



Creating Custom Platform Translations Microsoft Dynamics NAV 2017

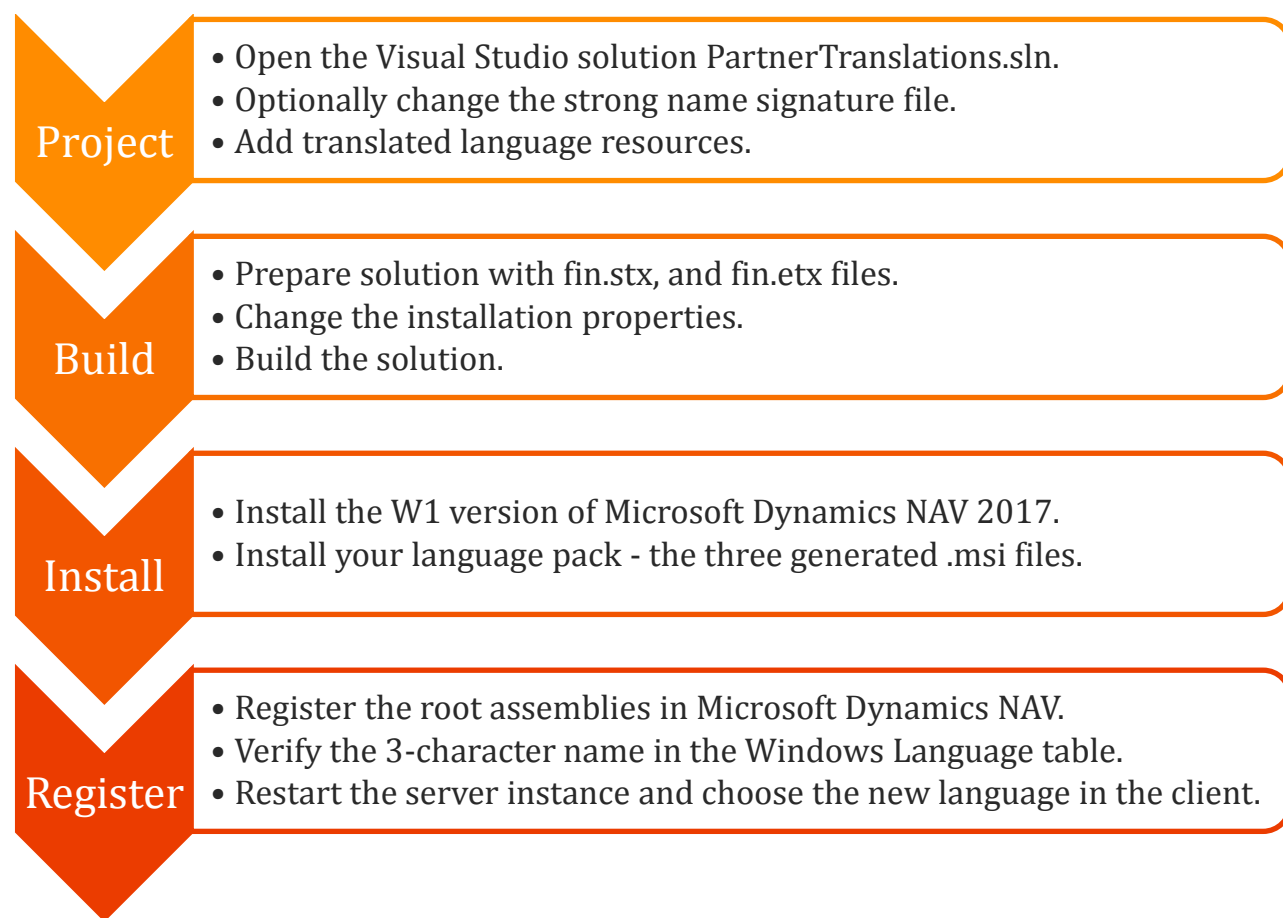
November 2016

Contents

High-level overview of the process	3
The Partner Localization and Translation Licensing Program and Custom Translations of the Microsoft Dynamics NAV platform resources	4
Platform strings maintained in managed assemblies and fin.stx and fin.etx files.	4
Creating a country/region-specific installer	5
Changes from the Microsoft Dynamics NAV 2013 R2 Partner Translation Toolkit	6
Changes to installation of Help content	6
New resource assemblies	6
Changes from the Microsoft Dynamics NAV 2015 Partner Translation Toolkit	7
Changes to Client Add-ins table	7
Changes resource assemblies	7
Project - Translate Existing Resource Files to the New Language	8
Unpack the PLLP Translation Toolkit	8
Install the WiX Toolset	8
Identify Visual Studio Solution in the Partner Translation pack	8
Customize strong name signature	9
Create and add language resource files	9
Important information when translating NAV resource strings	9
Using the ResX Translator tool in order to automatically translate all resources	9
Creating an account for Microsoft Translator	10
Use the ResX Translator to translate .resx files for Platform Assemblies	12
Add translated resource files to Visual Studio solution	12
Build – Create new language pack installers	13
Copy text resource files	13
Change installation properties	13
Build the solution	14
Install – Add your language pack to the W1 version	15
Install the new language pack	15
Register - Set trusted assemblies and select new language	16
Manually register root assemblies with strong name key token in the Add-ins table	16
Register root assemblies with the Partner Translation Tools	17
Restart the Microsoft Dynamics NAV Server service	18
Connect to the Microsoft Dynamics NAV Server and select the new language	18
Tips And Tricks	19
Add .resx files of a certain language to the satellite assembly projects	19
Validate correctly formatted strings in the translations	19
Ampersands in strings	20
Troubleshooting	21

High-level overview of the process

The following graphic illustrates the main steps in deploying Microsoft Dynamics NAV in a new localized version:



The following sections describe the steps you must take in order to deploy Microsoft Dynamics NAV in a new localized version.

The Partner Localization and Translation Licensing Program and Custom Translations of the Microsoft Dynamics NAV platform resources

Microsoft Dynamics NAV 2017 includes three large areas for translation:

- Application code and metadata in application objects in the database.
- Platform strings maintained in managed assemblies, and fin.stx and fin.etx files.
- Creating a country/region-specific installer.

This document covers only the second and third bullets, mainly focusing on providing partner translated resources for platform .NET assemblies, and building localized language packs.

Platform strings maintained in managed assemblies and fin.stx and fin.etx files.

Translated resources for the assemblies that Microsoft Dynamics NAV uses are provided in satellite assemblies. A satellite assembly contains the translated resources for another assembly, in another language, in a resource set for that language. It has the same name as the target assembly, but instead of the file name ending in .dll, it ends in .resources.dll. In addition, a satellite assembly for a certain language must be placed in a subdirectory of the target assembly with the name of the language identifier for that language. For more information, see [Language Identifier Constants and Strings](#).

For example, for the assembly with common strings that are used by the Microsoft Dynamics NAV platform, Microsoft.Dynamics.Nav.Language.dll, the satellite assembly for the Catalan language can be provided in the following subdirectory:

ca-ES\Microsoft.Dynamics.Nav.Language.resources.dll

Typically, satellite assemblies are provided by the vendor of the originating assemblies. The default rule in the .NET Framework is that if a vendor has signed an assembly with a strongly named signature, the satellite assembly must also be signed with that same signature; otherwise it will not be recognized by the .NET Framework. This behavior creates a problem for third-party vendors of partner translations for the Microsoft Dynamics NAV platform, because those vendors do not have access to the Microsoft strongly named signature and signing process.

To enable partners to provide translations in custom satellite assemblies, the Microsoft Dynamics NAV platform includes support for loading non-Microsoft satellite assemblies if the following requirements are fulfilled:

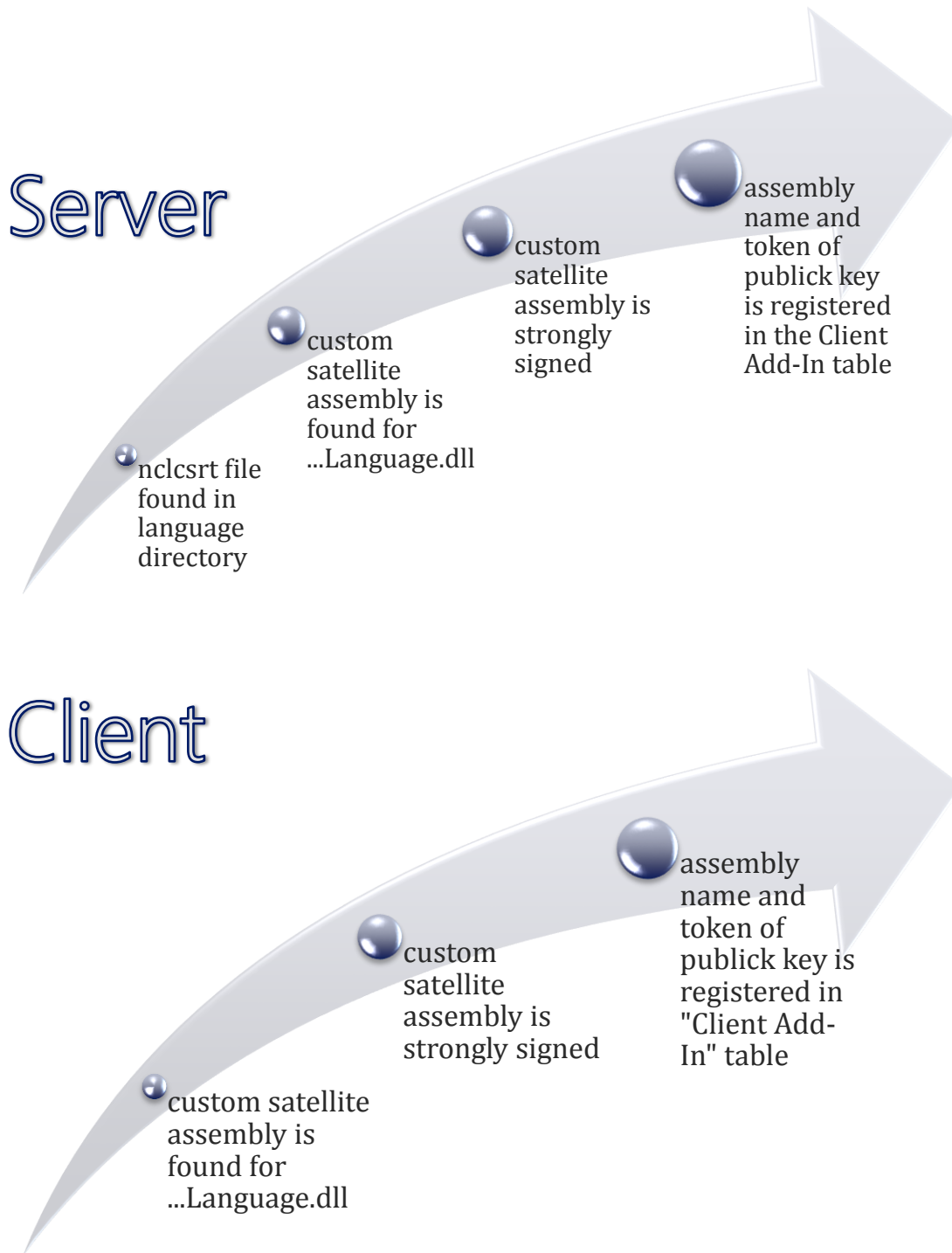
- The assembly follows the naming scheme for satellite assemblies and is placed in the respective folder with the correct language identifier.
- The assembly is fully strong name signed and the token of the public key is registered in the **Add-in** table, table 2000000069, in the database.

If the assembly targets an assembly that is stored in a subdirectory of the respective process executable (for example, for an add-in or .NET interop component), the custom satellite assembly is still placed in the folder with the language identifier in the directory of the executable.

For example, for the assembly Microsoft.Dynamics.Nav.Client.TimelineVisualization.dll that contains the Timeline visualization control from a control add-in in folder the Service\Add-ins\Timeline folder, a custom satellite assembly for Catalan would have the following file path:

Service\ca-ES\Microsoft.Dynamics.Nav.Client.TimelineVisualization.resources.dll.

The following diagram shows the checks that are done by an installation of Microsoft Dynamics NAV 2017 when a PLLP translation is loaded. If only checks on both the client and server machine are successful, then a PLLP translation will be available for selection to a client.



Creating a country/region-specific installer

After all translated artifacts have been made available, the .msi files are built and can be installed individually.

Changes from the Microsoft Dynamics NAV 2013 R2 Partner Translation Toolkit

Changes to installation of Help content

Microsoft Dynamics NAV 2013 R2 introduced a new format for the product Help. As a result, the language package installers for the Microsoft Dynamics NAV Windows client and Microsoft Dynamics NAV Server do not include .chm and .hh files for the application Help. Instead, you must translate the .html files that are included in the **en** folder on the Microsoft Dynamics NAV Help Server that you install as part of the W1 installation process, and you must make the translated Help files available in your Help Server deployment. For example, if you provide a Catalan localization of Microsoft Dynamics NAV, the Help Server must contain a **ca-ES** folder with the Help in Catalan.

You can deploy new translated Help content to your Help Server website by copying the translated files to the relevant locale-specific folder. For more information, see [Configuring Microsoft Dynamics NAV Help Server](#) in the MSDN Library.

New resource assemblies

Additional resource assemblies have been added since Microsoft Dynamics NAV 2013. These assemblies cover new functionality that has been introduced since Microsoft Dynamics NAV 2013.

Assembly name	Description
Microsoft.Dynamics.Nav.DocumentService.Types.dll	Service
Microsoft.Dynamics.Nav.ExcelAddin.dll	Windows client
Microsoft.Dynamics.Nav.OpenXml.dll	Windows client, Service

Changes from the Microsoft Dynamics NAV 2015 Partner Translation Toolkit

Changes to Client Add-ins table

The name of the table for storing references to localization assemblies has changed from **Client Add-ins** to simply **Add-ins**. Also the usage has changed because it may now contain references as well as packages assemblies for server side .Net Add-ins. All references to this table in the toolset have been updated accordingly. Also, a new set of PowerShell cmdlets have been added to the Microsoft Dynamics NAV 2017 Management Shell in order to maintain the content of this table from a PowerShell environment.

Changes resource assemblies

No new resource assemblies have been added since Microsoft Dynamics NAV 2015. One resource assembly was removed from the Web Client.

Assembly name	Description
Microsoft.Dynamics.Framework.UI.Web.WebClient.dll	Web client

Project - Translate Existing Resource Files to the New Language

Unpack the PLLP Translation Toolkit

After you have downloaded the PLLP Translation toolkit, unpack it to a top-level directory on your computer. This eliminates a limitation in the length of the file name and directory name within Visual Studio, which otherwise will be reached when compiling.

Install the WiX Toolset

The Visual Studio solution contains three projects, which will output the language pack installers for the Microsoft Dynamics NAV 2017 Server, Windows Client, and Web Client.

Building these projects requires that you have the WiX Toolset version 3.8 or 3.9 is installed on the computer. The WiX Toolset can be downloaded and installed from here: <http://wix.codeplex.com/>

Install the WiX Toolset according to the instructions. You can open the Visual Studio solution without errors when the WiX Toolset has been installed.

Identify Visual Studio Solution in the Partner Translation pack

The Partner Translation pack contains the Visual Studio solution translation projects PartnerTranslations.sln for the following target Microsoft Dynamics NAV libraries.

Target library name	Usage
Microsoft.Dynamics.Framework.UI.dll	All clients
Microsoft.Dynamics.Framework.UI.UX2006.dll	All clients
Microsoft.Dynamics.Framework.UI.UX2006.WinForms.dll	Windows client
Microsoft.Dynamics.Framework.UI.Web.dll	Web client
Microsoft.Dynamics.Framework.UI.WebBase.dll	Web client
Microsoft.Dynamics.Framework.UI.Windows.dll	Windows client
Microsoft.Dynamics.Framework.UI.WinForms.dll	Windows client
Microsoft.Dynamics.Framework.UI.WinForms.Controls.dll	Windows client
Microsoft.Dynamics.Framework.UI.WinForms.DataVisualization.dll	Windows client
Microsoft.Dynamics.Framework.UI.WinForms.DataVisualization.Timeline.dll	Windows client
Microsoft.Dynamics.Nav.Client.DynamicsOnlineConnect.dll	Windows client
Microsoft.Dynamics.Nav.Client.TimelineVisualization.dll	Windows client
Microsoft.Dynamics.Nav.Client.WebClient.dll	Web client
Microsoft.Dynamics.Nav.Client.WebCommon.dll	Web client
Microsoft.Dynamics.Nav.DocumentService.Types.dll	Service
Microsoft.Dynamics.Nav.ExcelAddin.dll	Windows client
Microsoft.Dynamics.Nav.Language.dll	All clients, Service
Microsoft.Dynamics.Nav.OpenXml.dll	Windows client, Service

Each project contains the platform resource files (.resx) for all languages shipped by Microsoft.

The Visual Studio solution file, PartnerTranslations.sln, contains a project for each of the platform assemblies that contain translatable resources. In order to create a custom translation, you add the respective resource files with custom translations to the respective project.

Creating Custom Platform Translations

Customize strong name signature

It is necessary to apply a strong name signature to each of the assemblies for which a custom partner translation will be created. The strong name signature is defined in the **Signing** tab of the **Project Designer** properties in Visual Studio for each respective project.

The projects in the template solution are already configured with a strong name signature, which can be used for testing. The public key token for this signature is: **15e5b2499d4875c1**. This signature is kept in one single signature file, **SampleStrongName.snk**, which has been linked into all projects.

The public key token of the strong name signature is needed to register the custom translations so that they can be picked up at run time.

Create and add language resource files

In order to generate satellite assemblies, each of the projects in the Visual Studio solution, PartnerTranslations.sln, must contain the respective resource files in the language for the satellite assembly culture.

The resource files can be found in every project directory. These files contain the platform translations that Microsoft provides for the various countries/regions. These are provided in order to give a good start for creating a custom translation from an existing translation.

In order to create a new translation, for example, for Catalan, ca-ES, a translator could take the following manual steps (below is a description of a tool that automates this process):

1. Select a language to translate from and identify the respective .resx file that is shipped with the translation pack. For example, for Spanish this would be the .resx files that contain the term "es-ES" in the file name.
2. Use a translation tool to open the .resx file. The translation result must be saved in a file that is renamed according to the target language, for example, Common.Language.Lang.ca-ES.resx for Catalan.

Important information when translating NAV resource strings

When translating strings, the person translating must make sure to keep format instructions in their respective translations (for example, "{0}"), and for certain strings which represent an enumeration, to provide the same separator between the translated term as in the original resource file. This is the case for the resource string named BooleanTypeOptions in the resource file Common.Language.Lang.[...].resx, which must preserve the comma between the translated terms for **No** and **Yes** in the target translation.

Without the respective translation of the application objects, the client might report an error on startup. In that situation, try to keep the original non-translated string for the string resource BooleanTypeOptions.

Using the ResX Translator tool in order to automatically translate all resources

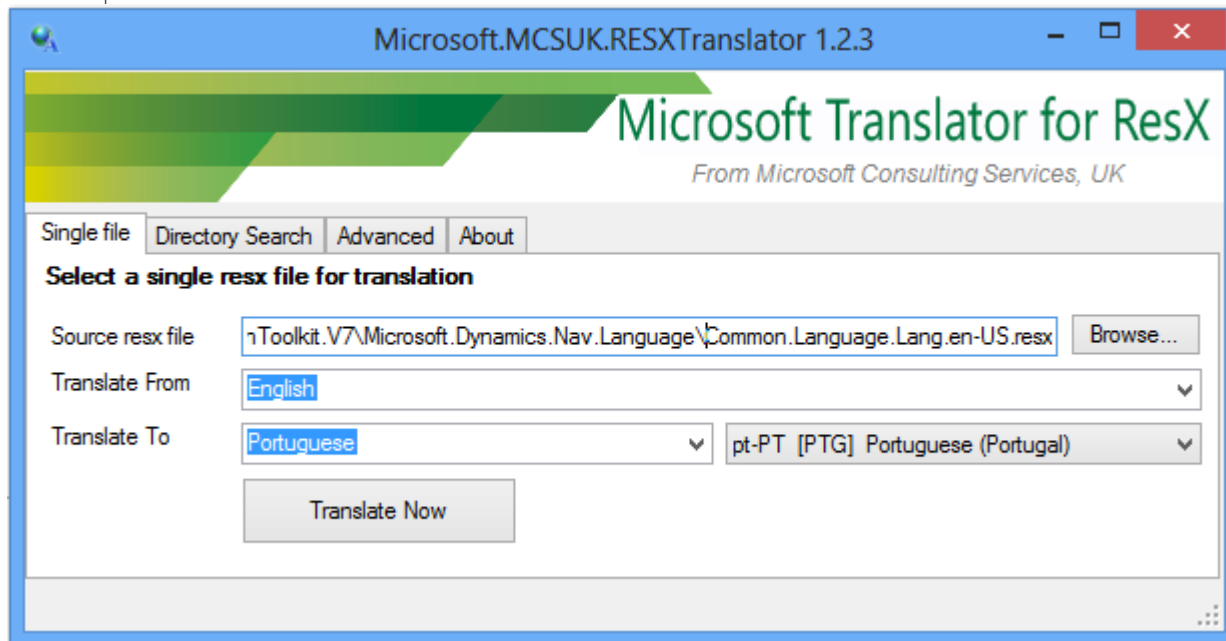
In order to ease the process of translating, translation professionals typically use a toolset of their choice that often includes targeted dictionaries, review tools, and version management. For a quick evaluation, you can use an easy translation tool such as ResX Translator, Microsoft.MCSUK.RESXTranslator.exe, which is based on the Microsoft Translator service on Azure. This tool not only creates a translation of an individual resource file from the selected source language to a selected target language, it also translates all resource files for all assembly projects to a target language in one step. By using this tool, you can create a new platform translation in a few seconds.

When using this tool, keep in mind that many translations will not be correct in the context where they are used in the Microsoft Dynamics NAV platform. The translation result will need a major review and correction by a professional translator with a background in business software, before the translated files can be used in a business environment.

Creating an account for Microsoft Translator

There is a broad set of translation tools available on the market that can help in the development process for translations. A simple tool for prototyping and experimenting with creating platform translations is the Microsoft Translator-based tool Microsoft Translator for ResX, which is available on CodePlex. For more information, see [Microsoft Translator for ResX](#).

Recommendation: You should not use translations of resources created by any translation service in a production version of a Microsoft Dynamics NAV solution. Instead, we recommend that you use translation professionals and domain experts to deliver and review a translation.



Microsoft Translator is a service on the Windows Azure Marketplace that translates text between many languages. This is a billed service that comes with many different pricing plans. For example, one plan allows the translation of 2,000,000 characters per month for free.

The creation of the access credentials on Windows Azure Marketplace is not simple. The requirements for billing lead to a more complex and subscription model. For a typical company, there will be one billing account created for multiple service subscriptions and applications.

The following section walks you through the process of creating an account on the Microsoft Data Market and creates an application on top of the Microsoft Translator service. This application delivers a client ID and client secret, which are needed in order to use the ResX Translation tool.

1. In order to use the ResX Translation tool for Microsoft Translator you need to get an account on the Azure Data Market with your Microsoft Live ID.
2. After logging into the Azure Data Market (ADM), subscribe to the [Microsoft Translator Service](#).

My Data

Details		Type	Status	
Microsoft Translator	Microsoft Translator	Subscription	Active	Use
	Microsoft Translator (Started on 4/13/2012)	2,000,000 Characters/month	1,812,199 Characters remaining	Show Terms Cancel

Creating Custom Platform Translations

Microsoft Dynamics NAV 2017

- While you are still logged into the ADM, you have to create an application on top of this service in order to use the service in the ResX Translator tool. Go the [developer application registration](https://datamarket.azure.com/developer/applications/) at <https://datamarket.azure.com/developer/applications/>. Choose the **Register** button to create a new application.
- Enter your information and choose the **Create** button.

Register your application

* Client ID

* Name

* Client secret

* Redirect URI

☐ Enable subdomain access

Description

* Required fields

◀ Cancel **CREATE**

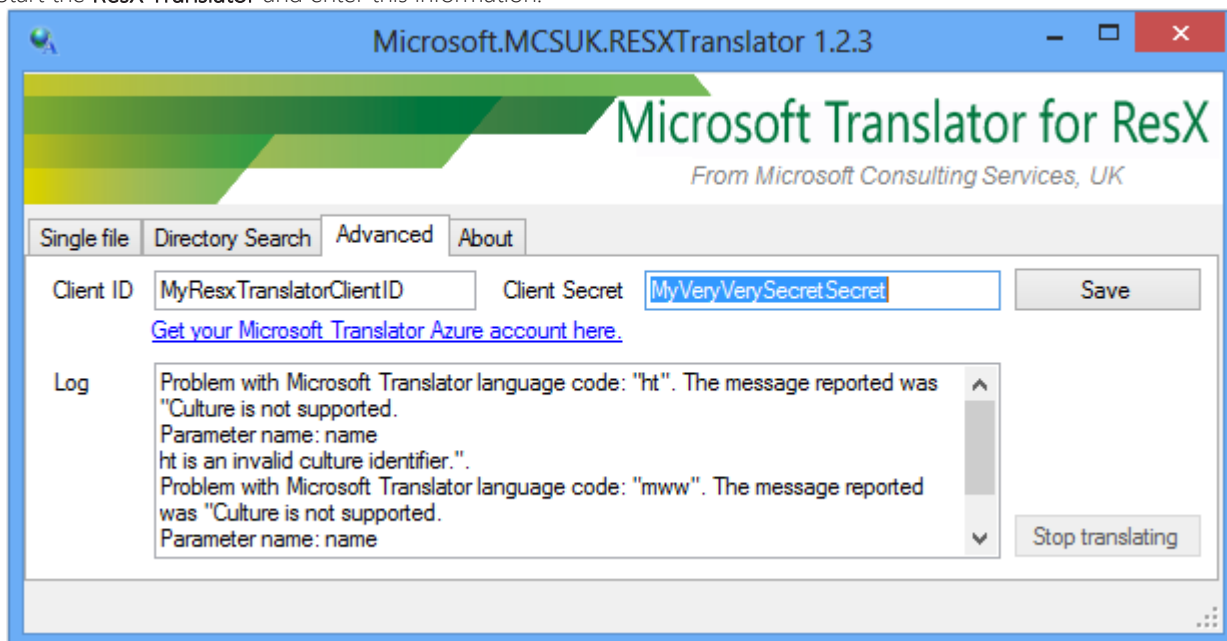
This will create your application for ResX translation.

Registered Applications

Client ID	Name	Description	Status
MyResxTranslation	MyResxTranslation		Active Edit Delete

Make note of the values that you have entered in the **Client ID** and the **Client Secret** fields from the application registration. You will have to enter them into the ResX Translator.

- Start the **ResX Translator** and enter this information.



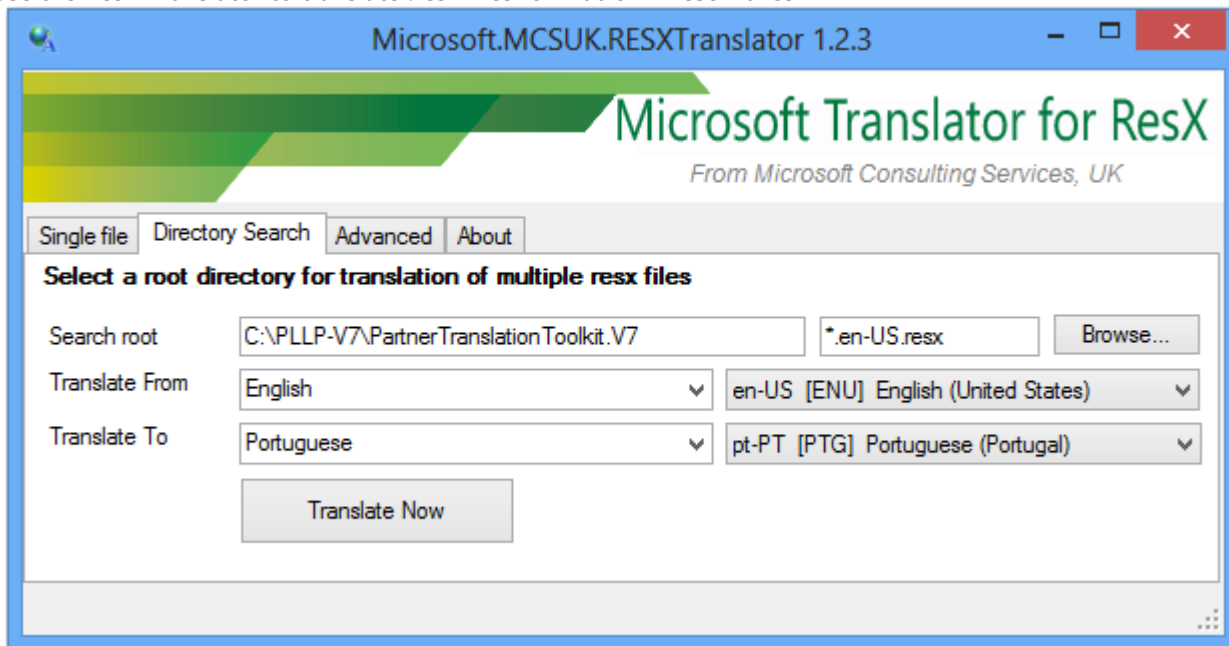
- Choose the **Save** button. After you choose **Save**, the tool will try to connect the service and load all available languages. Some error messages are expected for languages that cannot be mapped for software localization purposes.

Creating Custom Platform Translations

Microsoft Dynamics NAV 2017

You might see error messages in the Log, as shown in the screenshot above. These error messages are related to services languages which the tool cannot yet map to languages. These messages can be ignored.

Use the ResX Translator to translate .resx files for Platform Assemblies



To translate the resource files, perform the following steps in the ResX Translator:

1. Switch to the **Directory Search** tab.
2. Set **Search root** to the folder where you have the original resource files.
3. Set the **Translate From** field to **English** / **en-US**.
4. Set the **Translate To** field to **Catalan** / **ca-ES**.
5. Choose the **Translate Now** button.

A message will appear after the translation is complete.

Add translated resource files to Visual Studio solution

1. Open the Visual Studio PartnerTranslations.sln solution in the translation kit.
2. Add all your translated resource files to the respective projects in the solution. In the sample, these files all have the .ca-ES.resx resource extension and are found in the same folder of the original resource files that were translated.
Note: For PLLP translations, the satellite for add-ins should be placed in the satellite directories for the platform, and not in satellite directories for the add-in assemblies.

Build – Create new language pack installers

In addition to the satellite assemblies, the fin.stx files, and the fin.etx files are also picked up for deployment by the language pack installers. Visual Studio does not translate these files. Instead they must be placed in the folder structure where they will be picked up during the build process.

Copy text resource files

Typically, a partner who created a custom platform translation for Microsoft Dynamics NAV has had to translate the strings in the fin.stx file, get the translation file *sealed* by Microsoft, and distribute it to customer installations. This is still a requirement for Microsoft Dynamics NAV 2017, even though the overwhelming majority of translated strings are read by the platform from the resources in the .NET assemblies and the respective satellite assemblies.

The PLLP translation kit has a folder that contains all .stx and .etx files as they are shipped by Microsoft in the various countries/regions and languages. The files are shipped in order to ease the translation into a new language. These files, along with the Cronus.flf file, must be copied to the Payload\Local text files folder. Also in this case, a set of files have been added to make sure that the solution builds.

Change installation properties

The properties of the installation determine which folders are created during the installation of the language packs and what files are placed in the folders. The properties can be found in the Locals.wxi file located in the root directory of the translation kit. Open the Locals.wxi file and change the following properties:

- **CountryRegionCode**, to one of the country/region codes in Globals.wxi or create a new.
- **ModuleLanguage**, to the language of your choice.
- **LocalLangFolder**, to the .Net culture of the satellite assemblies.
- **LocalIsoFolder**, to the three letter ISO code for your language.

Sample of Locals.wxi (partial):

```
<?xml version="1.0" encoding="utf-8"?>
<Include xmlns="http://schemas.microsoft.com/wix/2006/wi">

    <!-- Installation properties related to the culture of the generated
    installer -->

    <!-- Set to the build number of Microsoft Dynamics NAV 2017, that you are
    building language packs for. -->
    <?define BuildVersion = "10.0.42572.0"?>

    <!-- Set to the correct CountryRegionCode from Globals.wxi if one has
    already been assigned to your language packs or create new if this is not
    the case. -->
    <?define CountryRegionCode = "ACF7"?>

    <!-- Name of the language in English -->
    <?define ModuleLanguage = "Catalan"?>

    <?define ResourceAssemblies = "..\Payload\Resource assemblies"?>

    <!-- .Net culture of the resource assemblies i.e. "ca-ES" -->
    <?define LocalLangFolder = "ca-ES"?>

    <!-- Three letter ISO code of the language i.e. "CAT" -->
    <?define LocalIsoFolder = "CAT"?>
```

You do not need to change the rest of the properties in the file.

Build the solution

To build the PartnerTranslation solution from inside Visual Studio, choose **Build Solution** from the **Build** menu. This will pick up all translated artifacts, and create three installer files (.msi):

- Server.msi – Located in the Server\Bin\Debug folder.
- RTC.msi – Located in the RTC\Bin\Debug folder.
- WebClient.msi – Located in the WebClient\Bin\Debug folder.

Warnings about obsolete constructor can be ignored, such as the following:

“System.Reflection.AssemblyFlagsAttribute.AssemblyFlagsAttribute(int) is obsolete: 'This constructor has been deprecated’

.

Install – Add your language pack to the W1 version

If have not already done so, install the W1 version of Microsoft Dynamics NAV 2017 from the product media.

Install the new language lack

Next, install the new language pack on the same computer. To do this, select and click each of the three .msi files listed below, one after the other. The order is not significant:

- Server\Bin\Debug\Server.msi
- RTC\Bin\Debug\RTC.msi
- WebClient\Bin\Debug\WebClient.msi

Each installer quickly shows a dialog, which disappears when the installer has completed.

The language pack is now installed, but you must make some final configuration.

Register - Set trusted assemblies and select new language

Manually register root assemblies with strong name key token in the Add-ins table

Microsoft Dynamics NAV cannot load untrusted assemblies. To enable your custom resource assemblies as trusted, you must register the strong name key in the database:

1. Open the Microsoft Dynamics NAV Development Environment.
2. On the **View** menu, choose **Object Designer**, and then choose **Table**.
3. Navigate to the **Add-in** table, table 2000000069, and then choose **Run**.
4. For each assembly, enter the name and the public key token used when signing the files. The assemblies to add can be found in the **Output** folder, or can be seen in the table in the [PROJECT section](#) of this document.

To find the public key token, use the **Strong Name** tool (**sn.exe**) with the **-T** parameter. At the command prompt, enter the following command:

sn.exe -T <assembly>

For example, enter the following command to find the public key token for the Microsoft.Dynamics.Framework.UI.dll file:

sn.exe -T Microsoft.Dynamics.Framework.UI.dll.

The sample assemblies that are included in the Partner Translation solution have the following signature:

15e5b2499d4875c1.

Finally, you need to update the Languages page:

1. In the Object Designer, choose Pages.
2. Navigate to the Languages page, page 9, and then choose Run.
3. Add the Catalan (CAT) language to it. The Windows code required can be looked up here:
<http://msdn.microsoft.com/en-us/library/dd318693.aspx>

Optionally, you can use PowerShell to register a strong name key for the custom resource assemblies.

New-NAVAddIn <instance> -AddInName <assembly> -PublicKeyToken <publickey> -Category LanguageResource

Register root assemblies with the Partner Translation Tools

Import the PartnerTranslationTools.fob file into the Microsoft NAV Development Environment.

If you run the Partner Translation Tools page, page 58886, the result should yield a setup such as what is shown in the following illustration.

The screenshot shows the 'Edit - Partner Translation Tools' window with the following data:

Custom Translation List (Filtered)

Base Assembly Name	Public Key Token	Version	Category	Description
Microsoft.Dynamics.Framework.UI	31bf3856ad364e35		Language Resource	
Microsoft.Dynamics.Framework.UI.UX2006	31bf3856ad364e35		Language Resource	
Microsoft.Dynamics.Framework.UI.UX2006.WinForms	31bf3856ad364e35		Language Resource	
Microsoft.Dynamics.Framework.UI.Web	31bf3856ad364e35		Language Resource	
Microsoft.Dynamics.Framework.UI.WebBase	31bf3856ad364e35		Language Resource	
Microsoft.Dynamics.Framework.UI.Windows	31bf3856ad364e35		Language Resource	
Microsoft.Dynamics.Framework.UI.WinForms	31bf3856ad364e35		Language Resource	

Language List

Code	Name	Windows Language ID	Windows Language Name
BGR	Bulgarian	1026	Bulgarian (Bulgaria)
CSY	Czech	1029	Czech (Czech Republic)
DAN	Danish	1030	Danish (Denmark)
DEA	German (Austrian)	3079	German (Austria)
DES	German (Swiss)	2055	German (Switzerland)
DEU	German	1031	German (Germany)
ELL	Greek	1032	Greek (Greece)
ENA	English (Australian)	3081	English (Australia)

Windows Languages List

Language ID	Primary Language ID	Name	Abbreviated Name	Ena...	Glo... Ena...	Form Ena...	Rep... Ena...	Data... Ena...	XML... Ena...	Primary CodePage	STX Fil...
1025	1025	Arabic (Saudi Arabia)	ARA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	720	<input type="checkbox"/>
1026	1026	Bulgarian (Bulgaria)	BGR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	866	<input type="checkbox"/>
1027	1027	Catalan (Catalan)	CAT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	850	<input type="checkbox"/>
1028	1028	Chinese (Traditional, Taiwan)	CHT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	950	<input type="checkbox"/>
1029	1029	Czech (Czech Republic)	CSY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	852	<input type="checkbox"/>
1030	1030	Danish (Denmark)	DAN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	850	<input checked="" type="checkbox"/>
1031	1031	German (Germany)	DEU	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	850	<input checked="" type="checkbox"/>

The **New custom translation** action opens a token editor where you can enter the public key token of the strong name signature of the PLLP translation libraries. Enter the key that you want to register a PLLP for. After closing the Token Editor, respective entries will be generated for the root names of the typical libraries in Microsoft Dynamics NAV.

Note: PLLP translations for other add-ins have to be registered manually in the **Add-In** table, table 2000000069, or in the **Custom Translation List** page, page 58887.

Restart the Microsoft Dynamics NAV Server service

To enable the newly added language, you must restart the Microsoft Dynamics NAV Server service:

1. Choose the **Start** button, point to **Administrative Tools**, and then choose **Services**.
2. In the **Services** window, right-click **Microsoft Dynamics NAV Server**, and on the shortcut menu, choose **Restart**.

Connect to the Microsoft Dynamics NAV Server and select the new language

After you have restarted the **Microsoft Dynamics NAV Server** service, you can then open the **Microsoft Dynamics NAV Client** and select the new language:

1. Choose the **Start** button, and then choose **Microsoft Dynamics NAV**.
2. Choose **Microsoft Dynamics NAV**, and then choose **Select Language**.
3. In the **Language Selection** field, select the new language, Catalan, and then choose the **OK** button.

Tips And Tricks

Add .resx files of a certain language to the satellite assembly projects

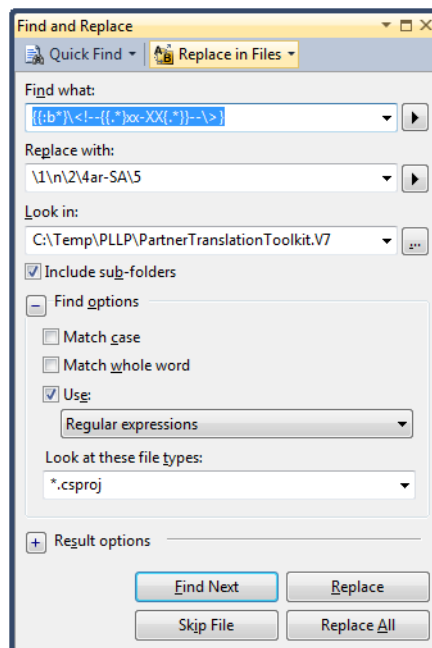
Each of the projects for the satellite assemblies in the PartnerTranslations.sln solution contains the references to the .resx files that should be used. If you open the .csproj file in text editor view in Visual Studio, you will be able to find XML instructions, as shown below:

```
<ItemGroup>
  <!--<EmbeddedResource Include="Resources.xx-XX.resX" />-->
  <EmbeddedResource Include="Resources.ca-ES.resX" />
</ItemGroup>
```

This XML instruction in the project file will compile a satellite assembly for Catalan in the ca-ES subdirectory.

In order to compile the satellite assembly for another culture, for example, Arabic, ar-SA, after you create all translated .resx files with the ResX Translator tool, you may use a regular expression based text modification of all .csproj files. In the **Find and Replace** dialog box in Visual Studio, you can add the new .resx files to all .csproj files with the following regular expressions:

- In the **Find What:** field:
`{ (:b*)\<!--({ .*}xx-XX{ .*})--> ,`
- In the **Replace With:** field:
`\1\n\2\4ar-SA\5.`



Validate correctly formatted strings in the translations

A number of the strings in the resources contain format elements that must also be found in your translated strings.

Microsoft.Dynamics.NAV.Language, Common.Language.Lang, BooleanTypeOptions

The most prominent string is the resource string **BooleanTypeOptions** in the Microsoft.Dynamics.NAV.Language library and Common.Language.Lang resource set. This string must use a comma (,) as the delimiter of the translated strings **No** and **Yes**. For example, in Arabic:

BooleanTypeOptions لا, نعم

If this comma is replaced by a different respective local character, the client will crash after startup and an error message similar to the one that follows can be found in the error log:

The language resource 'BooleanTypeOptions' does not follow the format of 'No,Yes'. The words No and Yes may be translated to local language.

Ampersands in strings

Another source of error is the placement of the ampersand symbol (&), which appears in many strings as the indicator of its following character.

Troubleshooting

The following issues may occur when you add an unsupported language to Microsoft Dynamics NAV.

Error	Possible cause	Solution
Text values are only shown in English after you change the language	The names for assemblies or folders are incorrect.	Ensure that the names for the assemblies and folders on the computers running the Microsoft Dynamics NAV Client and Server are correct.
	The assemblies are not signed or the signed assemblies are not registered in the Add-in table, table 2000000069.	Sign the assemblies and enter the public key tokens to the Add-in table.
Resources appear truncated in the user interface at run time.	Strings are too long.	Use shorter strings.

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.

United States and Canada toll free: (888) 477-7989 Worldwide: (1) (701) 281-6500 www.microsoft.com/dynamics

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, this document should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

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.

© 2016 Microsoft. All rights reserved. Microsoft, Microsoft Dynamics and the Microsoft Dynamics logo are trademarks of the Microsoft group of companies.