

*How to consume a Domino Web
Services from Visual Studio under
Security*

Summary

Authors.....	2
Abstract	2
Web Services.....	3
Write a Visual Basic Consumer	5

Authors

Andrea Fontana
IBM Champion for WebSphere
DeveloperWorks Contributor Author
Can be contact at: a.fontana@net2action.com

Abstract

In this article I want explain how we can develop a Web Services Consumer, using Microsoft Visual Studio Environment, and securing it using iCredential authentication mechanism.

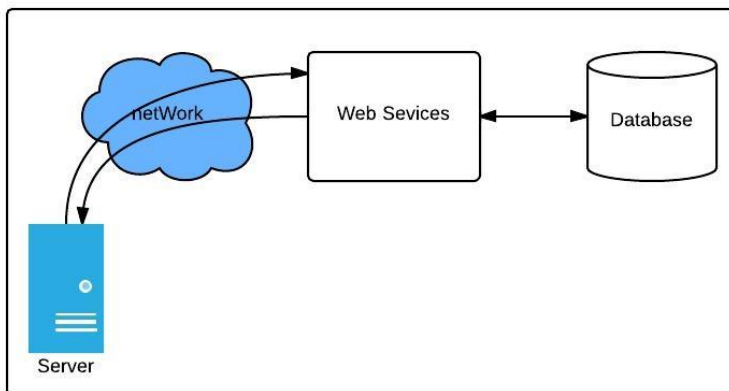
Web Services

A Web service is an archive of remote operations that can be called by sending messages over the Internet. A Web service provider publishes a Web service for query and use, and a Web service consumer calls operations from the service. A Web service provider makes available a WSDL (Web Services Description Language) document that defines the service interface. The WSDL document is in XML format. What happens behind the interface is up to the provider, but most providers map the interface to procedure calls in a supported programming language. Incoming requests from a consumer are passed through to the underlying code, and results are passed back to the consumer.

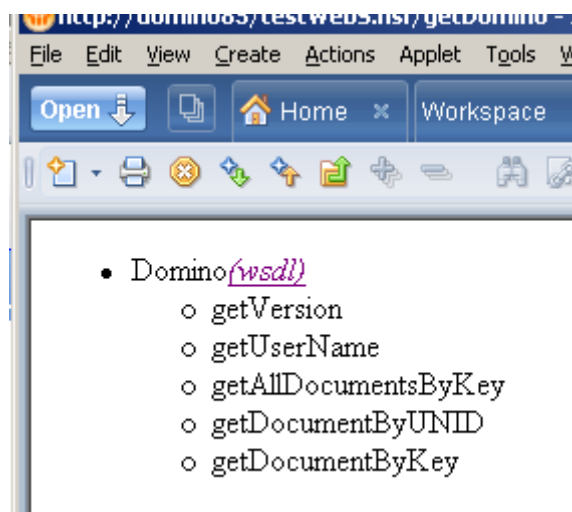
Lotus Domino maps the WSDL interface to an agent-like Web service design element that can be coded in LotusScript or Java. To be used, the Web service must be on a Domino server with HTTP enabled. (We can test the Web service through an HTTP session in the Notes client preview.) Access is through one of the following Domino URL commands:

- ?OpenWebService invokes the Web service in response to a SOAP-encoded message sent through an HTTP POST. An HTTP GET (for example, a browser query) returns the name of the service and its operations.
- ?WSDL returns the WSDL document in response to an HTTP GET.

Let's take a simple example.



In our sample my Web Services expose following function



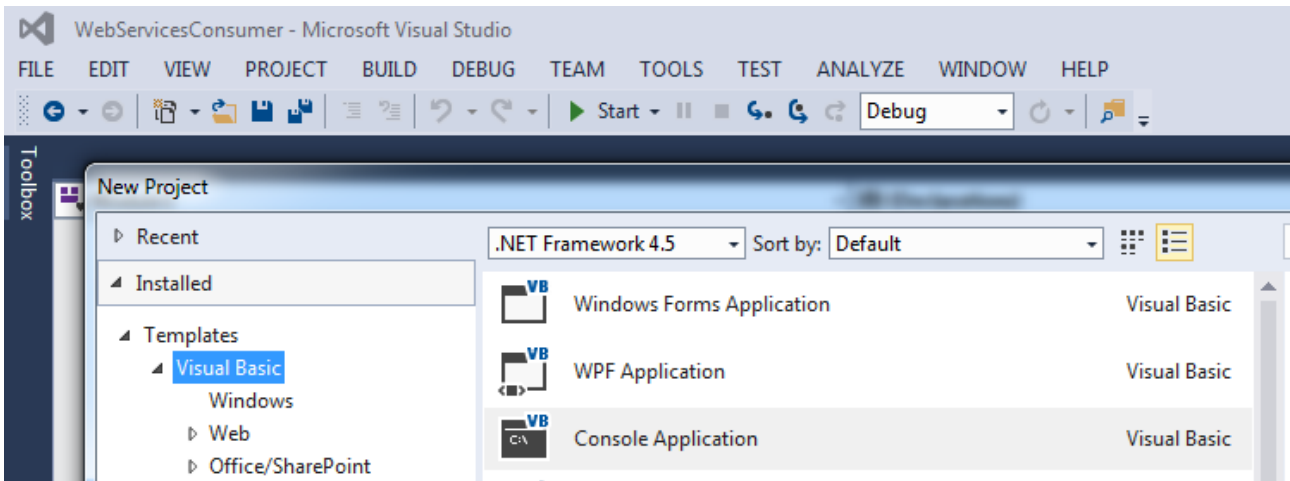
And this is part of my WSDL

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions targetNamespace="urn:webservices.n2a.com" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="urn:webservices.n2a.com"
xmlns:intf="urn:webservices.n2a.com" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <wsdl:types>
- <schema targetNamespace="urn:webservices.n2a.com" xmlns="http://www.w3.org/2001/XMLSchema">
<element name="getVersionReturn" type="xsd:string" />
<element name="getUserReturn" type="xsd:string" />
<element name="key" type="xsd:string" />
<element name="requestedFields" type="xsd:string" />
<element name="viewName" type="xsd:string" />
<element name="serverName" type="xsd:string" />
<element name="replicaId" type="xsd:string" />
<element name="filePath" type="xsd:string" />
- <complexType name="apachesoap_MapArray">
- <sequence>
<element maxOccurs="unbounded" minOccurs="0" name="item" type="apachesoap:Map" />
</sequence>
</complexType>
<element name="getAllDocumentsByKeyReturn" type="impl:apachesoap_MapArray" />
<element name="getDocumentByUNIDReturn" type="apachesoap:Map" />
<element name="getDocumentByKeyReturn" type="apachesoap:Map" />
</schema>
- <schema targetNamespace="http://xml.apache.org/xml-soap" xmlns="http://www.w3.org/2001/XML-
Schema">
- <complexType name="mapItem">
- <sequence>
<element name="key" nillable="true" type="xsd:string" />
<element name="value" nillable="true" type="xsd:string" />
</sequence>
</complexType>
```

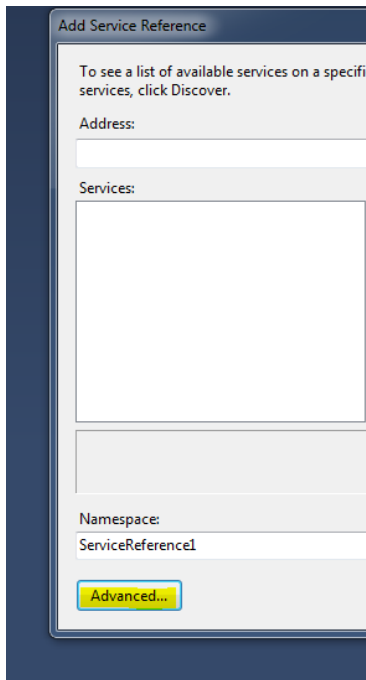
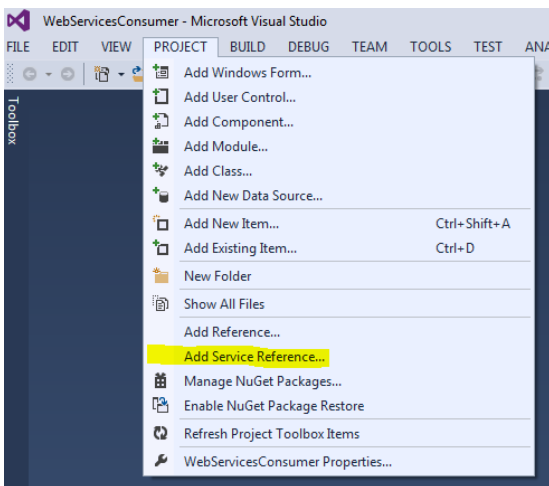
Now I set ACL to Anonymous NoAccess, and to consume Web Services we must authenticate over SOAP protocol.

Write a Visual Basic Consumer

In our sample we develop a sample Console Application to consume WebServices. Open you develop environment and create a new project as Console Application

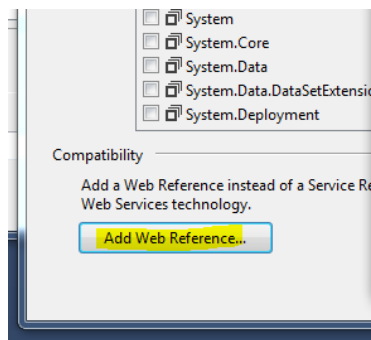


Add “Web Reference” from Project Menu select “Add Service Reference”

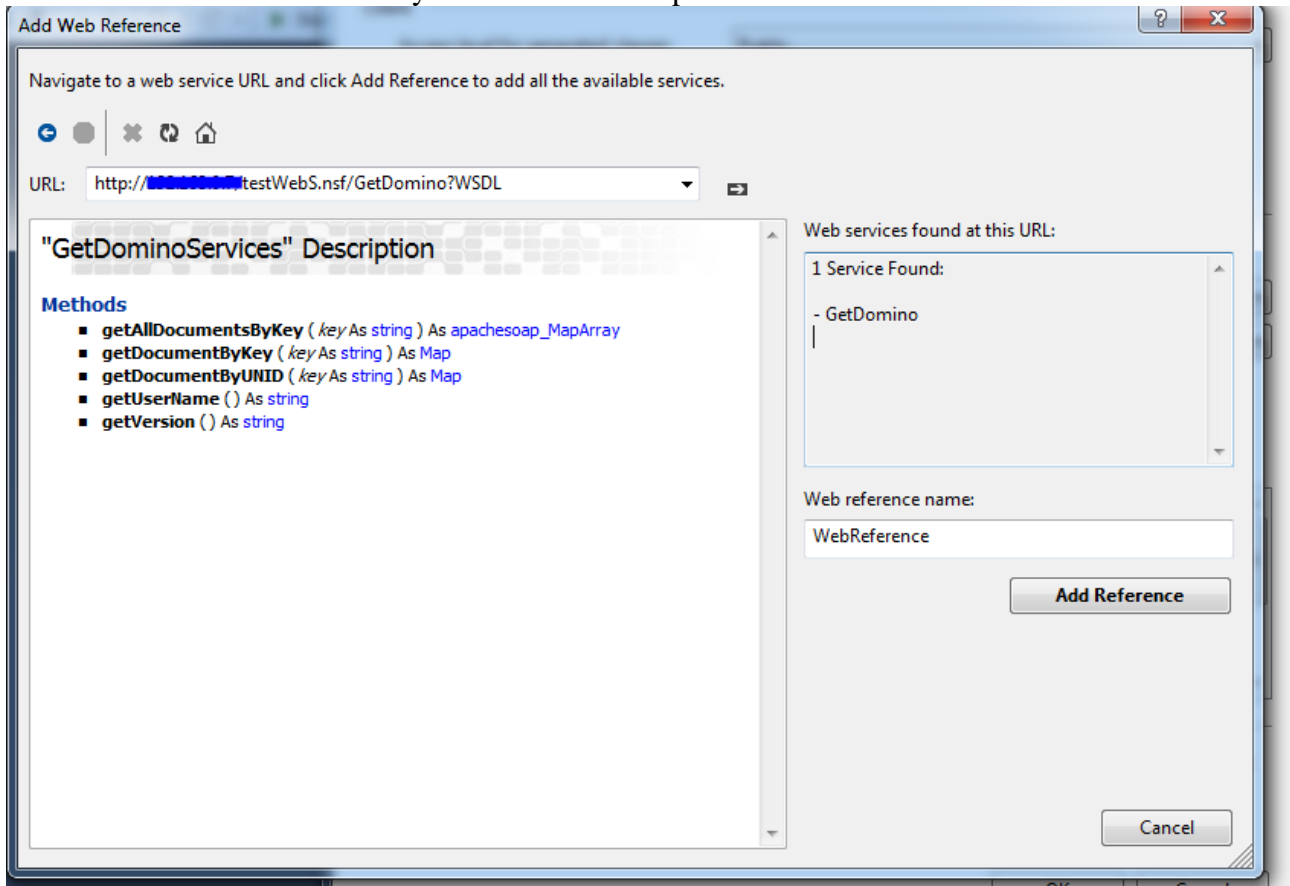


Then select “Advanced”

And select “Add Web Reference”



In next windows insert URL of your WDSL for sample



When you receive list of Methods you can click on "Add Reference".

Now we have all Reference need to develop our application.

In Our sample we have set ACL to no permit anonymous access, and we must extend some parameter to execute authentication in silent mode over SOAP protocol.

Basically all the client needs to do is create an authentication object, fill out the username and password, then pass them to the web service object.

In my Sample I define following Object

```
Dim userCredential As New Net.NetworkCredential
Dim myService As New ConsoleApplication1.GetDomino.GetDominoServices
Dim uri As New Uri(myService.Url)
Dim iCred As Net.ICredentials
```

userCredential is used to store Username and Password
myService is reference for our domino Web Services
uri is used to get Authentication method of services
iCred is used to make authentication

Initialize my object with following values

```
userCredential.Password = "password"
userCredential.UserName = "sadmin"
iCred = userCredential.GetCredential(uri, "basic")
```

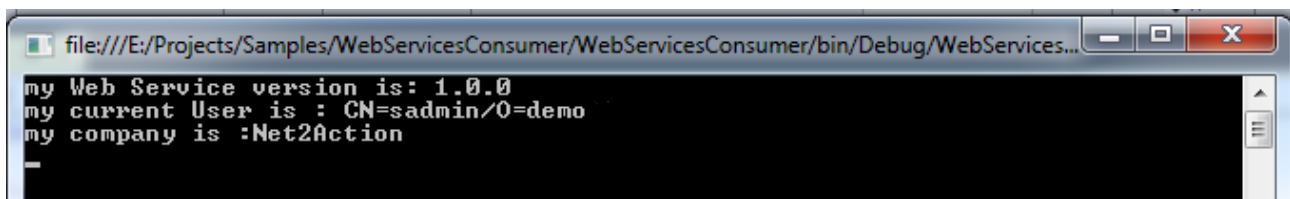
```
myService.PreAuthenticate = True
myService.Credentials = iCred
```

The **PreAuthenticate** property indicates whether to send authentication information with the initial HTTP request to the XML Web service.

Now I call Web Services and in my case my DB have one view called “Main” and I get name of company by key

```
Console.WriteLine("my Web Service version is: " & myService.getVersion)
Console.WriteLine("my current User is : " & myService.getUserName)
answer = myService.getDocumentByKey("000001", "companyName", "main", "", "", "")
Company = DirectCast(answer(0), ConsoleApplication1.GetDomino.mapItem).value
```

The final result is:

A screenshot of a Windows console window. The title bar shows the file path: file:///E:/Projects/Samples/WebServicesConsumer/WebServicesConsumer/bin/Debug/WebServices... The console output is as follows:

```
my Web Service version is: 1.0.0
my current User is : CN=sadmin/O=demo
my company is :Net2Action
-
```