# Microsoft Dynamics CRM SDK

Web Resource Utility Readme

The Web Resource Utility streamlines the task of uploading many files as Microsoft Dynamics CRM Web Resources in a consistent manner.

# Summary

The following provides a brief overview of the tool’s functionality and implementation.

## Purpose

This GUI developer tool is designed to expedite the process of locating and uploading files as Web Resources into Microsoft Dynamics CRM. Specifically, the tool provides the following functionality:

1. Create and save Microsoft Dynamics CRM server/organization information locally. User will connect to a single destination server/organization to upload web resources. User can save as many server/organization configurations as desired.
2. Choose an existing Microsoft Dynamics CRM Unmanaged Solution to contain the uploaded Web Resources after connecting to a server/organization.
3. Create and save a “Web Resource Packages” locally. A package is simply a set of selected files relative to a root directory that can be uploaded as web resources. Folder Browser dialogs, Search by Filename, and sortable grids help the user locate desired files. User has multiple options to name and upload the Web Resources. See “Using the Utility” for specifics.

## Technology

The Web Resource Utility is created using Windows Presentation Foundation (WPF) in C# .NET 4.0. The source code follows a MVVM design pattern, making use of [Data Binding](http://msdn.microsoft.com/en-us/library/system.windows.data.binding.aspx), [Command Binding](http://msdn.microsoft.com/en-us/library/system.windows.input.commandbinding.aspx), data validation, [LINQ to Objects](http://msdn.microsoft.com/en-us/library/bb397919.aspx), and [LINQ to XML](http://msdn.microsoft.com/en-us/library/bb387098.aspx). Data is retrieved and stored in two (2) XML files and Microsoft Dynamics CRM using the *OrganizationServiceProxy* and *OrganizationServiceContext* classes.

See “CrmServiceHelpers” sample for more information on connecting to Microsoft Dynamics CRM.

# Installation

The solution must be built in Visual Studio 2010 before running. Double-click the “WebResourceUtility.sln” to open the source code in Visual Studio 2010. Open and view the Solution Explorer.

## Solution Dependencies

This solution uses linked files and requires access to the following files in order to compile:

In the DataAccess folder:

* sdk\samplecode\cs\helpercode\CrmServiceHelpers.cs
* sdk\samplecode\cs\helpercode\deviceidmanager.cs
* sdk\samplecode\cs\helpercode\myorganizationcrmsdktypes.cs
* sdk\samplecode\cs\helpercode\optionsets.cs

You may need to manually update the links for these files before trying to build the solution or you may experience compilation errors.

This solution includes references to the following files:

* sdk\bin\microsoft.crm.sdk.proxy.dll
* sdk\bin\microsoft.xrm.sdk.dll

## Build the Solution

Right-click the “WebResourceUtility” SLN file in Solution Explorer and click build. An EXE will be produced in the bin/Debug folder.

## Run Utility

Double-click the EXE in the bin/Debug folder or press F5 to debug directly in Visual Studio 2010.

# Using the Utility

There are three (3) main tabs that play a role in the process of uploading web resources: Connections, Solutions, and Web Resource Packages. On each tab there is a Red/Green indicator and text representing the user’s progress on that tab. The user may only upload Web Resources when all indicators are Green and contain text.

## Connections

This tab provides a list of server/organization configurations and provides the following four (4) actions:

1. **New** – Creates a new connection.
2. **Delete** – Deletes the selected connection.
3. **Save** **Connections** – Saves all connections in the list.
4. **Connect!** – Prompts the user for password. This action causes authentication with CRM and retrieval of existing Unmanaged Solutions. Displays any errors that occur. Activates the selected connection if successful. The red indicator will turn green upon success.

Note: Allow up to a minute for this process to complete. When the process completes, if the query returns a single Unmanaged Solution named “Default”, it will auto-activate the Solution and take the user to the Packages tab. Otherwise, the user will be taken to the Solutions tab to choose an existing Unmanaged Solution.

## Solutions

This tab provides a list of Unmanaged Solutions and a single action:

1. **Choose –** Activates the selected solution. Only one (1) can be active at a time.

The Active Solution is the destination container for uploaded Web Resources. The Red indicator will turn to Green when a solution is active.

The Solution entity in CRM has a parent Publisher. The Publisher has a field to store its customization prefix, i.e., “new\_”. The customization prefix is always used in naming Web Resources in the CRM user interface and is therefore replicated in this utility.

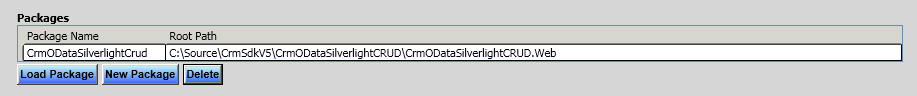
Example: “new\_/MySamplesPackage/ClientBin/MySilverlightSample.xap”

For more information on solutions see “Introduction to Solutions” in the Microsoft Dynamics CRM 2011 SDK.

## Web Resource Packages

This tab provides all the functionality to create and save packages, search for files, add web resources and upload resources to Dynamics CRM. For readability, this tab is broken down into multiple sections.

##### Packages section



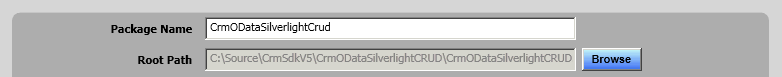
This section contains a grid that displays all saved Web Resource Packages and provides three (3) actions:

1. **Load Package** – Activates the selected package and displays the detail data. Only one (1) package can be active at a time for edit/upload. User must save any changes to active package prior to loading a new package.
2. **New Package** – Creates a new Package in the list. Auto-saves the Active Package and Activates the newly created package allowing it to be edited.
3. **Delete** – Deletes the package selected in the list. If the package selected in the list is also the Active Package, it will still be deleted.

##### Package Detail section

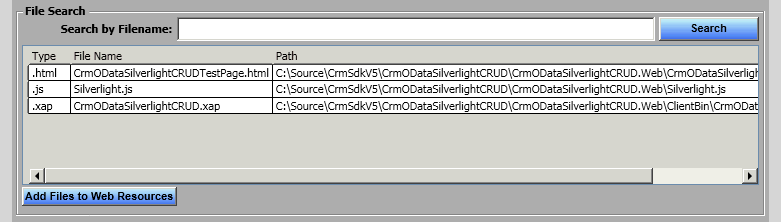
This section allows user to edit and save Package details, search for files, add files as Web Resources, upload Web Resources to CRM and view upload results in an output window.

**Package Info**



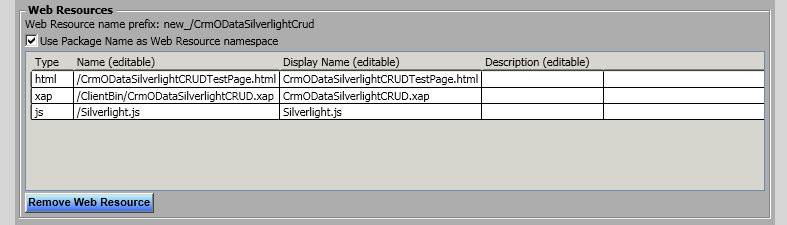
1. **Package Name** – Supply a meaningful name that describes a set of Web Resources. Don’t use spaces, hyphens or double forward-slashes. Only alphanumeric, periods, underscore and singe forward-slash are allowed.
2. **Root Path** – Use the Browse button to open a Folder Browser Dialog and select the folder containing all web files.

**File Search**



1. **Search by Filename –** User can refine the list of files by entering search text. Text is case-sensitive and will match any files where the Path contains the search text.
2. **Search –** User can refresh the list of files shown by clicking Search button.
3. **Add Files to Web Resources –** Converts the selected rows into the Web Resource Grid.

**Web Resources**



1. **Web Resource name prefix –** This label shows what the prefix of each Web Resource will be when uploaded to Microsoft Dynamics CRM. Web Resource name is a concatenation of the prefix and the Name column.

**NOTE:** the prefix, like “new\_” will only display if there is an Active Solution from the Solutions tab. The prefix comes from the solution’s publisher.

1. **Use Package Name as Web Resource namespace –** This checkbox determines whether or not the Package Name is used on the end of the prefix. It is recommended to keep related web resources under their own unique ‘namespace’ to help assure files names are unique. Consider the possibility that two different solutions have a javascript file named “account.js” and provide different functionality. Namespacing your files will help assure uniqueness and provides a top-level virtual folder structure under which all web resources of a package reside.

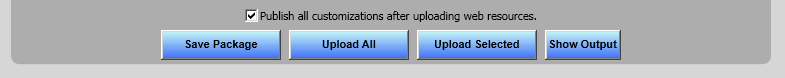
**Example**

* Publisher Prefix = “new\_”
* Package Name = “CrmODataSilverlightCrud”
* Name (editable) column = “/ClientBin/CrmODataSilverlightCRUD.xap”
* Web Resource Name used when uploading / searching for duplicates:

“new\_/CrmODataSilverlightCrud/ClientBin/CrmODataSilverlightCRUD.xap”

1. **Remove Web Resource –** Removes one or more selected Web Resources from the grid.
2. **Editable columns –** Display Name and Description columns can be changed as desired, without hesitation. There exists a ramification of changing the Name column from the default value. Changing the name can break any relative references between individual web resources. For example, you may have an HTML web resource that has a script tag referencing a JS web resource by relative path. Leaving the default names helps assure referential integrity. Changing the Name column should be done so with this understanding.

**Package commands**



1. **Publish all customization after uploading web resources** – This checkbox determines whether a *PublishAllXmlRequest* message is executed after the application has completed uploading all web resources. Publishing web resources ensures their content is updated and publicly available for use/testing.
2. **Save Package** – Saves the details of the Package to XML file, including the Package Name, Root Path and all Web Resources in the Web Resources Grid.
3. **Upload All** – Can only be clicked when all tabs are active (green). Uploads all Web Resources in the grid to Dynamics CRM. The tool will properly create or update based on the Web Resources existence in Dynamics CRM. Displays the output window and disables all buttons until process is complete.
4. **Upload Selected** - Can only be clicked when all tabs are active (green). Uploads the selected Web Resources in the grid to Dynamics CRM. The tool will properly create or update based on the Web Resources existence in Dynamics CRM. Displays the output window and disables all buttons until process is complete.
5. **Show Output** – Displays the output window containing the results of uploading Web Resources.

# XML Schema

There are two (2) XML files that store data for the utility. When you build the solution, the XML files will be added to the bin/Debug folder along with the .EXE.

1. **configurations.xml –** Store all connection information from the Connections tab.

**Example schema:**

<Configurations>

<Configuration name="MaxusBeta"

server="maxus-beta"

orgName="AdventureWorksCycle"

userName="administrator"

domain="maxus-betadom" />

</Configurations>

1. **packages.xml –** Stores all package and web resource information from the Packages tab.

**Example schema:**

<UtilityRoot>

<Packages>

<Package name="CrmODataSilverlightCrud" rootPath="C:\Source\CrmSdkV5\CrmODataSilverlightCRUD\CrmODataSilverlightCRUD.Web" isNamePrefix="True">

<WebResourceInfo name="/CrmODataSilverlightCRUDTestPage.html"

filePath="\CrmODataSilverlightCRUDTestPage.html"

displayName="CrmODataSilverlightCRUDTestPage.html"

type="html"

description="" />

<WebResourceInfo name="/ClientBin/CrmODataSilverlightCRUD.xap"

filePath="\ClientBin\CrmODataSilverlightCRUD.xap"

displayName="CrmODataSilverlightCRUD.xap"

type="xap"

description="" />

<WebResourceInfo name="/Silverlight.js"

filePath="\Silverlight.js"

displayName="Silverlight.js"

type="js"

description="" />

</Package>

</Packages>

</UtilityRoot>

# Known Issues

1. On the Connections screen, the “Connect!” button sometimes causes an error, “Unknown: 0x80043455” to be displayed. This is an exception arising in the DeviceIdManager.cs file. To resolve this sporadic error, simply click Connect again and re-enter your password. Repeat up to 3 times.
2. When uploading Web Resources to CRM, the service is ignoring the DisplayName column on the Web Resources grid. It defaults the Web Resources’ DisplayName equal to the Name. You can manually change the DisplayName value on the server.
3. When connecting to Microsoft Dynamics CRM Online, use the following values to connect:

|  |  |
| --- | --- |
| Label | Value |
| Server | crm.dynamics.com |
| Organization | <your organization name> |
| Username | <your Live Id e-mail address> |
| Domain | <leave blank> |

When prompted for your password enter your Windows Live Id password.

1. The application includes logic to only allow updating web resources where the IsCustomizable managed property equals true. The IsCustomizable managed property is supposed to be enforced only when the web resource is part of a managed solution. The additional constraint in this application was added to fix an issue where the application failed when attempting to update web resources in a managed solution where it is not allowed. An unwanted side effect is that the application is now unable to update web resources in an unmanaged solution where the IsCustomizable managed property equals false. If you intend to set the IsCustomizable managed property to true, the workaround is to wait until you have finished your development using this application and set the IsCustomizable property just before you export the solution as managed.

# Troubleshooting

## Build error(s)

If you get a build error when trying to build the utility, be sure to assure all files and assembly references resolve prior to building.

## Run-time errors

If you are experiencing errors when running the sample, check the following.

* Make sure the two XML files are available in the same directory as the EXE.

# Copyright

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, ActiveX, BizTalk, Excel, Great Plains, Internet Explorer, JScript, Microsoft Dynamics, MSN, Outlook, PivotTable, PivotChart, Visual Basic, Visual Studio, Windows, Windows Live, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.