

Microsoft Dynamics® AX 2012

# Consuming Web Services

White Paper

Microsoft Dynamics AX 2012 introduces a new model for how to consume external Web services. This white paper describes how to reference a Web service and provides example code that demonstrates how to use the service in X++.

Date: January, 2011

Author: Jim Travis, Senior Writer and Nima Kamoosi, Senior Developer, Application Integration Framework

Send suggestions and comments about this document to [adocs@microsoft.com](mailto:adocs@microsoft.com). Please include the title with your feedback.



---

## Table of Contents

|                                              |          |
|----------------------------------------------|----------|
| <b>Audience .....</b>                        | <b>3</b> |
| <b>Prerequisites.....</b>                    | <b>3</b> |
| <b>Introduction.....</b>                     | <b>3</b> |
| <b>Using the Bing API .....</b>              | <b>3</b> |
| <b>Install the Visual Studio tools .....</b> | <b>4</b> |
| <b>Create the service project .....</b>      | <b>4</b> |
| Add the service reference .....              | 5        |
| Add the project to the AOT.....              | 6        |
| Specify the deployment properties .....      | 6        |
| Verify the service reference.....            | 7        |
| <b>Using the Web service.....</b>            | <b>8</b> |

## Introduction

The previous version of Microsoft Dynamics® AX enabled you to consume external Web services from X++ code and to consume Web services hosted by Microsoft Dynamics AX from .NET Framework languages, such as Microsoft Visual C#®. To use a Web service, you added a reference in the **Web references** form, as described in the [Consume Web Services](#) topic on MSDN (<http://msdn.microsoft.com/en-us/library/cc654149.aspx>).

Microsoft Dynamics AX 2012 continues to enable Web services scenarios. However, you now use Microsoft Visual Studio to create and add a reference to a Web service. This white paper walks you through creating a reference to the Bing™ API Web service and provides sample code for a job that consumes the service.

## Audience

This white paper is intended for developers who integrate Web services with Microsoft Dynamics AX.

## Prerequisites

To benefit from this white paper, you should have experience in the following areas:

- Writing code in .NET Framework languages or X++
- Using Microsoft® Visual Studio®
- Setting up Microsoft Dynamics AX

You must have Microsoft Visual Studio 2010 installed on the computer that is running Microsoft Dynamics AX 2012.

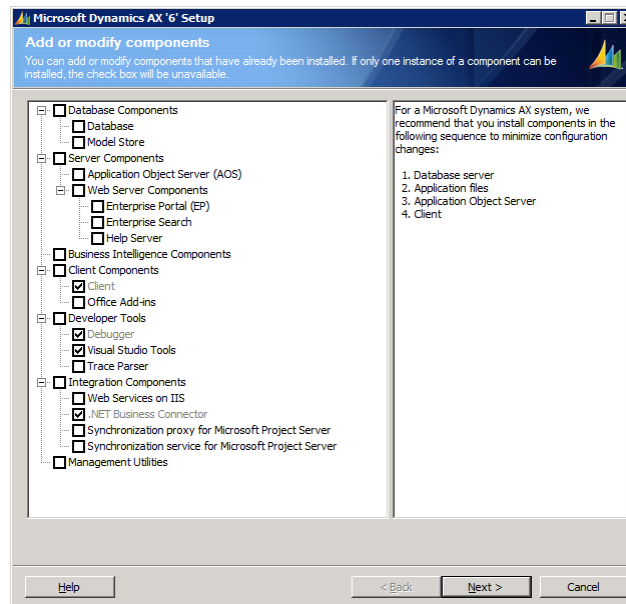
## Using the Bing API

To use the Bing API Web service, you must have the necessary resources and provision your application by getting an AppID. For more information about the Bing API and to create an AppID, see the [Bing Developer Center](#) (<http://www.bing.com/developers>).

## Install the Visual Studio tools

Follow these steps to install the Visual Studio tools for Microsoft Dynamics AX:

1. Run Microsoft Dynamics AX 2012 setup.
2. Under **Install**, click **Microsoft Dynamics AX components**.
3. Click **Next** to go to the **Add or modify components** page.
4. Under **Developer Tools**, select **Visual Studio Tools**.

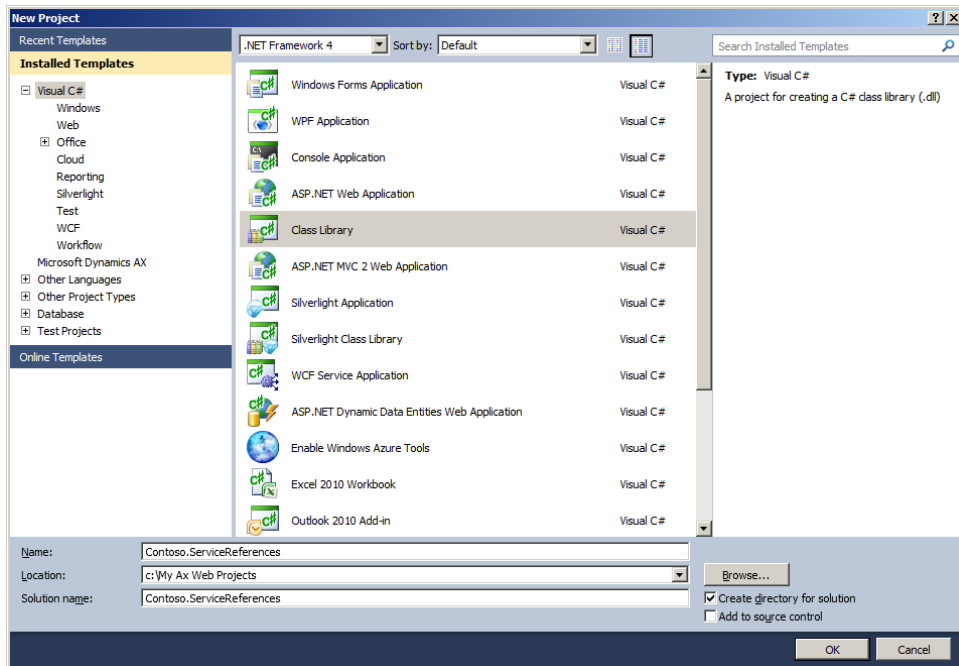


5. Click **Next** to step through the remaining setup pages.
6. Click **Install** to install the tools.
7. Click **Finish** to close the wizard.
8. Exit setup.

## Create the service project

Microsoft Visual Studio 2010 can create a project that Microsoft Dynamics AX can use to build a .NET assembly (and its configuration file) that exposes types from a Web service reference. Follow these steps to create a project with the Bing API Web service reference:

1. Open Microsoft Visual Studio 2010.
2. Create a new Visual C# **Class Library** project. (You can use other .NET Framework languages, but this walkthrough uses Visual C#.)
3. For the project name, type "Contoso.ServiceReferences".



4. Click **OK** to create the project.

## Add the service reference

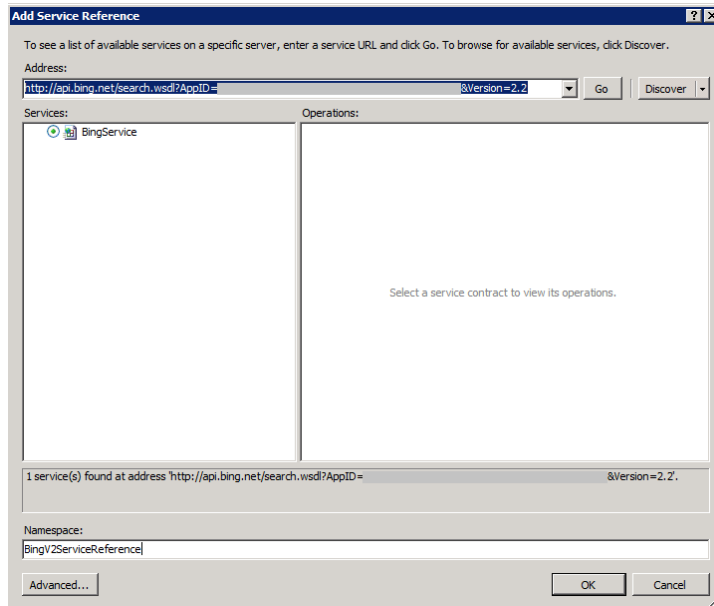
Follow these steps to add a reference to the Bing API Web service to your project:

1. In **Solution Explorer**, right-click the project name, and then click **Add Service Reference**.
2. In the **Add Service Reference** dialog box, type the URL for the Web service in the **Address** box. The URL for the Bing API Web service is:

<http://api.bing.net/search.wsdl?AppID=YourAppId&Version=2.2>

Replace the *YourAppId* text in the URL with your AppID.

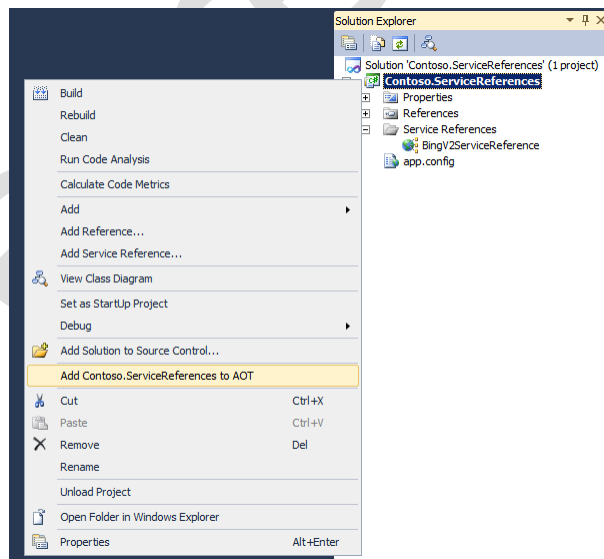
3. Click **Go** to locate the service.



4. Click **OK**.

## Add the project to the AOT

Next, add the project to the Application Object Tree (AOT) in Microsoft Dynamics AX. In **Solution Explorer**, right-click the project name, and then click **Add Contoso.ServiceReferences to AOT**.



Microsoft Dynamics AX imports the project and builds it.

## Specify the deployment properties

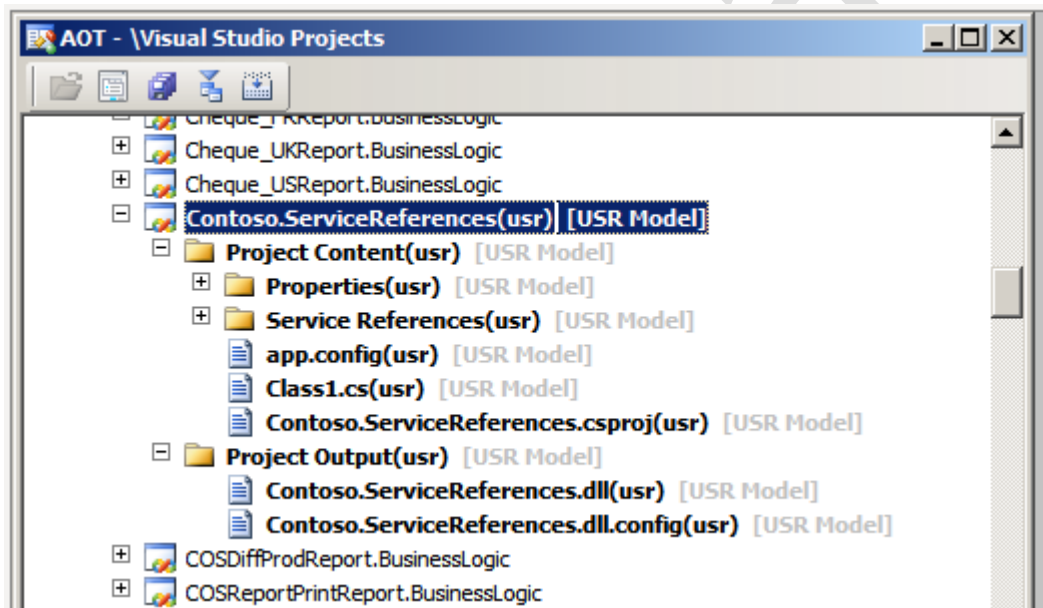
Next, set the deployment properties in Visual Studio. In the **Properties** window, specify the following values for the deployment properties.

| Name             | Value |
|------------------|-------|
| Deploy to client | Yes   |
| Deploy to EP     | No    |
| Deploy to server | Yes   |
| Deploy to SSRS   | No    |

## Verify the service reference

Follow these steps to verify that the reference to the Bing API Web service appears in the AOT:

1. Restart the Microsoft Dynamics AX client.
2. Open the development workspace.
3. In the AOT, browse to **Visual Studio Projects > C Sharp Projects**.
4. Verify that **Contoso.ServiceReferences** is listed as a project.



## Using the Web service

You can now use the Bing API Web service. In this section, you will create a job that performs a Bing search.

### Creating and configuring a service client object

To consume a Web service in Microsoft Dynamics AX 2009, you added a service reference and then created and used the service client object by using code similar to the following example.

```
WindowsLiveSearch.MSNSearchPortTypeClient searchService; // declare the service object
...
searchService = new WindowsLiveSearch.MSNSearchPortTypeClient(); // create the service object
searchResponse = searchService.Search(searchRequest); // use the service to issue a request
```

In Microsoft Dynamics AX 2012, you construct and configure an instance of a service client object by using code similar to the following example.

```
// Retrieve the X++ type for the Bing service client object.
clientType =
CLRInterop::getType("Contoso.ServiceReferences.BingV2ServiceReference.BingPortTypeClient");
// Use the AifUtil class to create an instance of the service client object.
_client = AifUtil::CreateServiceClient(clientType);
```

### Running the job

Jobs run on the client by default. If you use a Web service from an X++ class, remember to use the client or server method modifiers, or set the **RunOn** property on a specific class, to specify the location where the code is to be executed.

Create a new job. In the **Jobs Editor**, enter the following X++ code.

```
// Job to run a Bing search
// Search for "Dynamics AX"
static void CallBingService(Args _args)
{
    #define.AppId("Your AppID goes here")

    // variable for service client
    ClrObject clientType;
    // variable for service client type
    Contoso.ServiceReferences.BingV2ServiceReference.BingPortTypeClient _client;

    // variables for web query objects
    Contoso.ServiceReferences.BingV2ServiceReference.SearchRequest request;
    Contoso.ServiceReferences.BingV2ServiceReference.SourceType[] sourceTypes;
    Contoso.ServiceReferences.BingV2ServiceReference.SearchResponse response;
    Contoso.ServiceReferences.BingV2ServiceReference.WebResponse webResponse;
    Contoso.ServiceReferences.BingV2ServiceReference.WebResult[] webResults;
    Contoso.ServiceReferences.BingV2ServiceReference.WebResult webResult;

    int integer;
    str string;
    System.Exception ex;

    new InteropPermission(InteropKind::ClrInterop).assert();
```



```

// Always try and catch errors as CLR exceptions
try
{
    // Retrieve the X++ type for the Bing service client object.
    clientType =
CLRInterop::getType("Contoso.ServiceReferences.BingV2ServiceReference.BingPortTypeClient");
    // Use the AifUtil class to create an instance of the service client object.
    _client = AifUtil::CreateServiceClient(clientType);

    // Create the request
    request = new Contoso.ServiceReferences.BingV2ServiceReference.SearchRequest();
    request.set_AppId(#AppId);
    request.set_Query("Dynamics AX");
    sourceTypes = new Contoso.ServiceReferences.BingV2ServiceReference.SourceType[1]();
    sourceTypes.SetValue(Contoso.ServiceReferences.BingV2ServiceReference.SourceType::Web,
0);
    request.set_Sources(sourceTypes);

    // Configure the response
    response = _client.Search(request);

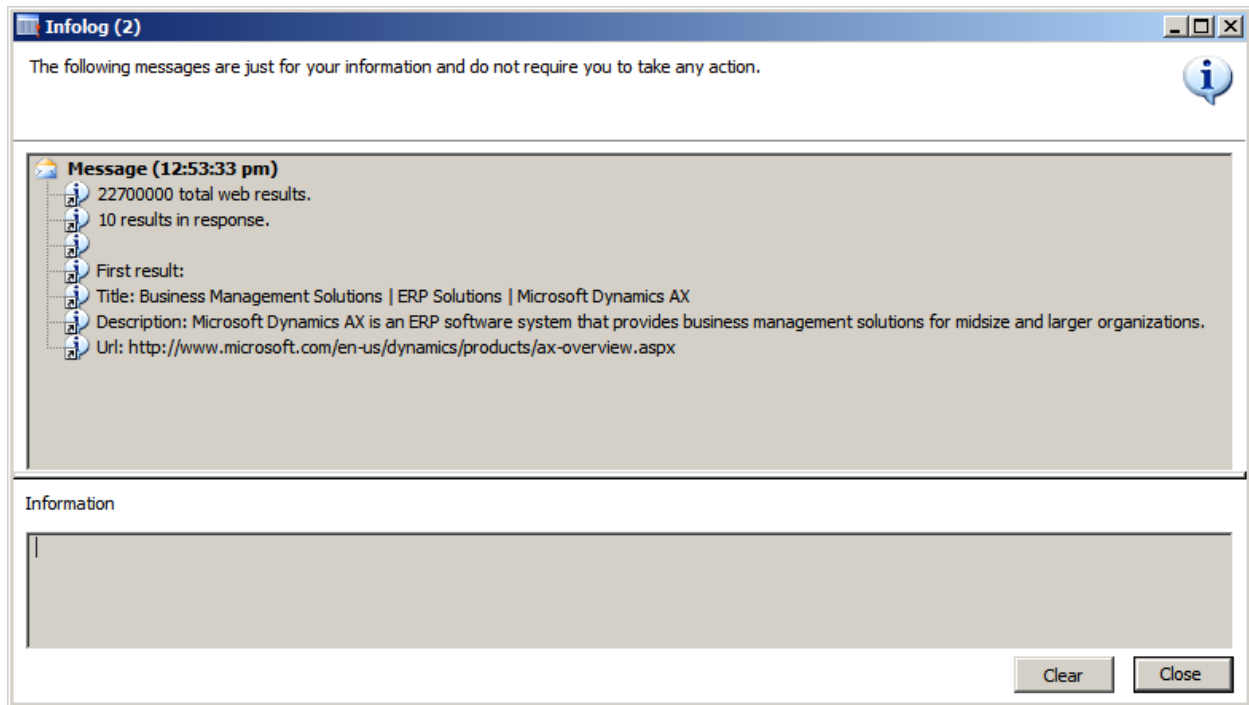
    webResponse = response.get_Web();

    // Get the search results
    webResults = webResponse.get_Results();
    webResult = webResults.GetValue(0);

    // Display the results in the Infolog
    integer = webResponse.get_Total();
    info(strFmt("%1 total web results.", integer));
    integer = webResults.get_Count();
    info(strFmt("%1 results in response.", integer));
    info("");
    info("First result:");
    string = webResult.get_Title();
    info(strFmt("Title: %1", string));
    string = webResult.get_Description();
    info(strFmt("Description: %1", string));
    string = webResult.get_Url();
    info(strFmt("Url: %1", string));
}
catch(Exception::CLRError)
{
    // handle the exception
    ex = CLRInterop::getLastException();
    info(ex.ToString());
}
}

```

Note that after the job runs, the following Infolog is displayed.



Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989  
Worldwide +1-701-281-6500  
[www.microsoft.com/dynamics](http://www.microsoft.com/dynamics)

This document supports a preliminary release of a software product that may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, the Microsoft Dynamics Logo, Microsoft Dynamics, Visual Studio, Visual C#, and Bing are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

**Microsoft**