TWS_Tools\HttpConfig.exe*". Click on the "*SSL*" tab.



Address:Port	Cert Name	Store
0.0.0.0:8172	WMSvc-DEV2008	MY

Click on "*Add*".


SSL Configuration

Application

IP Address: 0.0.0.0

Port: 8001

GUID: {20BF58CF-45AF-40C7-98A4-2BD4E6067C31}

Certificate

Cert store:

Cert name:

Cert hash:

Client Certificates

Revocation freshness time:

URL retrieval timeout:

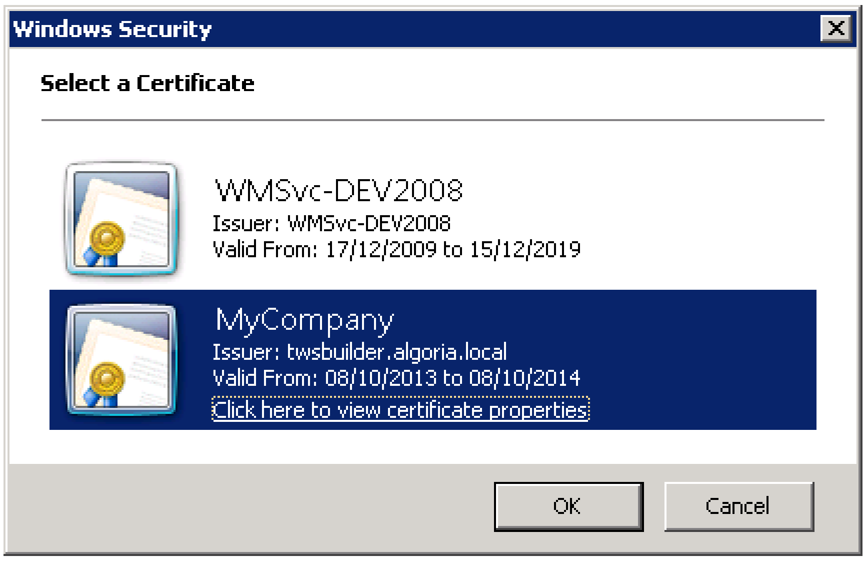
SSL CTL identifier:

SSL CTL store name:

- [ ] Client cert not verified for revocation
- [ ] Use only cached client cert revocation
- [ ] Use revocation freshness time setting
- [ ] No client cert usage check
- [ ] Use DS mapper
- [ ] Negotiate client certificates
- [ ] Do not route to raw ISAPI filters

Fill "*IP Address*" with "0.0.0.0".
Fill "*Port*" with "8001".
Fill "*GUID*" by clicking the "New" button.

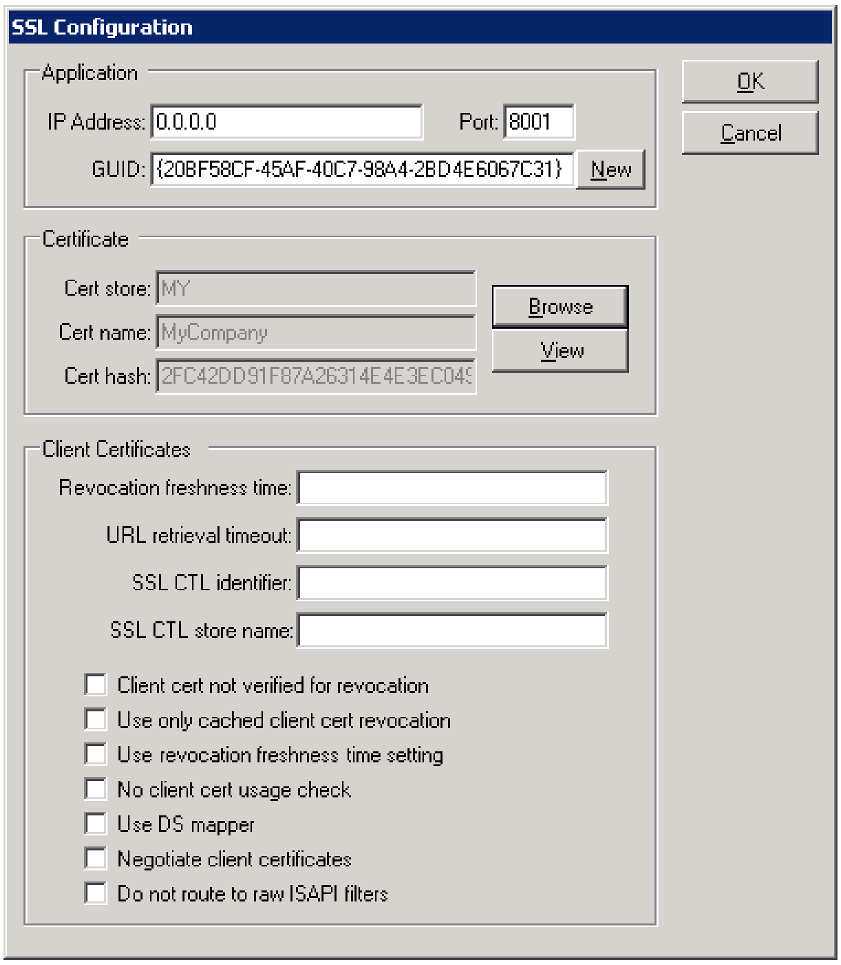
Choose your certificate by clicking "*Browse*" and select your certificate.


Windows Security

Select a Certificate

WMSvc-DEV2008
Issuer: WMSvc-DEV2008
Valid From: 17/12/2009 to 15/12/2019

MyCompany
Issuer: twsbuilder.algoria.local
Valid From: 08/10/2013 to 08/10/2014
Click here to view certificate properties


SSL Configuration

Application

IP Address: 0.0.0.0   Port: 8001

GUID: {20BF58CF-45AF-40C7-98A4-2BD4E6067C31}

Certificate

Cert store: MY

Cert name: MyCompany

Cert hash: 2FC42DD91F87A26314E4E3EC04

Client Certificates

Revocation freshness time:

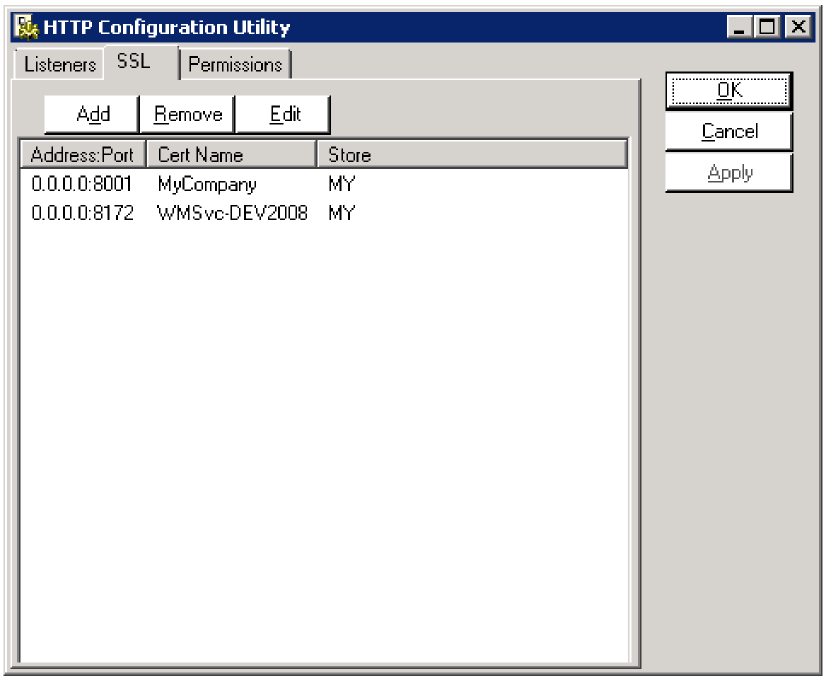
URL retrieval timeout:

SSL CTL identifier:

SSL CTL store name:

- [ ] Client cert not verified for revocation
- [ ] Use only cached client cert revocation
- [ ] Use revocation freshness time setting
- [ ] No client cert usage check
- [ ] Use DS mapper
- [ ] Negotiate client certificates
- [ ] Do not route to raw ISAPI filters

And Click “*OK*”.

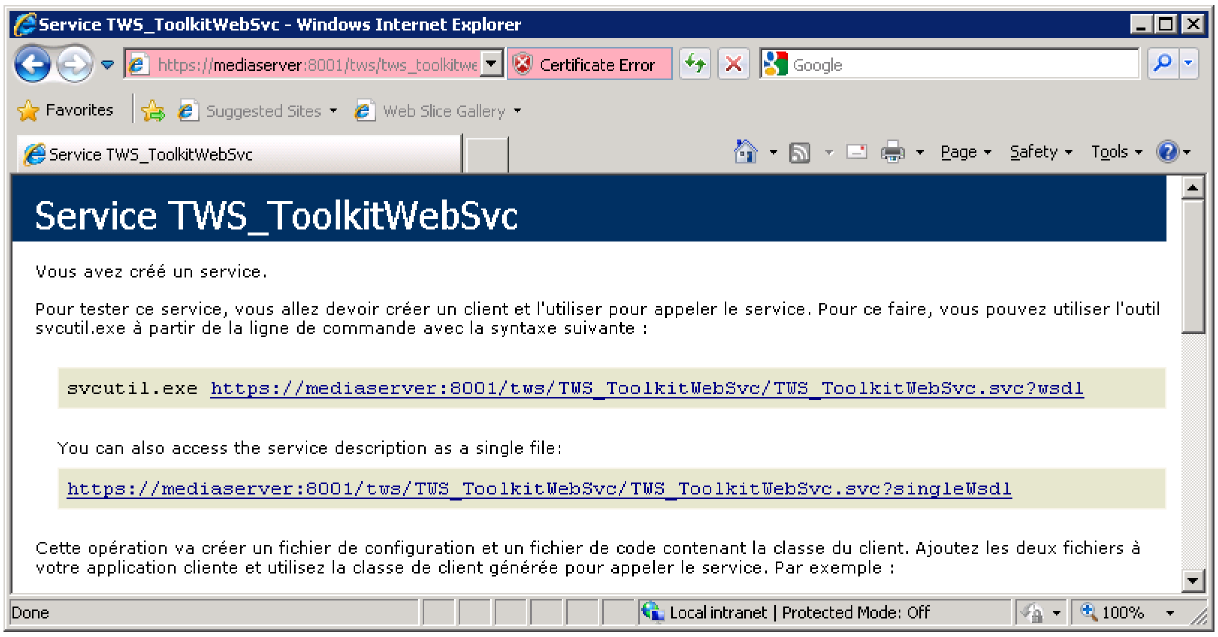

HTTP Configuration Utility

Address:Port	Cert Name	Store
0.0.0.0:8001	MyCompany	MY
0.0.0.0:8172	WMSvc-DEV2008	MY

Now Click “*Apply*”, and then “*OK*”.

Your certificate is installed on the port *8001* and ready to be used.
You can go to https://TWSserver:8001/tws/tws_toolkitwebsvc/tws_toolkitwebsvc.svc to test it. If you see a web page displaying, the configuration is OK.

If your certificate is self-signed, you should see a warning prior to the web page, click "*Continue*".


Service TWS_ToolkitWebSvc - Windows Internet Explorer
https://mediaserver:8001/tws/tws_toolkitwe
Certificate Error
Google
Favorites
Suggested Sites
Web Slice Gallery
Service TWS_ToolkitWebSvc
Page
Safety
Tools
Done
Local intranet | Protected Mode: Off
100%

Service TWS_ToolkitWebSvc

Vous avez créé un service.

Pour tester ce service, vous allez devoir créer un client et l'utiliser pour appeler le service. Pour ce faire, vous pouvez utiliser l'outil svcutil.exe à partir de la ligne de commande avec la syntaxe suivante :

```
svcutil.exe https://mediaserver:8001/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc?wsdl
```

You can also access the service description as a single file:

```
https://mediaserver:8001/tws/TWS_ToolkitWebSvc/TWS_ToolkitWebSvc.svc?singleWsdl
```

Cette opération va créer un fichier de configuration et un fichier de code contenant la classe du client. Ajoutez les deux fichiers à votre application cliente et utilisez la classe de client générée pour appeler le service. Par exemple :

Use TWS WebSockets in SSL

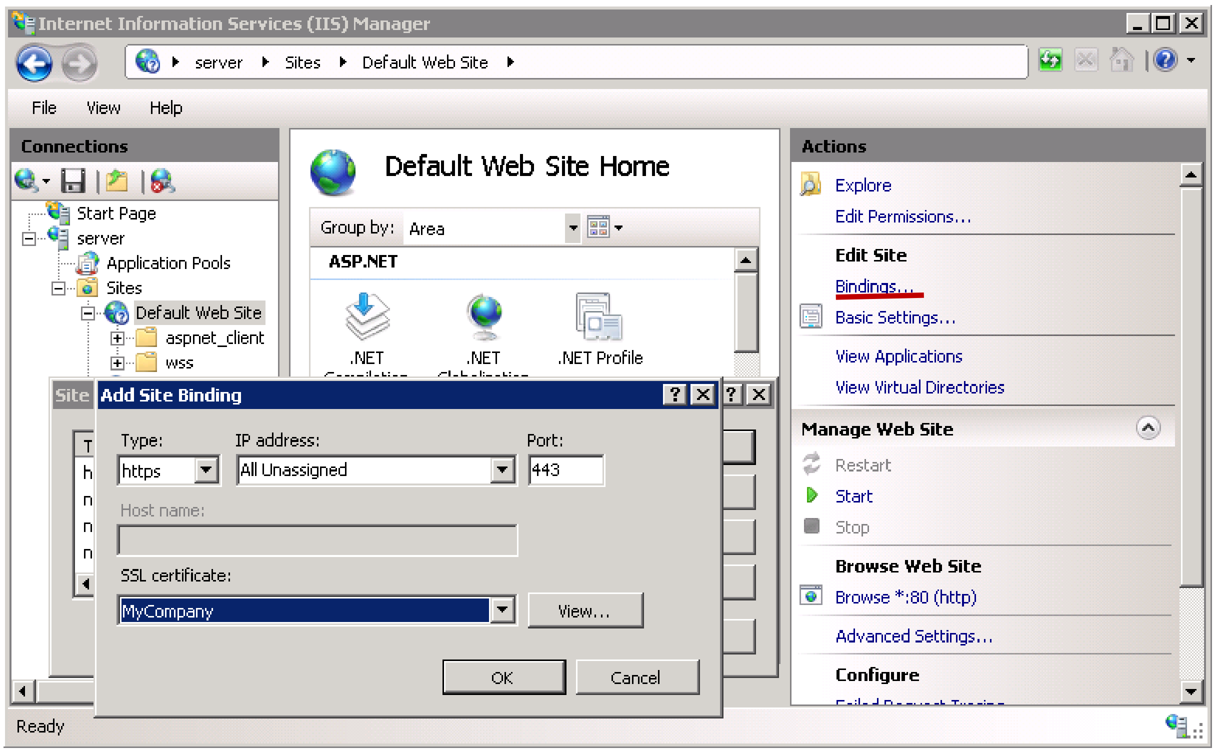
Create an SSL Certificate

To create an SSL Certificate see chapter "*Use TWS Web Services in SSL*".
If you already have your certificate, make sure this certificate will be the same that would be used on your website and for Web Sockets in SSL.

Affect the certificate to a web site in IIS

To add a certificate to your web site in IIS, select your site in the IIS Manger interface.
Click on "*Bindings*" in the Actions menu, then "*Add*" a new Site Binding.
Choose "*https*" for the type, fill the IP Address and the port as you want or leave the default values.

Choose the SSL certificate you already create and validate.



Add Site Binding

Type: https
IP address: All Unassigned
Port: 443
Host name:
SSL certificate: MyCompany
View...
OK
Cancel

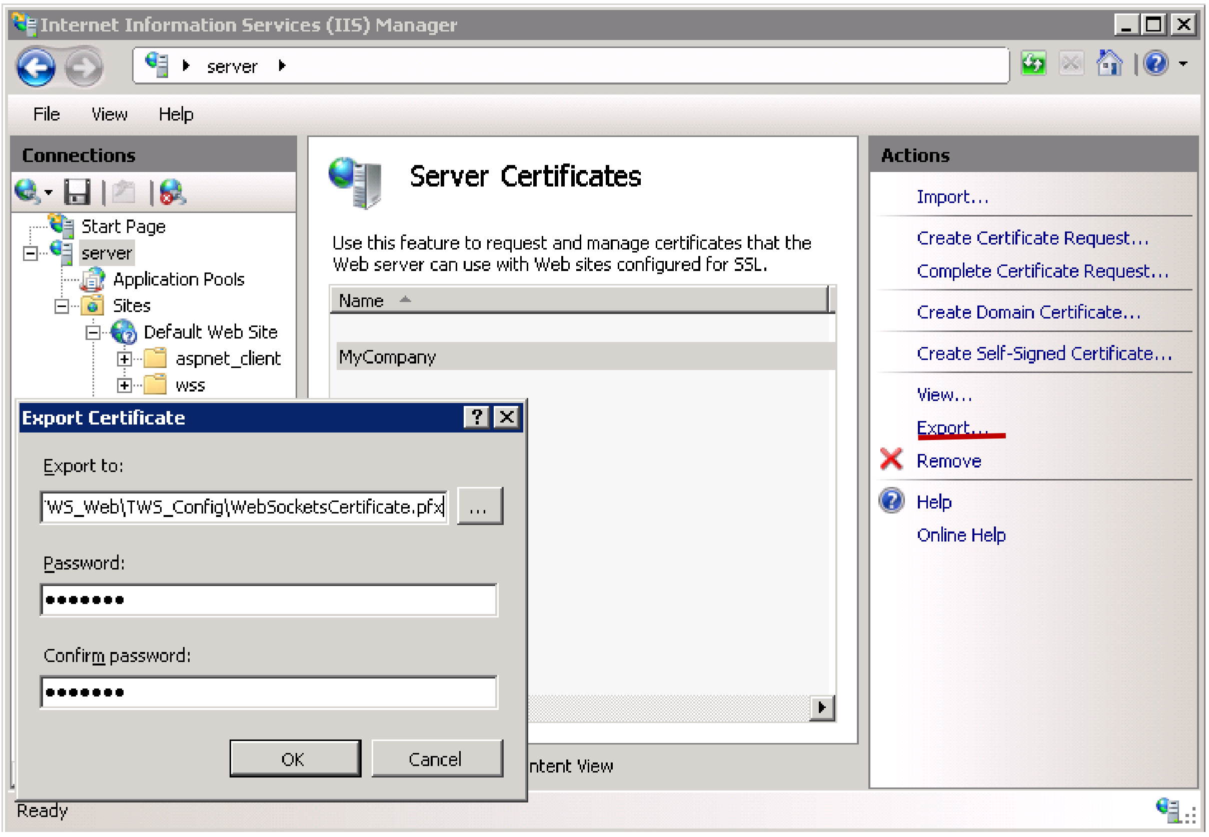
The SSL certificate is affected to your web site and you access it in https.

Use the same SSL certificate for the TWS Web Sockets

In the IIS Manager interface, select the server node and double click on the “*Server Certifcates*” feature to open it. Select your certificate and click on the Export button in the Actions menu. Fill the path to the certificate file where TWS will get the certificate. And fill the password value.

The default values that TWS will use to get the certificate are:

- for the file path: “C:\Program Files(x86)\TWS4\TWS_Web\TWS_Config\WebSocketsCertificate.pfx”
- for the password: “algoria”


Internet Information Services (IIS) Manager
server
File View Help
Connections
Start Page
server
Application Pools
Sites
Default Web Site
aspnet_client
wss
Server Certificates
Use this feature to request and manage certificates that the Web server can use with Web sites configured for SSL.
Name
MyCompany
Actions
Import...
Create Certificate Request...
Complete Certificate Request...
Create Domain Certificate...
Create Self-Signed Certificate...
View...
Export...
Remove
Help
Online Help
Export Certificate
OK
Cancel
Ready

Export to: WS_Web\TWS_Config\WebSocketsCertificate.pfx

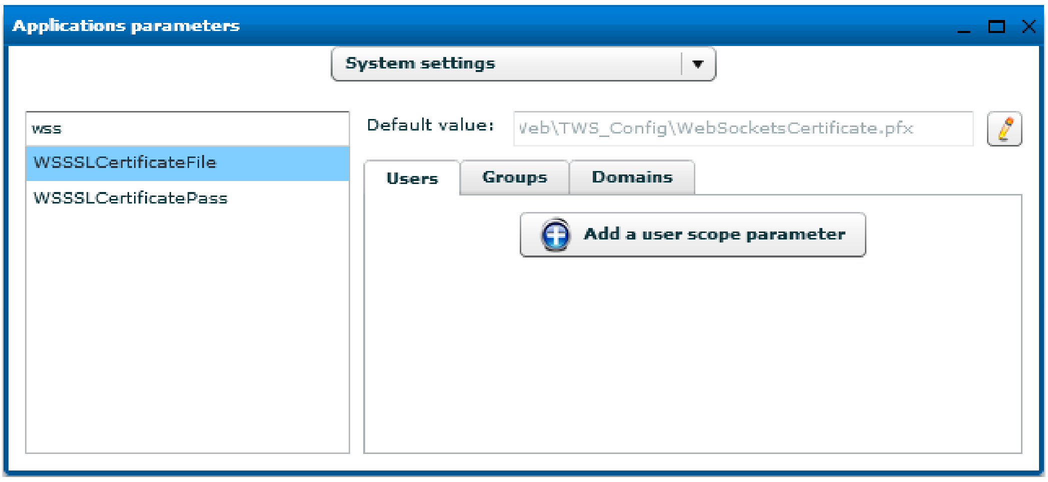
Password: •••••••

Confirm password: •••••••

If you fill different values for the export path and the password of the SSL certificate, you can modify TWS settings so that it will get the correct values. To modify the settings, go to the TWS Administration menu, click Applications then Applications Parameters. Select in the list System Settings and search “wss”.

The settings are the following:

- ***WSSSLCertificateFile***: the file path to the export SSL certificate.
- ***WSSSLCertificatePass***: the password of the export SSL certificate.



Default value: Veb\TWS_Config\WebSocketsCertificate.pfx

After the settings modified, restart the TWS4$TWS_EventServices.

080318