



Fisheries and Oceans  
Canada

Pêches et Océans  
Canada

# Electronic logbook system

## Web Service (Data transmission)

Version 3.6

June 03, 2022

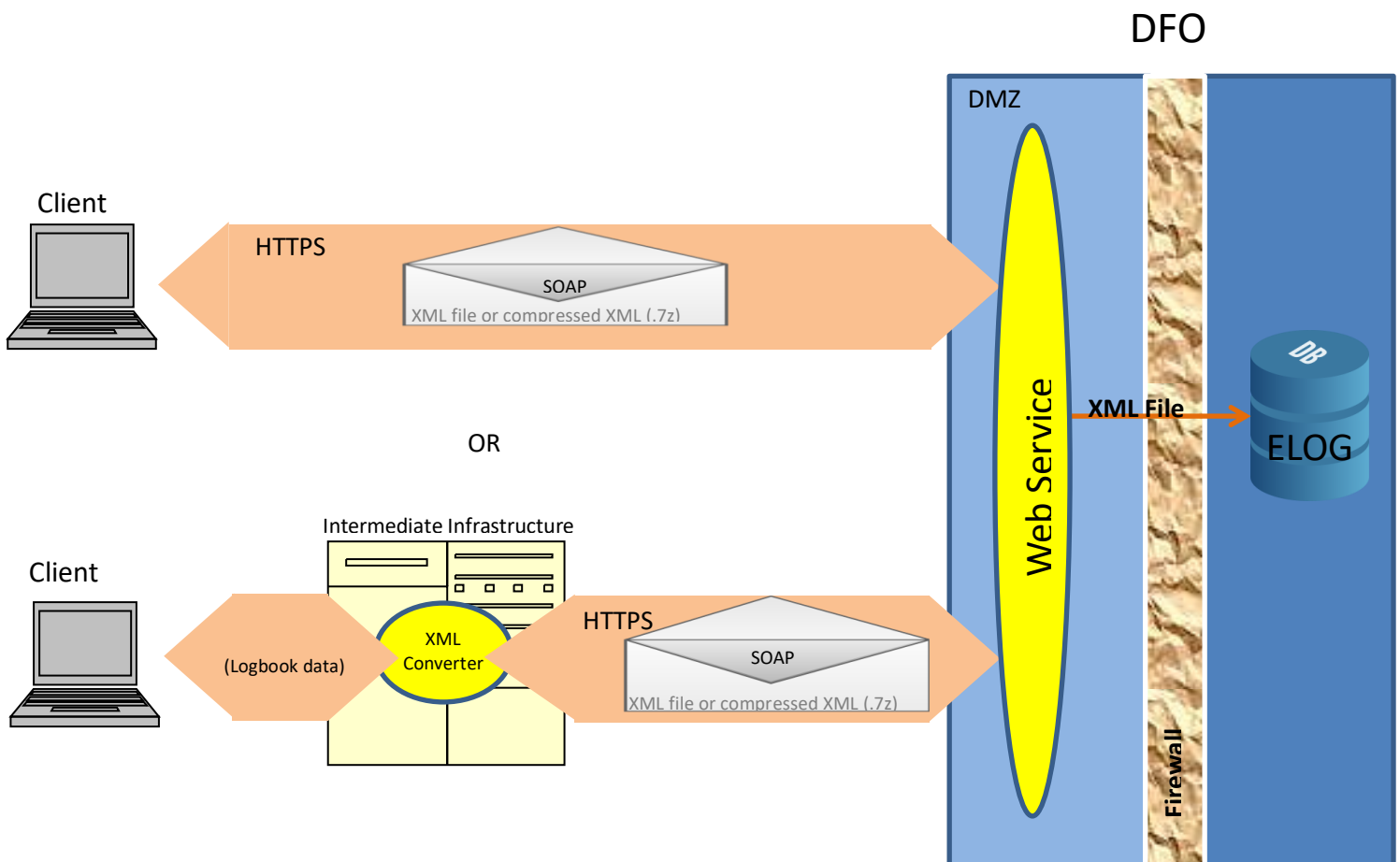
# Table Of Contents

|   |    |
|---|----|
| 1. Introduction .....                                   | 3  |
| 2. Environments .....                                   | 4  |
| 3. Web methods of the ElogXMLFileTrans Web Service..... | 4  |
| 3.1 SaveIncomingFile .....                              | 5  |
| 3.1.1 General operation.....                            | 5  |
| 3.1.2 Request.....                                      | 6  |
| 3.1.2.1 Parameters.....                                 | 6  |
| 3.1.2.2 .NET Examples.....                              | 6  |
| 3.1.2.2.1 Manually written SOAP envelope example:.....  | 6  |
| 3.1.2.2.2 Proxy object examples .....                   | 7  |
| 3.1.3 Response .....                                    | 9  |
| 3.1.3.1 No error detected.....                          | 9  |
| 3.1.3.2 Error detected .....                            | 11 |
| 3.2 ValidateElogKey .....                               | 12 |
| 3.2.1 General operation.....                            | 12 |
| 3.2.2 Request.....                                      | 12 |
| 3.2.2.1 Parameter .....                                 | 12 |
| 3.2.2.2 .NET Examples.....                              | 13 |
| 3.2.2.2.1 Manually written SOAP envelope example:.....  | 13 |
| 3.2.2.2.2 Proxy object examples .....                   | 13 |
| 3.2.3 Response .....                                    | 15 |
| 3.2.3.1 No error detected.....                          | 15 |
| 3.2.3.2 Error detected .....                            | 15 |
| 4. Error Codes and Messages.....                        | 16 |

## 1. Introduction

As part of the Electronic Logbook System (ELOG), fishermen will be required to submit their logbook data to the Department of Fisheries and Oceans (DFO) in the form of electronic data.

DFO will only accept data received through the provided web service. That said, different intermediate infrastructures and technologies can be used when transmitting fisherman data, but ultimately DFO will only accept data transiting through its web service.



The purpose of this document is to describe the DFO ElogXMLFileTransfer web service. This web service is used to receive electronic logbook data transmitted by the fisherman using the SaveIncomingFile method.

Modifications made to this document since the previous version are indicated in red in the document.

## 2. Environments

Two environments are available:

### **Test environment:**

This environment will be used by software developers to test the sending of data to DFO during the development of their solution.

URL: <https://inter-w01-uat.dfo-mpo.gc.ca/ws/ElogXMLFileTransfer/ElogXMLFileTransfer.asmx>

### **Production environment:**

This environment will be used by fishers to officially transmit their data to DFO. **This environment must under no circumstances be used for testing.**

URL: <https://inter-w01.dfo-mpo.gc.ca/ws/ElogXMLFileTransfer/ElogXMLFileTransfer.asmx>

The DFO web service WSDL is always **https** (HyperText Transfer Protocol Secure). As the name suggests, the s in an https request ensures that data remain private and integral.

## 3. Web methods of the ElogXMLFileTrans Web Service

The ElogXMLFileTransfer web service includes the following web methods:

|                     |   |
|---------------------|---|
| SaveIncomingFile :  | This method is used to transmit electronic logbook data to DFO.                     |
| ValidateElogKey:    | This method allows you to check the validity of an ELOG key.                        |
| CreateLogString :   | Reserved for the exclusive use of DFO. This method is not documented in this guide. |
| DeleteLog :         | Reserved for the exclusive use of DFO. This method is not documented in this guide. |
| ListLogs :          | Reserved for the exclusive use of DFO. This method is not documented in this guide. |
| SetMx:              | Reserved for the exclusive use of DFO. This method is not documented in this guide. |
| TestWsAvailability: | Reserved for the exclusive use of DFO. This method is not documented in this guide. |
| Write_master_file:  | Reserved for the exclusive use of DFO. This method is not documented in this guide. |

\*\* This guide will not describe the web methods reserved for the exclusive use of DFO.

## 3.1 SaveIncomingFile

### 3.1.1 General operation

When receiving the data, this web method will perform the following operations:

- ❖ Convert<sup>1</sup> the ELOG key \*
- ❖ Validate the ELOG key. \*
- ❖ Determine the confirmation number \*
- ❖ Register the transmission attempt in the load register \*
- ❖ Convert the XML file name \*
- ❖ Validate of the XML file name \*
- ❖ If applicable, decompress the logbook data (.7z) \*
- ❖ Convert the logbook data \*
- ❖ If the data is in XML format:
  - Read and validate the following XML element:
    - **GENERAL INFO.CIE\_ID** \*
    - **GENERAL INFO.REG\_ID** \*
    - **GENERAL INFO.FIN** \*
    - **GENERAL INFO.VRN** \*
    - **GENERAL \_INFO.FORM\_VER\_ID** \*
    - **GENERAL INFO.SOFT\_VER** \*
    - **TRIP.LGBK\_UID** \*
    - **REPORT.REPORT\_UID** \*
    - **TRIP.START\_DT** \*
    - **EFFORT.LIC\_NO** \*
  - Check if the XML file has already been transmitted\*
- ❖ If no errors were detected, send an XML response containing the following:
  - Confirmation number of reception **<CONF>**
  - Error Code **<ERR>** - Contains WS0000 when no error has been detected
  - Vessel Registration Number **<VRN>**
  - Fisher Identification Number **<FIN>**
  - Logbook unique identifier **<LGBK\_UID>**
    - **If the form is 233 Inactivity Reports, then you will receive <REPORT\_UID> instead**
- ❖ If an error has been detected, send a response containing the following:
  - Confirmation number of reception **<CONF>**
  - Error code **<ERR>**

\* As soon as an error is detected, the loading process is interrupted and a response containing the error code is returned.

---

<sup>1</sup> Convert a base-64 digits to an equivalent 8-bit unsigned integer array and then decodes all the bytes in the specified byte array into a string representing the actual data.

In the event of a web service crash, an email will be automatically sent to the national administrator of the Electronic Logbook system.

### 3.1.2 Request

#### 3.1.2.1 Parameters

This web method contains the following 3 parameters:

- p\_eologkey:** A string representing the fish harvester's ELOG key encoded in base-64.
- p\_filename:** A string, representing the name of the file containing the log data, encoded in base-64. If the file is compressed, the file name extension will be .7Z instead of .XML before its encoded in base-64
- p\_body:** A string encoded in base-64 representing:
- o The XML file containing the log data when the file name extension entered in p\_filename is .XML
- OR
- The **compressed** XML file containing the log data when the file name extension entered in p\_filename is .7Z

#### 3.1.2.2 .NET Examples

Data transmission is done via SOAP (Simple Object Access Protocol).

The following .NET examples show how SOAP transmission can be accomplished in two ways: by writing a SOAP envelope manually or by using a proxy object.

##### 3.1.2.2.1 Manually written SOAP envelope example:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SaveIncomingFile xmlns="http://tempuri.org/">
      <p_eologkey>Elogkey_b64</p_eologkey>
      <p_filename>Filename_b64</p_filename>
      <p_body>XMLData_b64</p_body>
    </SaveIncomingFile>
  </soap:Body>
</soap:Envelope>
```

Access the web service with the **HttpWebRequest** object.

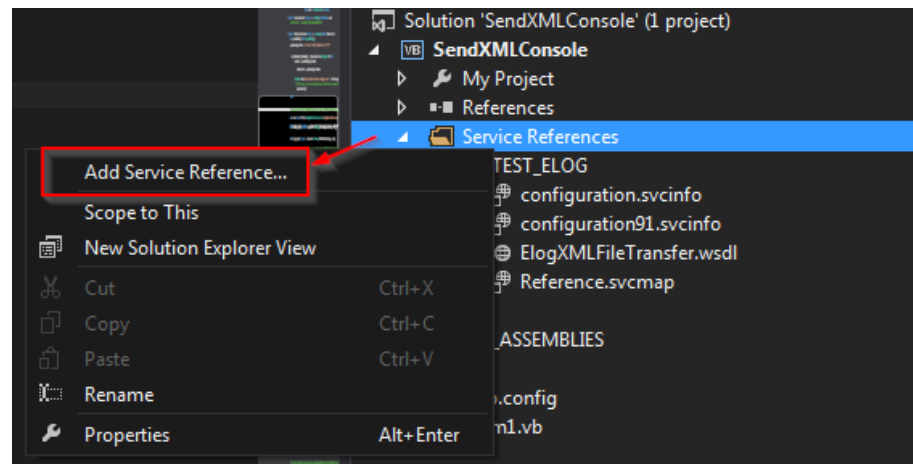
```
Dim myReq As HttpWebRequest = WebRequest.Create(param_url)
```

The value of the param\_url parameter is the WSDL or the URL of the Web Service.

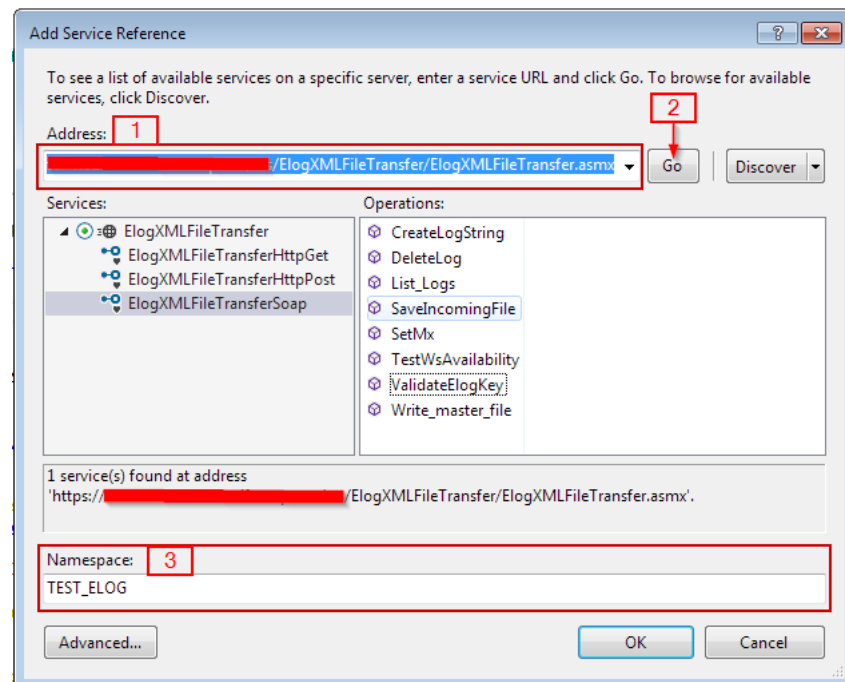
After setting the HttpWebRequest object's built-in parameters, the soap envelope is then attached and sent to DFO with the object's built-in method and functions. See : [How to: Send data by using the WebRequest class](#)

### 3.1.2.3.2 Proxy object examples

Add the ELOG web service reference with Add Service Reference:



1. Add the URL or WSDL address of the web service
2. Press Go to download the functions
3. Name the instance of your Subject: **TEST\_ELOG**



4. In the source code, create an instance of the web service object named **client**:

```
Dim client As New  
TEST_ELOG.ElogXMLFileTransferSoapClient("ElogXMLFileTransferSoap")
```

5. Encode the 3 parameters in base 64 format. For example:

```
Dim Elogkey_string As String = "ELOG_KEY_GOES_HERE"  
Dim Elogkey_byte() As Byte = System.Text.Encoding.UTF8.GetBytes(Elogkey_string)  
Dim Elogkey_b64 As String = Convert.ToBase64String(Elogkey_byte)
```

6. Invoke the **SaveIncomingFile** method of the **client** object in a Try / Catch that catches exceptions of type TimeoutException, FaultException, and CommunicationException, among other things.

```
Dim response As String = client.SaveIncomingFile(Elogkey_b64,  
Filename_b64, cBody64)
```

7. From the alphanumeric response beginning with the letters "WS", the 3rd party application will proceed from here with its own built-in validation processes.



### 3.1.3 Response

The SaveIncomingFile web method's response is in XML format.

The data transmission to DFO is considered successful only if the <ERR> element contained in the XML response is equal to "**WS0000**". In all other cases, the transmission of the data should be considered a failure and should be sent later if necessary.

#### 3.1.3.1 No error detected

If no errors were detected during the reception process, the returned XML file will consist of the following:

|                   |  |
|-------------------|--|
| <b>XML header</b> | <?xml version="1.0" encoding="UTF-8"?> |
|-------------------|--|

| <b>Beginning tag</b> | <b>End tag</b> | <b>Content</b>  |
|----------------------|----------------|---|
| <WS_RESP>            | </WS_RESP>     | Top most XML node containing the web method's response.   |
| <CONF>               | </CONF>        | Contains the transmission's confirmation number. This number confirms that the XML file has been transmitted to the Web Service.<br>e.g. 191  |
| <ERR>                | </ERR>         | Will always contain WS0000 for successfully validated XML files.  |
| <VRN>                | </VRN>         | Vessel Registration Number<br>e.g. 98882  |
| <FIN>                | </FIN>         | Fisherman Identification Number<br>e.g. 411023657   |
| <LGBK_UID>           | </LGBK_UID>    | Unique identifier for the logbook<br>e.g. ATCKWL<br>Warning: If the XML file contains more than one log, there will be as many instances of this element as logbooks.   |
| <REPORT_UID>         | </REPORT_UID>  | Unique identifier for the inactivity report<br>e.g. AAABBB<br>This node is returned only in the case form 233, inactivity report, is used.<br>Warning: If the XML file contains more than one inactivity report, there will be as many instances of this element as there are inactivity reports. |

**Sample XML response from a successfully processed file.**

```
<?xml version="1.0" encoding="UTF-8"?>
<WS_RESP>
  <CONF>191</CONF>
  <ERR>WS0000</ERR>
  <VRN>98882</VRN>
  <FIN>411023657</FIN>
  <LGBK_UID>ATCKWL</LGBK_UID>
  <LGBK_UID>FWEYUL</LGBK_UID>
</WS_RESP>
```

**Sample XML response from a successfully processed Inactivity file.**

```
<?xml version="1.0" encoding="UTF-8"?>
<WS_RESP>
  <CONF>191</CONF>
  <ERR>WS0000</ERR>
  <VRN>98882</VRN>
  <FIN>411023657</FIN>
  <REPORT_UID>AAABBB</REPORT_UID>
  <REPORT_UID>CCDDDD</REPORT_UID>
</WS_RESP>
```

### 3.1.3.2 Error detected

If an error is detected during the reception process, the returned XML file will consist of the following:

|                   |   |
|-------------------|---|
| <b>XML header</b> | <code>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</code> |
|-------------------|---|

| <b>Beginning tag</b>         | <b>End tag</b>                | <b>Contents</b>  |
|------------------------------|-------------------------------|--|
| <code>&lt;WS_RESP&gt;</code> | <code>&lt;/WS_RESP&gt;</code> | Top most XML node containing the web method's response.  |
| <code>&lt;CONF&gt;</code>    | <code>&lt;/CONF&gt;</code>    | Contains the transmission's confirmation number. This number confirms that the XML file has been transmitted to the Web Service.<br>e.g.: 191<br>The confirmation number returned may be empty if the DFO database is experiencing difficulties or is not available. The confirmation number is provided to the Web service from the database. If the database is not available, the transmission will be rejected and the error code WS1031 will be returned. |
| <code>&lt;ERR&gt;</code>     | <code>&lt;ERR&gt;</code>      | Contains the error code corresponding to the error detected during the data receiving process<br>e.g. WS1033   |

#### Sample XML response from a unsuccessfully processed file:

```
<?xml version="1.0" encoding="UTF-8"?>
<WS_RESP>
  <CONF>191</CONF>
  <ERR>WS1033</ERR>
</WS_RESP>
```

Note: If no XML file containing the response has been returned or the confirmation number (<CONF>) is null or equal to 0, the transmission is considered "failed" and must be retried later if necessary. A CONF number means that the transmission has been successful. This does not mean that the client has sent a valid XML file, only that the transmission has been done.

## 3.2 ValidateElogKey

The Web method ValidateElogKey allow the verification of the validity of the ELOG key passed as a parameter. This method allows the users to confirm that the ELOG key they have entered in their client application is valid and will allow them to successfully transmit their logbooks when the time comes.

Client applications should allow this test to be performed each time the ELOG key is modified in the client application but also before each fishing season. The ValidateElogKey web method is a read only web method as opposed to SaveIncomingFile which saves actual XML files.

### 3.2.1 General operation

When receiving the ELOG\_KEY in parameter, the ValidateElogKey web method will perform the following operations:

- ❖ Converting<sup>1</sup> the ELOG key \*
- ❖ Validation of the ELOG key. \*

\* As soon as an error is detected, the validation process is interrupted and an XML response containing the error code is returned. Besides being expired or unregistered, an error code can be receive for the following reasons:

- the key was not given, the parameter was empty.
- the client application did not convert the key in a valid base 64 string
- the ELOG key as less than 24 characters
- the key was not entered or converted to uppercase before base 64 conversion. Converting a lowercase string value to base 64 renders different values than a uppercase string value.

### 3.2.2 Request

#### 3.2.2.1 Parameter

This web method contain the following parameter:

**p\_elogkey:** String representing fisher's ELOG key, encoded in base 64.

---

<sup>1</sup> Converting: convert a base-64 digits to an equivalent 8-bit unsigned integer array and then decodes all the bytes in the specified byte array into a string representing the actual data.

### 3.2.2.2 .NET Examples

Data transmission is done via SOAP (Simple Object Access Protocol).

The following .NET examples show how SOAP transmission can be accomplished in two ways: by writing a SOAP envelope manually or by using a proxy object.

#### 3.2.2.2.1 Manually written SOAP envelope example:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ValidateElogKey xmlns="http://tempuri.org/">
      <p_elogkey>Elogkey_b64</p_elogkey>
    </ValidateElogKey>
  </soap:Body>
</soap:Envelope>
```

Access the web service with the **HttpWebRequest** object.

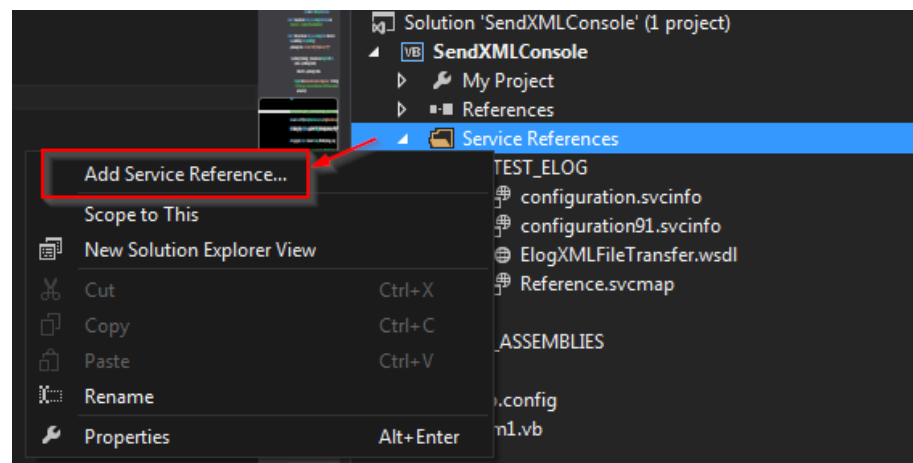
```
Dim myReq As HttpWebRequest = WebRequest.Create(param_url)
```

The value of the param\_url parameter is the WSDL, the URL of the Web Service.

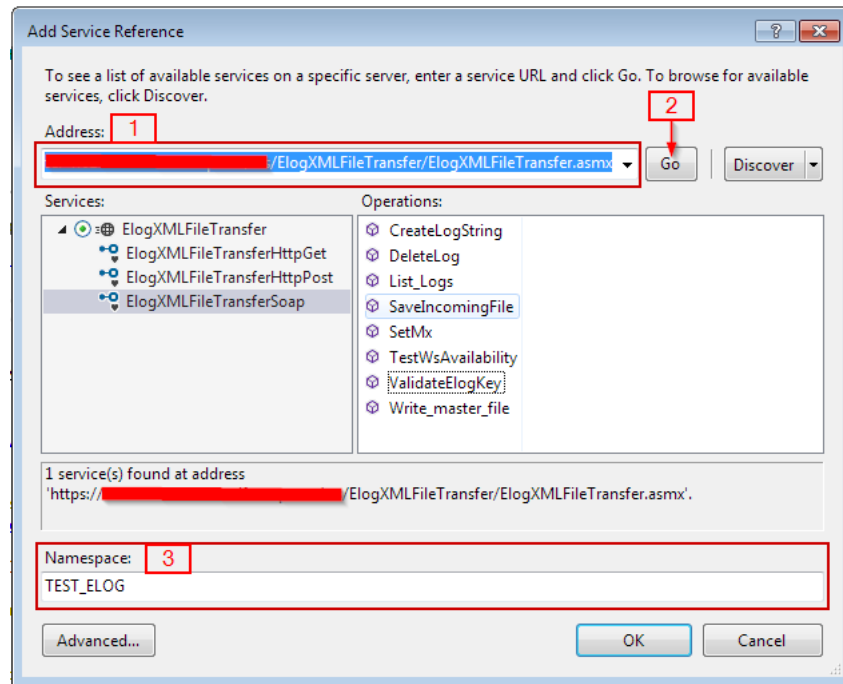
After setting the HttpWebRequest object's built-in parameters, the soap envelope byte array is then attached and sent to DFO with its built-in method and functions. See : [How to: Send data by using the WebRequest class](#)

#### 3.2.2.2.2 Proxy object examples

Add the ELOG web service reference with Add Service Reference:



1. Add the **URL** or **WSDL** address of the web service
2. Press Go to download the functions
3. Name the instance of your Subject: **TEST\_ELOG**



4. In the source code, create an instance of the web service object named **client**:

Dim client As New

TEST\_ELOG.ElogXMLFileTransferSoapClient("ElogXMLFileTransferSoap")

5. Encode the parameter in base 64 format. For example:

Dim Elogkey\_string As String = "ELOG\_KEY\_GOES\_HERE"

Dim Elogkey\_byte() As Byte = System.Text.Encoding.UTF8.GetBytes(Elogkey\_string)

Dim Elogkey\_b64 As String = Convert.ToBase64String(Elogkey\_byte)

6. Invoke the ValidateElogKey method of the client object in a Try / Catch that catches exceptions of type TimeoutException, FaultException, and CommunicationException, among other things.

Dim response As String = client.ValidateElogKey(Elogkey\_b64)

7. From the alphanumeric response beginning with the letters "WS", the 3rd party application will proceed from here with its own built-in validation processes.

### 3.2.3 Response

The ValidateElogKey web method's response is in XML format.

The ELOG key is considered valid only if the <ERR> element contained in the XML response is equal to "WS1000". In all other cases, validation should be considered a failure and should be tried again later if necessary.

#### 3.2.3.1 No error detected

If no errors were detected during the validation process, the returned XML file will consist of the following:

|                   |  |
|-------------------|--|
| <b>XML header</b> | <?xml version="1.0" encoding="UTF-8"?> |
|-------------------|--|

| Beginning tag | End tag    | Content  |
|---------------|------------|--|
| <WS_RESP>     | </WS_RESP> | Top most XML node containing the web method's response.                  |
| <ERR>         | </ERR>     | Will always contain <b>WS1000</b> for a successfully validated ELOG key. |

#### Sample XML response from a successfully validated key.

```
<?xml version="1.0" encoding="UTF-8"?>
<WS_RESP>
  <ERR>WS1000</ERR>
</WS_RESP>
```

#### 3.2.3.2 Error detected

If an error was detected during the validation process, the returned XML file will consist of the following:

|                   |  |
|-------------------|--|
| <b>XML header</b> | <?xml version="1.0" encoding="UTF-8"?> |
|-------------------|--|

| Beginning tag | End tag    | Contents   |
|---------------|------------|--|
| <WS_RESP>     | </WS_RESP> | Top most XML node containing the web method's response.  |
| <ERR>         | <ERR>      | Contains the error code corresponding to the error detected during the ELOG key validation process e.g. WS1003 |

#### Sample XML response from a unsuccessfully validated key:

```
<?xml version="1.0" encoding="UTF-8"?>
<WS_RESP>
  <ERR>WS1003</ERR>
</WS_RESP>
```

## 4. Error Codes and Messages

| Error code    | Description  | SaveIncomingFile | ValidateElogKey |
|---------------|--|------------------|-----------------|
| <b>WS0000</b> | File successfully transferred  | x                |                 |
| <b>WS1000</b> | Valid ELOG key   |                  | x               |
| <b>WS1001</b> | ELOG key not provided in the SOAP statement  | x                | x               |
| <b>WS1002</b> | File is not a valid XML file   | x                |                 |
| <b>WS1003</b> | ELOG key not found in the DFO database   | x                | x               |
| <b>WS1004</b> | Filename not specified in the SOAP statement   | x                |                 |
| <b>WS1005</b> | Invalid filename format in the SOAP statement  | x                |                 |
| <b>WS1006</b> | The maximum number of daily transmissions has been reached.<br>Wait until tomorrow to send your logbook or contact DFO to request an increase of your daily transmissions limit. | x                |                 |
| <b>WS1010</b> | Application developer identifier has not been specified in the data file (GENERAL_INFO.CIE_ID)   | x                |                 |
| <b>WS1011</b> | Application developer identifier does not exist in the DFO database (GENERAL_INFO.CIE_ID)  | x                |                 |
| <b>WS1012</b> | Application developer identifier is not numeric or is not the right length (GENERAL_INFO.CIE_ID)   | x                |                 |
| <b>WS1013</b> | DFO administrative region has not been specified in data file (GENERAL_INFO.REG_ID)  | x                |                 |
| <b>WS1014</b> | DFO administrative region identifier does not exist in the DFO database (GENERAL_INFO.REG_ID)  | x                |                 |
| <b>WS1015</b> | DFO administrative region identifier is not numeric or is not the right length (GENERAL_INFO.REG_ID)   | x                |                 |
| <b>WS1016</b> | Fisher's identification number has not been specified in the data file (GENERAL_INFO.FIN)  | x                |                 |
| <b>WS1017</b> | Fisher's identification number does not exist in the DFO database (GENERAL_INFO.FIN)   | x                |                 |
| <b>WS1018</b> | Fisher's identification number is not numeric or is not the right length (GENERAL_INFO.FIN)  | x                |                 |
| <b>WS1019</b> | Vessel registration number does not exist in the DFO database (GENERAL_INFO.VRN)   | x                |                 |
| <b>WS1020</b> | Vessel registration number is not numeric or is not the right length (GENERAL_INFO.VRN)  | x                |                 |
| <b>WS1021</b> | Form version identifier has not been specified in data file (GENERAL_INFO.FORM_VER_ID)   | x                |                 |
| <b>WS1022</b> | Form version identifier does not exist in DFO database (GENERAL_INFO.FORM_VER_ID) or is not valid against dates  | x                |                 |
| <b>WS1023</b> | Form version identifier is not numeric or is not the right length (GENERAL_INFO.FORM_VER_ID)   | x                |                 |
| <b>WS1024</b> | Client application version has not been specified in the data file (GENERAL_INFO.SOFT_VER)   | x                |                 |



| Error code    | Description  | SaveIncomingFile | ValidateElogKey |
|---------------|--|------------------|-----------------|
| <b>WS1025</b> | Client application version exceed the maximum length allowed (GENERAL_INFO.SOFT_VER)     | x                |                 |
| <b>WS1026</b> | XML data file does not contain any data  | x                |                 |
| <b>WS1031</b> | DFO Web service is out of order, please retry later                                      | x                | x               |
| <b>WS1033</b> | A problem occurred while deflating the compressed data file (.7z)                        | x                |                 |
| <b>WS1034</b> | The same file name has already been received by DFO                                      | x                |                 |
| <b>WS1035</b> | A problem has occurred while decoding the 64 bits string of the filename                 | x                |                 |
| <b>WS1036</b> | A problem has occurred while decoding the 64 bits string of the ELOG key                 | x                | x               |
| <b>WS1037</b> | A problem has occurred while decoding the 64 bits string of the body                     | x                |                 |
| <b>WS1038</b> | XML content is not valid against XSD   | x                |                 |
| <b>WS1039</b> | Cannot find an XSD for this form version   | x                |                 |
| <b>WS1040</b> | This version of the client application is not "qualified"                                | x                |                 |
| <b>WS1041</b> | The logbook unique identifier has not been specified in the data file (TRIP.LGBK_UID)    | x                |                 |
| <b>WS1042</b> | The logbook unique identifier must be made of six uppercase letters (TRIP.LGBK_UID)      | x                |                 |
| <b>WS1043</b> | The report unique identifier has not been specified in the data file (REPORT.REPORT_UID) | x                |                 |
| <b>WS1044</b> | The report unique identifier must be made of six uppercase letters (REPORT.REPORT_UID)   | x                |                 |
| <b>WS1045</b> | Parameter "p_filename" of the SOAP statement is empty. Filename is missing.              | x                |                 |
| <b>WS1046</b> | Parameter "p_body" of the SOAP statement is empty. XML file is missing                   | x                |                 |