# Microsoft Dynamics CRM SDK

## GeneratePicklistEnums Readme

## Overview

This sample extension to the code generation tool (CrmSvcUtil.exe) generates enumerations for the option sets, state codes, and status codes in a Microsoft Dynamics CRM organization. When compiled to an assembly and loaded by the CrmSvcUtil program, this extension generates an output source code file containing these enumerations that can be included in your application’s project. These enumerations make your code more readable and are supported by Microsoft Visual Studio IntelliSense. For more information on running CrmSvcUtil.exe, see the related [topic](http://msdn.microsoft.com/en-us/library/gg327844.aspx) in the Microsoft Dynamics CRM SDK.

## Contents

* **CodeCustomizationService.cs** - Implements **ICustomizeCodeDomService** to manipulate the [Code Document Object Model](http://msdn.microsoft.com/en-us/library/650ax5cx.aspx) (DOM).
* **FilteringService.cs** - Implements **ICodeWriterFilterService** that provides custom logic to indicate if a given item (option set, state, or status code) will have code generated for it.
* **NamingService.cs** – Implements custom logic to build enumeration type names that will appear in the generated output file.
* **GeneratePicklistEnums**.**csproj** – Visual Studio project file.
* **GenerateOptionSets**.**bat** – Batch command file to execute the CrmSvcUtil.exe command with the sample extension. You must customize the command in the batch file as appropriate for your Microsoft Dynamics CRM installation.

## Instructions

## How to Build the Sample

To build this sample extension, follow these steps.

1. Edit the GenerateOptionSets.bat file as appropriate for your Microsoft Dynamics CRM installation. Specifically, indicate the correct server and organization in the /url portion of the command and specify the http or https protocol depending on how your server is configured.
2. Open the solution file GeneratePicklistEnums.sln in Visual Studio 2010.
3. Press F6.

## How to Run the Sample

1. Change the working folder to the bin\Debug folder of your project.
2. Run the GenerateOptionSets.bat file.

**Expected Results**

You should see an OptionSets.cs file in your working folder.

If you get a run-time authentication error, you may need to provide a user name and password argument to the command in the batch file because your default credentials were not found in the system credential cache. For more information, refer to the SDK topic [Create Early-Bound Entity Classes with the Code Generation Tool (CrmSvcUtil.exe)](http://msdn.microsoft.com/en-us/library/gg327844.aspx).

A sample OptionSets.cs output file is provided in the SampleCode\CS\HelperCode folder. The sample located at SampleCode\CS\QuickStart makes use of this sample enumerations file.

**About the Extension Source Code**

By default, the sample extension does not generate state codes. This is required because the standard output of CrmSvcUtil, without this extension, includes these state codes. Developers using the early-bound types in their applications, for example: SampleCode\CS\HelperCode\MyOrganizationCrmSdkTypes.cs, will be able to include the types file and the enumerations file in their application and compile it.

If you are not using this sample extension in conjunction with the standard types output from CrmSvcUtil.exe, both of these preprocessor directives should be removed or commented out so that state code enumerations will be generated in the output file.

FilteringService.cs: #define SKIP\_STATE\_OPTIONSETS

CodeCustomizationService.cs: #define REMOVE\_PROXY\_TYPE\_ASSEMBLY\_ATTRIBUTE

## About the Generated Code

When reviewing the extension’s source code, one thing that you should be aware of is how the code handles duplicate option names or names that appear to be duplicated after removing spaces to form a valid identifier. The code generates a valid identifier name for the option in an option set, and if that identifier has already been used in the option set, it will append an underscore and number to the end of the identifier. For example, if an option set contained options “Item,” “Item,” “ItemValue,” and “Item Value”, the generated enumeration option names would be “Item”, “Item\_2”, “ItemValue,” and “ItemValue\_2.” It is worth noting that if an option’s name is changed and the code for the option sets is regenerated, any existing code will be broken because the code generated is based on the name of the option.

## Important Notes

* The sample files are not intended to be used in a production environment. You should deploy this sample to a test environment and examine it for interaction or interference with other parts of the system.
* Before you deploy this sample to a production environment, make sure that you consider the existing customizations you may have implemented in Microsoft CRM 2013.

## 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.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, ActiveX, BizTalk, JScript, Microsoft Dynamics, Outlook, SharePoint, SQL Server, Visual Basic, Visual Studio, Windows, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.