Getting Started with Checkbox Web Services

INTRODUCTION

This guide will demonstrate how to reference and use the Checkbox Web Services using Visual Studio 2010 and C#.

FOUNDATION

Checkbox 5 Web Services are built with the Windows Communication Foundation (WCF). We recommend building your application using the latest version of the .NET Framework (version 4.0 at the time of this document's release) in order to take advantage of improvements in the WCF stack.

Each of our Web Services have a Web Services Description Language (WSDL), which informs the client how to communicate with the service. This means that the services may be accessed using a number of different clients (environments and languages, i.e. Java). It is up to you, the developer, to reference the Web Services appropriately.

CHANGES IN CHECKBOX VERSION 5

If you are familiar with using Web Services in prior versions of Checkbox, it is no longer required to use the "Web Services User" role in order to access the Web Services in Checkbox version 5. Any registered Checkbox user can be used to connect to a Web Service, but the authenticated user must have the appropriate security permissions to make any changes to Checkbox items (surveys, styles, libraries, invitations, etc.).

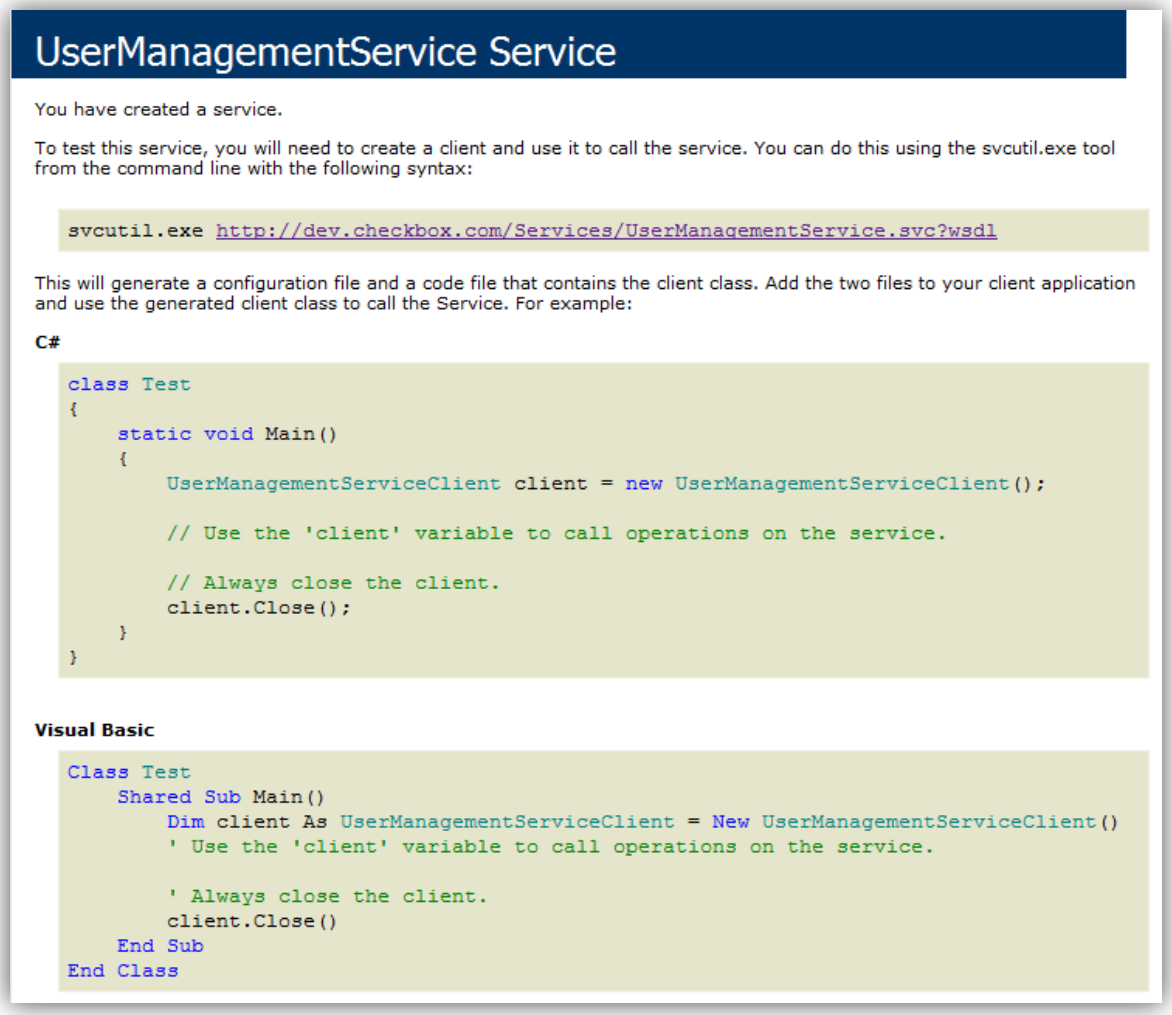
There is no need to install or copy any additional files. The Web Services are all located in the Services directory of your Checkbox installation. After successfully installing the Checkbox website they will be ready to use.

HOW DO I KNOW IF THE WEB SERVICES ARE READY TO USE?

The first thing you will want to do is ensure that your Web Services are installed correctly and can be accessed by your application. The easiest way to do that is to type a service's URL into a web browser.

Let's take a look at the User Management Service by visiting the following URL: http://[CHECKBOX URL]/Services/UserManagementService.svc

The following screen shot should be similar to what you see if the Web Services are accessible.



UserManagementService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://dev.checkbox.com/Services/UserManagementService.svc?wsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        UserManagementServiceClient client = new UserManagementServiceClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

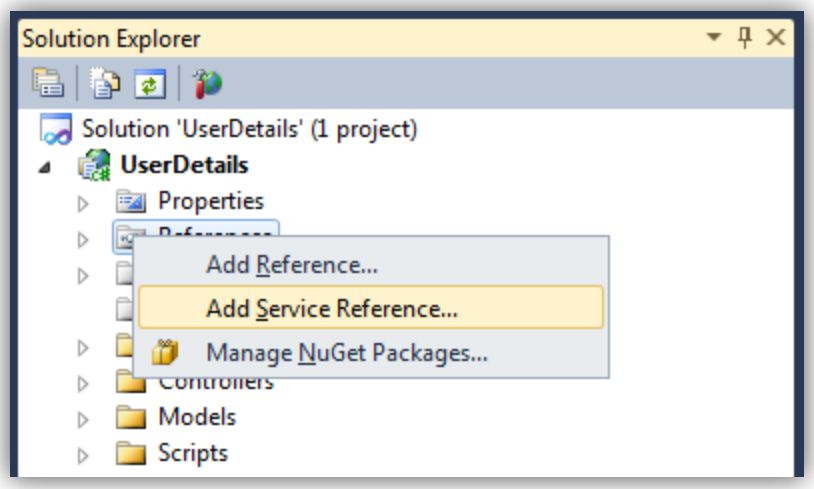
```
Class Test
    Shared Sub Main()
        Dim client As UserManagementServiceClient = New UserManagementServiceClient()
        ' Use the 'client' variable to call operations on the service.

        ' Always close the client.
        client.Close()
    End Sub
End Class
```

ADDING A SERVICE REFERENCE

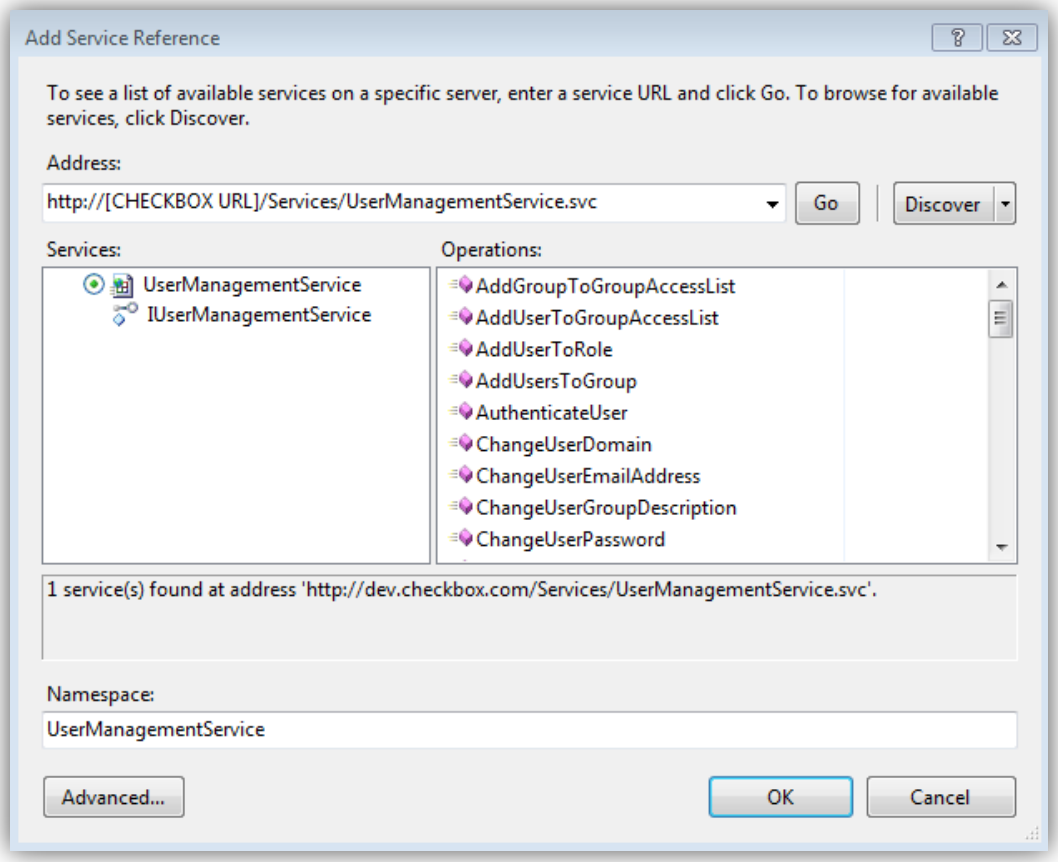
Now that we know our service is working, we can reference it in our project.

Note: we are using C# and Visual Studio 2010 for this example, but you may use whatever environment or language you would like.

1) Create a new project or open an existing one
2) Right click the "References" folder
3) Find and select "Add Service Reference" (for .NET Framework 3.0 and up) or "Add Web Reference" (for .NET Framework 2.0 and 1.1).



4) Enter the address for the service you want to use
5) Click "Go"
6) Select the contract you want to use
7) Enter a name for the proxy class that will be created
8) Click "OK"



CONSUMING YOUR NEW SERVICE CLIENT

Now that the Service Reference has been successfully created it can be used in a project.

Here is an example of what a C# code file may look like to consume the "UserManagementService" client we created.

```
using System;
using UserManagement.UserManagementService;

namespace UserManagement
{
    class Program
    {
        static void Main(string[] args)
        {
            var client = new UserManagementServiceClient();
            var result = client.AuthenticateUser("admin", "admin");

            // was the call successful?
            if(!result.CallSuccess)
                throw new Exception(result.FailureMessage);

            // authentication was successful, now we have an auth token
            var authToken = result.ResultData;

            // let's use the auth token to get all user groups
            // we have permission to view
            var allUserGroupsResult = client.ListUserGroups(authToken, // auth token
                                                            -1,        // page number
                                                            -1,        // page size
                                                            null,      // sort field
                                                            true,      // sort ascending
                                                            null,      // filter field
                                                            null);     // filter value

            if (!allUserGroupsResult.CallSuccess)
                throw new Exception(result.FailureMessage);

            var groupList = allUserGroupsResult.ResultData;

            // now we can do something with the group list
            // ....
        }
    }
}
```

WELL, THAT WAS FUN. NOW WHAT?

You have reached the end of our Getting Started guide. If you followed all of the steps in this document, you should have enough information to start accessing the Checkbox 5 Web Services.

Complete, working examples of simple projects with premade Web Service clients are freely available upon request. Please contact your account representative, or email sales@checkbox.com, for more information.