

## Introduction

TSW Localization is a feature of TSW that allows the TSW application to be available in several languages. This document presents the steps involved in translating (localizing) a TSW language resource library. The localization process consists of five steps:

1. The satellite resource DLLs translation
2. The system language files translation
3. The 3PII database file translation
4. The DACR mapping files translation
5. The runtime online help translation

**Note:** Read through this entire document before executing the procedure.

## System requirements

Table 1 lists the software required to translate the TSW Resource, Language and Help files.

**Table 1: Required TSW software components**

Parameter	Requirements
Operating system	Windows 10 or Windows 11
Online help authoring tool	OxygenXML Editor 23.1 or higher
Screen resolution for TSW	1024x768 or higher (1280x1024 is recommended)

## Satellite Resource DLLs translation

This section deals with translating all the text strings that are embedded inside the Resource DLLs. The full process consists of the following steps:

1. Preparing the .txt file
2. Translating the .txt file
3. Generating the Translated Satellite Resource DLLs

**Note:** The working directory for all steps in this section is `WFW\ResourceDLLs` inside the Localization kit.

### Preparing the .txt file

To prepare the .txt file prior to translation, rename the `tswui.LanguageCode.txt` file to a name TSW can recognize.

**Note:** The file `tswui.LanguageCode.txt` is located in `WFW\ResourceDLLs\en-US` of the Localization Kit.

Replace **LanguageCode** with the required language code, found at [http://msdn.microsoft.com/en-us/library/ms533052\(vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533052(vs.85).aspx). For example, for Czech, the file is `tswui.cs.txt` and for French (Canada), it is `tswui.fr-CA.txt`. The renamed file is now ready for translating.

**Note:**

1. In the sections that follow, `tswui.fr-CA.txt` is used as an example.
2. The file `tswui.txt` in `WFW\ResourceDLLs\en-US` is there for backup restoral or reference only.

**Important:** If the `tswui.LanguageCode.txt` file is moved to another directory, the `rules.txt` file must be either copied to the same directory or its other location specified inside the Localize application by clicking on the Tools menu and then clicking on Options. See [Translating the .txt file](#) for information regarding the Localize application.

### Translating the .txt file

To translate the `tswui.fr-CA.txt`, use the Localize application. It simplifies the translation by supporting rules, preserving the format of the file and strings, and by allowing easy sorting and filtering.

**Note:**

- Install the TSW and Localize software on the same PC prior to running the Localize application.
- If the Localize application is not used to translate the .txt file, any other text editor can be used, as long as it preserves the Unicode format while saving, and has the capability to show multi-line text in a single cell. If the editor modifies the content, ensure the saved values are in proper form. For example, for assembly `tswui.g.en-US.resources.eulatext.baml` and key `Run_9:System.Windows.Documents.Run.$Content`, Excel reformats content to **1**. After saving, change it back to **1**.

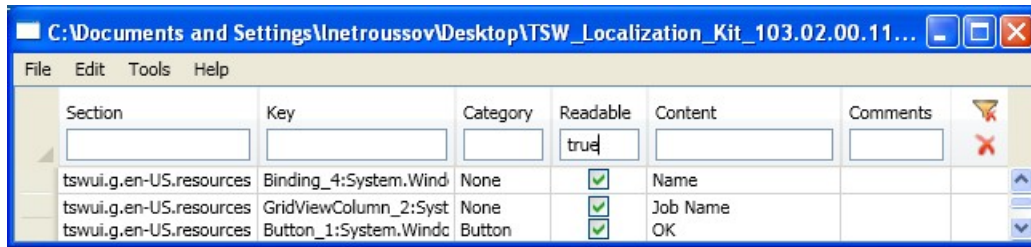
To translate the file, complete the following steps:

1. Double-click the `Localize.exe` file under `WFW\Localize` of the Localization kit to launch the Localize application.
2. Inside the application, click the **File** menu and then click **Open**. Inside the **Open** window that appears, browse to the location of the `tswui.fr-CA.txt` file and double-click to open it.
3. Click the **Filter** icon at the top right-hand side of the Localize application.
4. In the new row that appears, type **true** under the **Readable** column. See [Figure 1](#). Click away or wait for a short period. All rows that do not have a check mark in the **Readable** column are hidden.

**Note:** Unless specifically instructed, it is safe to hide rows where **Readable** is **false**. The filter may be used for any of the columns to find any existing text. See the last note of section for details on where these special cases can be found.



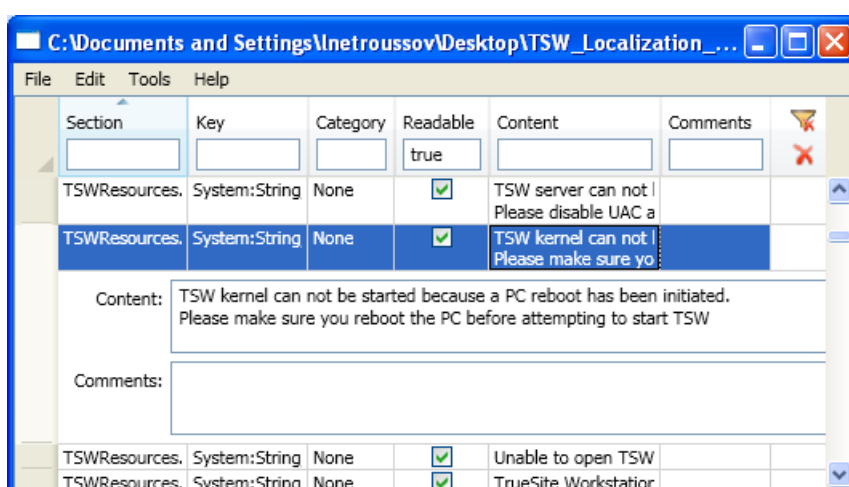
Figure 1: Applying the filter



5. Translate the strings located under the **Content** column. The **Comments** column contains tips that may be helpful during the translation. See [Table 2](#) for translation guidelines. Save the `tswui.fr-CA.txt` file periodically during the translation.

**Note:** The `rules.txt` file determines which rows are hidden in the Localize application, and contains comments for specific rows. These comments appear in the **Comments** column.

Table 2: Guidelines for string translation

Task	Guidelines
Sort by the <b>Section</b> column	<ul style="list-style-type: none"> <li>Click a column's header to sort in an ascending or descending order. The second click the same column's header reverses the sort order.</li> <li>Sort or filter by the <b>Section</b> column so that each dialog can be completed one at a time. For example, translate the <b>About</b> Node dialog by filtering the section for <code>tswui.g.en-US.resources:aboutnodedlg.bam1</code>. Limit the number of rows to that dialog by typing <b>aboutnodedlg</b> into the filter field under the <b>Section</b> column.</li> <li>Filters work on parts of strings and support * and ? wildcards.</li> </ul>
What to translate	<p>After filtering the <b>Readable</b> column and choosing a specific <b>Section</b>, filter the <b>Key</b> column for the following strings and translate the <b>Content</b> strings:</p> <ul style="list-style-type: none"> <li>.Content</li> <li>.Title</li> <li>.Header</li> <li>.ToolTip</li> <li>.InputGestureText</li> <li>.OKButtonText</li> <li>.ButtonText</li> <li>.ResourceString.Value</li> <li>.Text</li> <li>.ErrorContent</li> <li>.\$Content</li> </ul> <p><b>Important:</b> Modify words with a pound sign (#) extremely carefully when translating strings associated with <code>.\$Content</code>. For example, in the case of <code>#Grid_1</code>; if the Content string does not have any other human-readable text, you do not need to translate it. A notable exception is Section: reportview, Key: TextBlock_2.</p>
How to translate	<ul style="list-style-type: none"> <li>Some Content and Comments strings are too long to be displayed fully. The Localize application repeats them in a <b>Details</b> pane. See <a href="#">Figure 2</a>. The Content string can be modified by selecting a row and clicking the cell in the <b>Content</b> column. The Content and the Comments strings in the details pane are for reference only.</li> </ul> <p style="text-align: center;"><b>Figure 2: Details pane and editing content</b></p> 

**Table 2: Guidelines for string translation**

Task	Guidelines
How to Translate	<ul style="list-style-type: none"> <li>• If the original text is contained in multiple lines, the multiple-line format has to be maintained according to the original text. For example, keep two lines if the original cell had two lines.</li> <li>• Maintain {#}, where # is a digit, usually 0, 1, 2, etc. These are placeholders for text to be added dynamically by the code. For example, There are {0} seconds remaining.</li> <li>• The underscore denotes the character that follows it as a shortcut key when navigating through the application using the keyboard. For better keyboard navigation, use underscores. Avoid any duplicate shortcut characters within the same Section (for example, Can't have Fi_le and Too_ls). Underscored characters do not need to be the same characters as in English.</li> <li>• The XML escape sequences need to be used. The following are examples of escape sequences and what they correspond to:                         <ul style="list-style-type: none"> <li>- &amp;amp; &amp;</li> <li>- &amp;quot; "</li> <li>- &amp;gt; &gt;</li> <li>- &amp;lt; &lt;</li> <li>- &amp;apos; '</li> </ul> </li> <li>• For best email results, the strings used for email content in the emaildictionary.baml section have to be the same length. Starting with String_7 these strings have a "..." at the end of them. For example, after translating, locate strings that end with "..." and count the longest one. Make all of the strings the same length by adding extra dots where necessary.</li> <li>• Clicking on another row "confirms" a translation. Pressing <b>Esc</b> once reverts to the previous version of the string. The filter settings cannot be changed before string translation are confirmed or canceled.</li> </ul>
If the Localize application is not used	<p>Translating strings that do not match the, <b>Readable</b>= "false", and filters in the <b>What to translate</b> row can produce undesirable results during runtime.</p> <ul style="list-style-type: none"> <li>• Never delete or add any new rows.</li> <li>• Never modify any column but <b>Content</b> column.</li> <li>• Never modify any empty cell inside <b>Content</b> column.</li> <li>• Read and follow comments in the rules.txt file.</li> </ul>

## Generating the translated Satellite Resource DLLs

After the `tswui.fr-CA.txt` file is translated and saved, the last step is to generate two Resource DLLs for the translated language. Follow these steps to generate the DLLs:

**Note:** If the Localize application is not used, follow the steps below to use the file `GenerateSatelliteDLLs.bat` file to obtain the two Resource DLL files.

1. Launch the Windows Command Prompt and browse to the directory where the `GenerateSatelliteDLLs.bat` file is located.

2. Execute the command `GenerateSatelliteDLLs <language code>`. The two DLL files are generated as follows:

```
- <language code>\tswui.resources.dll
- <language code>\TSWResources.resources.dll
```

For example, executing `GenerateSatelliteDLLs fr-CA` would produce the two files below:

```
- fr-CA\tswui.resources.dll
- fr-CA\TSWResources.resources.dll
```

The `fr-CA` is the folder located under the same directory as the `.bat` file. Copy this folder to the locations mentioned in steps 5 and 6 in [Generating the translated Satellite Resource DLLs](#).

1. Copy the `LocBam1.exe` file from the ResourceDLLs folder in the Localization Kit, into `C:\Autocall\TrueSite\TSW\Client`.

2. With the `tswui.fr-CA.txt` still open in the Localize application, click the **Tools** menu and then click **Options**.

3. In the **Options** dialog box that appears, select the new location of the `LocBam1.exe` file and click **OK** to close the window.

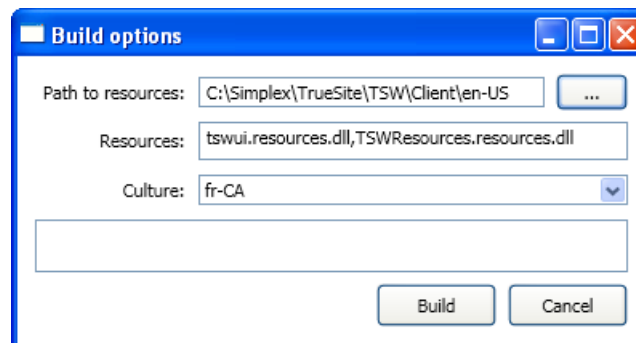
4. Back in the main screen, click the **Tools** menu and then click **Build**.

5. In the **Build Options** dialog box (Figure 3), select the Path to Resources (the `en-US` folder under `C:\Autocall\TrueSite\TSW\Client`) and the Culture (`fr-CA`). Click the **Build** button to start building the DLLs.

**Note:** If the **Build Options** dialog box displays the message **Cannot find LocBam1.exe**, the path to the `LocBam1.exe` file needs to be changed to the one in step 1.

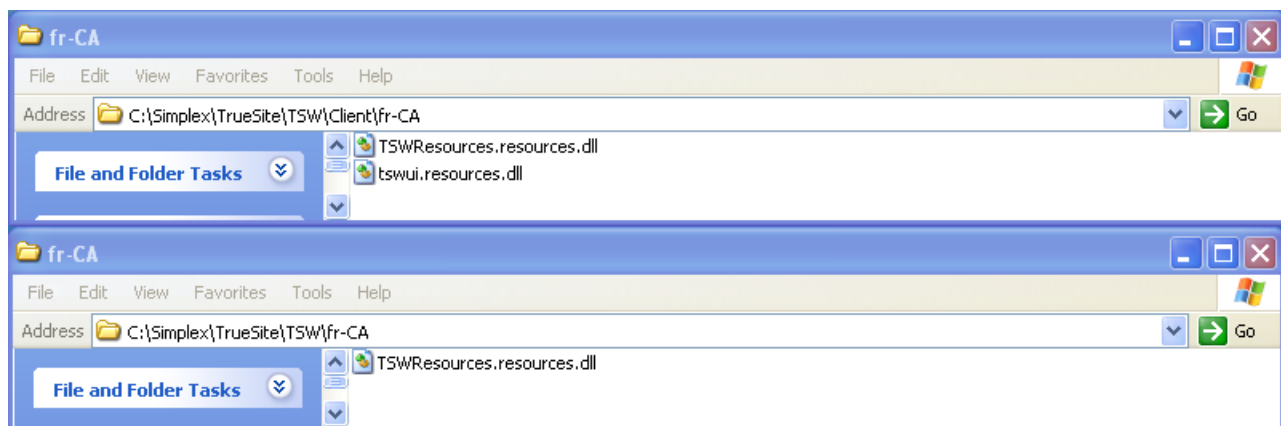
If there were no errors during the build, the **Build Options** dialog box closes and the two Resource DLLs (`tswui.resources.dll` and `TSWResources.resources.dll`) are generated inside the `fr-CA` folder which is located inside `C:\Autocall\TrueSite\TSW\Client`. If there was an error, a `LocBam1` file is created that can be used to diagnose and fix the build problem.

Figure 3: Build options for the .TXT File



6. Copy the `fr-CA` folder, paste it into `C:\Autocall\TrueSite\TSW` and delete the `tswui.resources.dll` from the folder pasted in the new location (Figure 4).

Figure 4: Target client and server files



## System language files translation

This section deals with translating the language files that TSW uses. The language files translation is a three-step process:

1. Setting the Language
2. Translating the `string.dat` file
3. Modifying the `sos.ini` file

### Note:

- `.dat` files must stay in ANSI format - not Unicode.
- If you use non Western character encoding, for example Arabic, set the Windows System ANSI Code page before you start the **Localize** app:

To change your system's locale, complete the following steps:

1. Access the Windows **Control Panel**.
2. Select **Region and Language**.
3. For Windows 10, click **Clock, Language, and Region** first; and then click **Region**.
4. Click the **Administrative** tab.
5. Under **Language for non-Unicode** programs, click **Change System Locale**.
6. Choose the locale for your required language.
7. Click **OK**.
8. Restart Windows, start the Localize app, and translate the `.dat` file.

## Setting the language

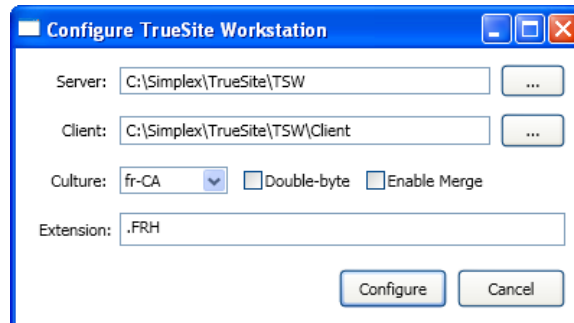
Update the TSW configuration files so the TSW can use the translated language files. Follow the steps to update the files with and without the Localize Application:

### With the Localize Application

1. Inside the Localize application, click the **Tools** menu and then click **Configure TSW**.
2. In the **Configure TrueSite Workstation** dialog box that appears, select the path to the Server installation directory. The default location is the TSW installation folder `C:\Autocall\TrueSite\TSW`.

**Note:** In the case of a client-only installation, a valid server path is not required.

Figure 5: Configure TrueSite Workstation dialog box



3. Select the path to the Client installation directory. The default location is `C:\Autocall\TrueSite\TSW\Client`.

**Note:** If the error **Cannot find d:\path\to\TSWClient.exe.config** displays, the Client's installation directory is incorrect.

4. Select the **Culture** that matches the one selected for the Resource DLLs (`fr-CA`).

**Note:** For steps 5, 6 and 7, if the `string.dat` file has not been translated yet, leave the **Double-byte** and **Enable Merge** checkboxes unchecked, and enter **.ENG** as the Extension.

5. Check the **Double-byte** checkbox or do not check **Double-byte** checkbox. Refer to section [Translating the String.dat and String.bin files](#) to see which case applies.
6. Put a checkmark in the **Enable Merge** checkbox only when default point labels need to be translated during job build in the TSW Configurator.

**Note:** The setting of the **Enable Merge** checkbox corresponds to the line `[Configurator32] MERGE` inside the `sos.ini` file.

7. Enter the same extension as the one used in the section [Translating the String.dat and String.bin files](#).
8. Click **Configure** to start the TSW configuration file updates.

If the configuration succeeds, the **Configure TrueSite Workstation** dialog box closes. Error messages appear if the configuration fails.

### Without the Localize Application

To update the configuration files manually, use a text editor to access the <applicationSettings> section and change the value of the language entry from the default (en-us) to the required language code (fr-ca) in the following configuration files:

- tswkrnl.exe.config in the TSW folder (C:\Autocall\TrueSite\TSW)
- tswclient.exe.config and tswdesktop.exe.config in the Client folder and TswServerDesktop.exe.config (C:\Autocall\TrueSite\TSW\Client)
- TSWBackupUtility.exe.config in the TSW folder (C:\Autocall\TrueSite\TSW)
- TswFileTransfer.exe.config in the TSW folder (C:\Autocall\TrueSite\TSW)

**Important:** To avoid TSW startup failure, use an editor that does not change the format of the files, for example, Notepad. Do not use Microsoft Word.

<b>Full installation</b>	For a full installation, the language must be set in all .config files. For a client-only installation, there is no tswkrnl.exe.config file and the language must be set in the tswclient.exe.config, tswdesktop.exe.config and TswServerDesktop.exe.config files.
<b>Re-installation or upgrade</b>	On a re-installation or an upgrade: <ul style="list-style-type: none"> <li>• Copy the existing tswkrnl.exe.config to tswkrnl.exe.config.backup.</li> <li>• Copy the existing tswclient.exe.config to tswclient.exe.config.backup.</li> <li>• Copy the existing tswdesktop.exe.config to tswdesktop.exe.config.backup and TswServerDesktop.exe.config to TswServerDesktop.exe.config.backup</li> </ul> If you have an existing translation, verify that these files have the language set properly after the new installation completes.

### Translating the string.dat and string.bin files

This section describes the translation process of the string.dat and string.bin TSW language files. You can locate these files, and BuildLg.exe, inside the WFW folder of the Localization Kit.

Complete the following steps to perform the translation with and without the Localize application:

1. Create a copy of the string.dat file and rename the copy to reflect the translation language. For example, French.dat if the translation is in French.
 

**Note:** The filename must follow the "8.3" convention: an eight-character-maximum filename and a three-character-maximum extension. The file must reside in a folder the path to which contains only "8.3"-named folders. For example, C:\Autocall\TrueSite\TSW is acceptable.
2. Create a copy of the string.bin file and rename the copy to reflect the translated name of the .dat file. For example, string.bin becomes French.bin if the translation is in French.
 

**Important:** If you move French.dat to another directory, copy French.bin, and rules.txt to the same directory.
3. Translate the string.dat file using the Localize application or without it. See Table 3 for guidelines to translate using either method
4. Once the .dat file is translated and saved, generate the language files for the translated language. See Table 4 for steps to generate the files with and without the Localize application.

**Note:** string.dat contains 47 different sections. For translation purposes, 13 of these sections do not require translation as they contain TSW Configurator-specific strings. Each section begins with a line that starts with GROUP\_DEFINE, followed by ten digits and a string that identifies the section. For example, GROUP\_DEFINE 1173748968 "helpeng", where "helpeng" is the language section identifier. The following 13 sections do not require translation: "ale", "color", "ctled", "edit", "hw", "hwdb", "netdb", "odbe", "opt", "prog", "rh", "soserr", "util"

**Table 3: Guidelines for string translation**

With the Localize Application	Without the Localize Application
<ul style="list-style-type: none"> <li>• Typing <b>true</b> under the <b>Readable</b> column hides the rows that do not need to be translated.</li> <li>• The <b>Content</b> column is the only one that needs to be translated.</li> <li>• The <b>Comments</b> column contains additional tips to help with the translation.</li> </ul>	You can use any text editor, for example Notepad, to translate the French.dat. The format of the translated strings is as follows: <pre>STR_DEFLABEL_DP0 //MAX 40 ALARM ACTIVE LIST FULL "LISTE D'ALARME ACTIVE COMPLÈTE"</pre> <p><b>Note:</b> Do not modify the first line of the pair. The second line is the translation. Do not translate lines beginning with GROUP_DEFINE.</p>
<p><b>Note:</b></p> <ol style="list-style-type: none"> <li>1. Do not use tildes. All tildes (~) are safe to remove.</li> <li>2. All formatting codes ("% immediately followed by one or two letters) must be preserved as is. Formatting codes indicate that a given string is replaced/changed when displayed.</li> <li>3. Special characters, including "\n", "\r", "\t" (newline, carriage return, and tab respectively) must be preserved as is.</li> </ol>	

**Table 4: Generating language files with and without the Localize Application**

With the Localize Application	Without the Localize Application
<ol style="list-style-type: none"> <li>Click the <b>Tools</b> menu and then click <b>Build</b>. <b>Note:</b> If the <b>Build</b> menu is disabled, ensure that the path to the <code>Buildlg.exe</code> file is correct by clicking the <b>Tools</b> menu and then <b>Options</b>.</li> <li>Inside the <b>Build Options</b> dialog box that appears choose the output directory for the translated language files. In the current example, the location is <code>C:\Autocall\TrueSite\TSW\Language</code>.</li> <li>Choose the file extension for the translated language files. The file extension format must be <code>.abc</code>. <b>Note:</b> The file extension for the translated language files must correspond to the one set in the <a href="#">Setting the language</a> section. In the current example, it is <code>.FRH</code>.</li> <li>Check the <b>Double-byte character</b> set checkbox if the translation is in a language that requires a double-byte character set, such as Chinese. <b>Note:</b> The setting for the <b>Double-byte character</b> set checkbox must match the setting for the <b>Double-byte</b> checkbox in the <a href="#">Setting the Language</a>.</li> <li>Click <b>Build</b> to generate the translated language files.</li> </ol> <p><b>Note:</b> If the message <code>Cannot find buildlg.exe</code> displays in the <b>Build Options</b>, correct the path to the <code>Buildlg.exe</code> file by clicking the <b>Tools</b> menu and then <b>Options</b>.</p>	<ol style="list-style-type: none"> <li>Launch the Windows Command Prompt and navigate to the directory containing the <code>.dat</code> and <code>.bin</code> files by using appropriate commands.</li> <li>Once you access the appropriate directory, enter the command <code>buildlg string .ext 1</code> (Figure 7). The breakdown of the command is as follows: <b>Buildlg:</b> Corresponds to the build-language-file command. <b>String:</b> Corresponds to the translated <code>filename.dat</code> and <code>filename.bin</code> files. <b>.ext:</b> Corresponds to the file extension of the translated files. For example, <code>.FRH</code> for French. <b>1:</b> Corresponds to the single byte character set. It can be <b>2</b> for a double-byte character set, such as Chinese.</li> <li>After the <code>buildlg string .ext 1</code> command runs, copy the translated language files from translation directory (<code>C:\WFW</code>) into <code>C:\Autocall\TrueSite\TSW\Language</code> (Figure 9).</li> </ol>

Figure 6: Build options for .dat file

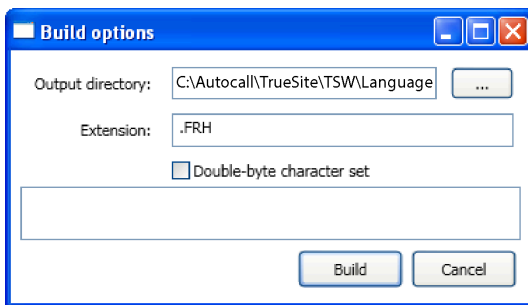


Figure 7: Building the language files

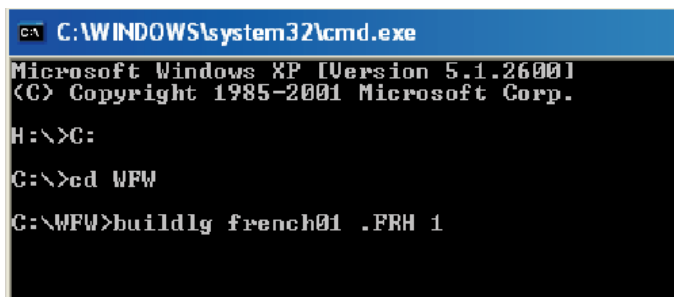


Figure 8: Successful completion of the Buildlg command

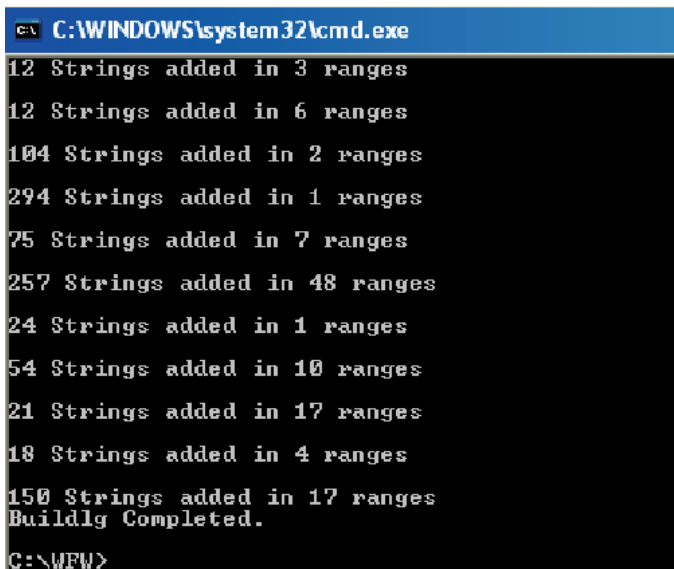
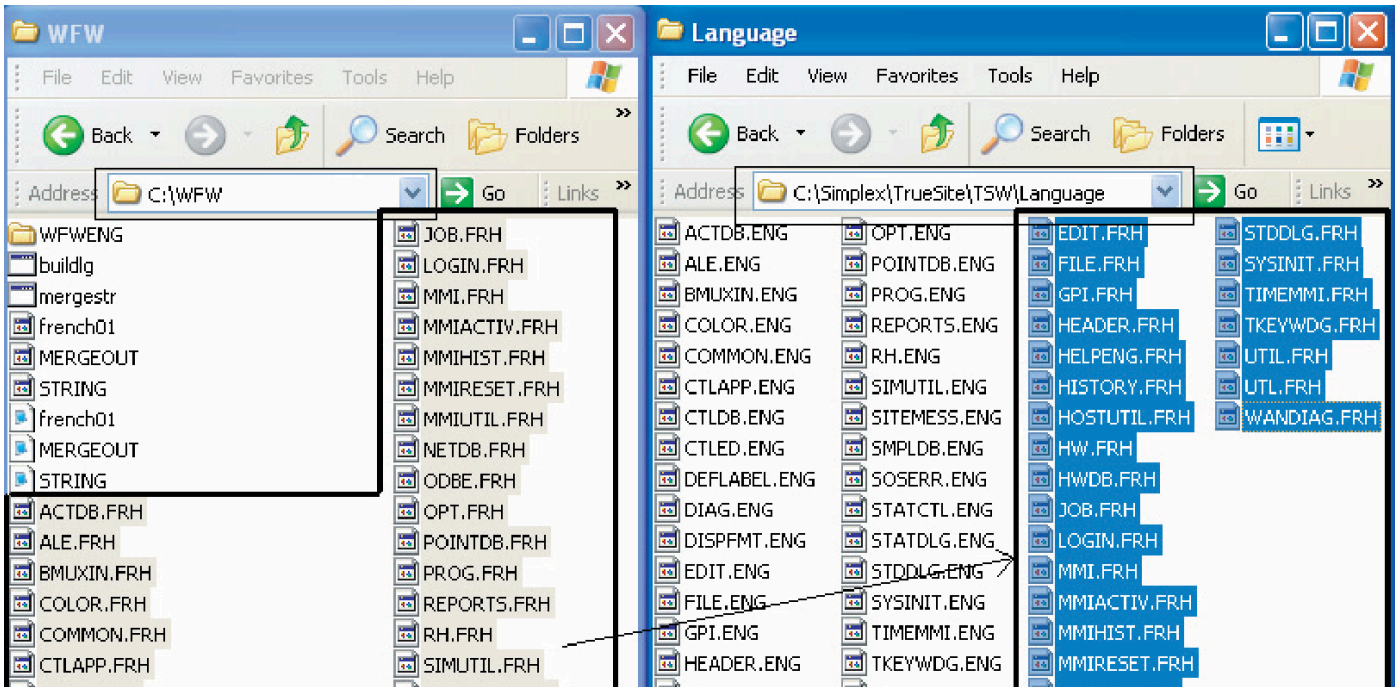




Figure 9: Copying the French language files into the TSW language directory





## Modifying the SOS.INI File

Update the Windows `sos.ini` file with the new language setting so TSW uses the translated language files. Follow the steps to update the `sos.ini` file:

1. Open the `sos.ini` file located inside the `C:\Windows` directory.
2. Search for the string `[Default32]` and locate the Language string underneath it. The Language string specifies the language library that TSW loads when starting up.

**Note:** If there is no Language line, English is selected by default. To enable a language change, modify the Language line (or added) in the following form (case must be preserved):

```
[Default32]
Language=.ENG
"ENG" specifies the 3-letter language extension. For French, the setting is:
[Default32]
Language=.FRH
```

3. Replace the `<LANGUAGE>` entry by the 3-letter short form of the language of choice in the following strings:

```
[Default32]
ALPHABET=<LANGUAGE>
```

For French, for example, the previous lines become:

```
[Default32]
ALPHABET=FRH
```

4. Search for the string `[Configurator32]` and set `MERGE` entry to `TRUE`. See the string below:

```
[Configurator32]
MERGE=TRUE
```

You can now run the TSW with the translated language files.

### Note:

1. The TSW Configurator is not translated.
2. The job database is translated when **MERGE=TRUE** is set and a job is built. Point labels are not translated until the job is rebuilt.
3. Set **MERGE=TRUE** to only translate system point labels. Set **MERGE=FALSE** for any other job modifications.
4. Character sets for languages like Chinese require changing the string `DoubleByteCharSet=FALSE` to **TRUE** in the **[Default32]** section of `sos.ini` file. This must match the last parameter from the `Buildg` command as mentioned in [Table 4](#).

## 3PII database translation procedure

This section describes the translation of the `.db` file included in the Localization Kit. This file is used for 3PII device strings. Follow the steps below to translate the file:

1. Open the **3PII** folder located inside the `WFW` folder within the Localization Kit.
2. Open the file `N3DB.LanguageCode.db` using the Localize application. This is the `.db` file associated with the latest version of TSW.
3. Click the **File** menu and click **Save As**. Save the file with a language code appropriate to the language. For example, for Canadian French, use the `fr-CA` language code and save the file as `N3DB.fr-CA.db`.
4. Click a row that you need to translate. Type into the Content column the translation of the text inside the Original column.
5. Repeat step 4 for every row that you need to translate. Click the **File** menu and then click **Save** to save the changes.
6. Copy the `.db` file, `N3DB.dbsec`, `N3DB.mhk`, and paste them into an appropriate folder. For example, paste `N3DB.fr-CA.db` into: `C:\Autocall\TrueSite\TSW\fr-CA`. If an appropriate folder does not exist you must create it.

**Note:** You must not rename `N3DB.dbsec` or `N3DB.mhk`.

7. Rename the `.db` file to `N3DB.db`.

The TSW can now use the translated `N3DB.db` file.

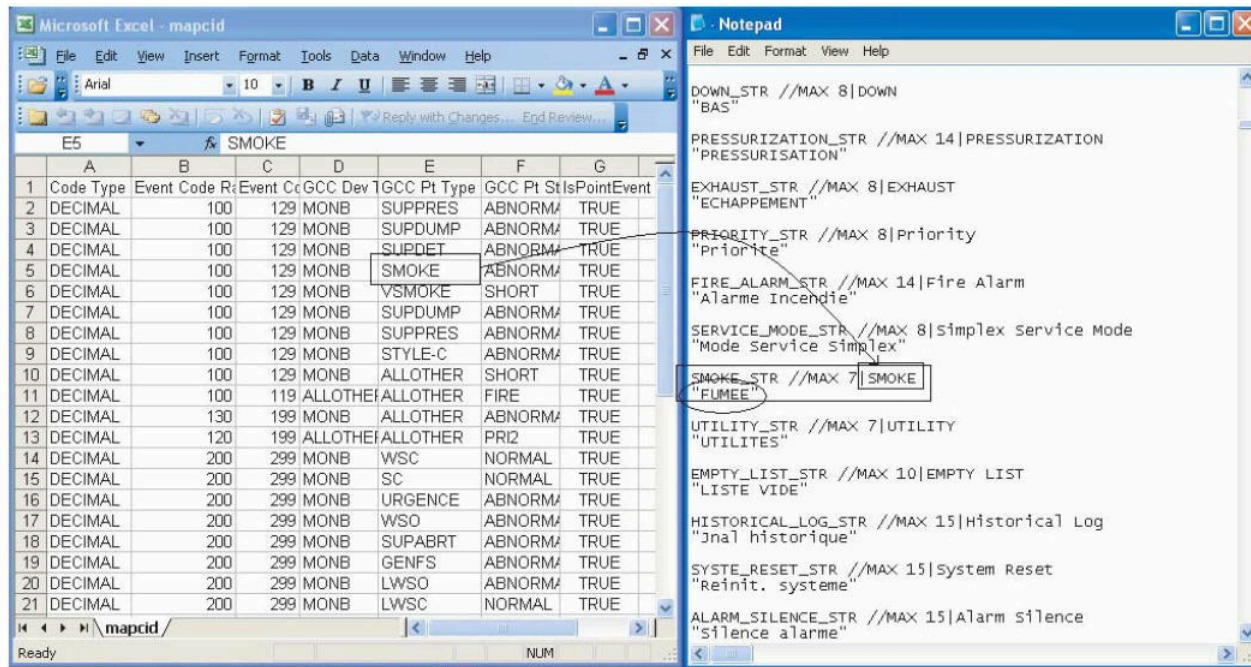
## DACR mapping files translation procedure

The Digital Alarm Communication Receiver (DACR) is a peripheral device used to communicate a panel's alarms to TSW through serial communication. To support communication with the DACR devices, TSW includes a particular set of files specific to DACR. Depending on the configuration, there may be seven or eight files, each one including information about various point types. Only the `mapcid.csv` file requires some translation. Follow the steps below:

1. Rename the `mapcid.csv` file located inside `C:\Autocall\TrueSite\TSW` to a name similar to `mapcid_english.csv`. This will be the original English language copy in case a rollback to English is necessary.
2. Open the `mapcid.csv` file located inside the `WFW\csv` of the Localization Kit, in a text editor such as Wordpad.
3. Translate the strings under the column **GCC Pt Type**.

**Note:** The original strings under the **GCC Pt Type** column must match the original English strings of the `string.dat` file. The translated strings under the **GCC Pt Type** column must match the strings of the translated `string.dat` file. For example, in [Figure 10](#) because the **SMOKE** string is translated as "FUMEE" inside the `.dat` file, the **SMOKE** string inside the `mapcid.csv` file must be changed to "FUMEE".

Figure 10: `DACR .csv` file translation



## ES Net Card digital pseudo points' labels translation procedure

For the digital points created for ES Net cards, the labels are stored in `DevicesAndPoints.xml` file and should be translated for each of the supported languages.

Please follow the steps below:

1. Rename `DevicesAndPoints.xml` file located in `C:\Autocall\TrueSite\TSW` to a name similar to `DevicesAndPoints_english.xml`. This will be the original English language copy in case of a rollback to English is necessary.
2. Open the `DevicesAndPoints.xml` file located inside `WF\WCSV` of the Localization Kit, in a text editor that supports your language character set.
3. Locate the first **NGNCardPoints** tag.
4. For each **NGNCardPoints** record, translate the **Description** and **Prefix** fields.
5. Copy the modified `DevicesAndPoints.xml` file to `C:\Autocall\TrueSite\TSW`.

## ES Net Diagnostics Tool translation procedure

This section describes translating the ES Net Diagnostics Service that is part of the TSW installation. TSW with ES Net Support provides a localized service with French and English languages. This means that when you switch TSW locale to the French language, the ES Net Diagnostics tool switches to the French language also.

**Note:** Local and remote TSW clients may have different localizations.

For ES Net Diagnostics support in other languages, complete the following steps:

1. Install the ES Net Diagnostics service.
2. Open the Services Control Panel by running the `services.msc` command, then stop the ES Net Diagnostics service.
3. Open `\Service\programs\web.browser\app\i18n` within the ES Net Diagnostics installation folder. The default path for the installation folder is `C:\Johnson\controls\ES Net Diagnostics`.
4. Open the `language.json` file using any text editor. The file contains key-value pairs of the strings that have to be translated into the required language.
5. Provide the required values' translations.
6. Set the language code. For example, `"_locale": "es"` for Spanish.
7. Save the translated file.
8. Restart ES Net Diagnostics in the Services Control Panel.
9. Restart TSW. The language automatically updates to the locale you installed.

## TSW online help translation procedure

The TSW software contains embedded help files that describe the functionality of the TSW. If you need to translate this TSW help content, complete the following steps:

1. Contact Technical Support to obtain the source files for the help content.
2. Purchase a license for the XML authoring tool OxygenXML Author, version 23.1 or higher.
3. Translate the source content, then use OxygenXML to transform this content to a .chm file. Refer to the OxygenXML documentation at the following web address: <https://www.oxygenxml.com/doc/versions/24.0/ug-editor/topics/chm-output-format.html>
4. Access the directory `Autocall\TrueSite\TSW\Client` and backup the previous version of `Truesite_Workstation_Help.chm` by moving the file to another directory, or renaming the file.
5. Add the .chm file that you generated from OxygenXML to the same location and name the file `Truesite_Workstation_Help.chm`.
6. Start the TSW, then select **Help Topics** from the **Help** menu, or press F1, to view the new help content.

## Testing the translation results

When the Localization process is complete, it is useful to run tests to ensure there are no issues with the translation. The following list summarizes the changes made to the TSW environment after completing the previous steps:

### Note:

1. For a client-only installation, only steps 1, 6, and 8 below are valid.
2. The string `<language code>` corresponds to the translated language code (for example, `fr-CA` for French).

1. New <code>TSWUI.Resources.dll</code> and <code>TSWResources.Resources.dll</code> files are created and moved inside <code>C:\Autocall\TrueSite\TSW\Client\&lt;language code&gt;</code> .
2. New <code>TSWResources.Resources.dll</code> is created and moved inside <code>C:\Autocall\TrueSite\TSW\&lt;language code&gt;</code> .
3. The new language files, such as with extension <code>.FRH</code> for French, are created and copied to the <code>C:\Autocall\TrueSite\TSW\Language</code> directory.
4. The <code>mapcid.csv</code> file is updated for the DACR jobs.
5. The file <code>sos.INI</code> , located inside <code>C:\windows</code> is modified to provide support for the new language.
6. The files <code>tswclient.exe.config</code> , <code>tswdesktop.exe.config</code> and <code>TswServerDesktop.exe.config</code> inside <code>C:\Autocall\TrueSite\TSW\Client</code> are modified to set the new application language.
7. The files <code>tswkrnl.exe.config</code> , <code>TSWConfigurator.exe.config</code> , <code>SSWriter.exe.config</code> , <code>gred.exe.config</code> inside <code>C:\Autocall\TrueSite\TSW</code> , are modified to set the new application language.
8. The new online help file is created and copied to <code>C:\Autocall\TrueSite\TSW\Client</code> , where the installation has been made to the default location.
9. The file <code>TSWBackupUtility.exe.config</code> inside <code>C:\Autocall\TrueSite\TSW</code> , is modified to set the new application language.
10. The file <code>TswFileTransfer.exe.config</code> inside <code>C:\Autocall\TrueSite\TSW</code> , is modified to set the new application language.

## Testing procedures

Both the TSW Configurator and the TSW Runtime can be validated for successful translation. Follow the steps below to validate:

**Note:** The TSW Configurator is part of a complete TSW installation only (Server and Client).

### Testing the TSW Configurator Translation

Two areas of the Configurator are translated: the Local Point labels and the **Utility Control** button labels. These can be verified as follows:

1. Start the TSW Configurator and log in.
2. Click **Configure** and then click **Point Config**.
3. Click **View**. Inside the **Owner Class** section, ensure only the **Local** checkbox is checked and click **OK**.

The resulting points list has translated custom labels.

### Utility Control button labels

1. Start the TSW Configurator and login.
2. Click **Configure** and then click **Utility Screens** and then **New**.
3. Enter a custom label and any number of rows.

The resulting view is a grid containing some rows. Each row can be configured to act in a particular way on particular points. For each of these, the corresponding Control and Button labels are translated. See the following steps for an example of what comes next:

1. Double-click anywhere in the **Mode** column.
2. Select **CD** (City Disconnect) from the drop-down list and click **OK**. The row is now marked as **CD** in the **Mode** column, with a translated label in the **Label** column.
3. Double-click any of the associated buttons to open a new window containing the button label at the bottom. This label is also translated.

### Testing the TSW Runtime Translation

In the case of TSW Runtime, translate the entire user interface. The following are sample TSW Runtime locations and functionalities that you can test to ensure the translation was successful:

---

**Note:** For best results, ensure the language for the TSW Client matches the language on the TSW Server machine.

1. Verify the login screen.
2. Open all menus and verify the text for each of them.
3. Open all tabs and verify that the labels and contents have been translated.
4. Invoke and verify pop-up dialogs, such as the **Point Status & Control** dialog box.
5. Try some other specific operations, such as generating and viewing a report from the **Reports** tab or exporting and viewing a historical log file.
6. Open the TSW online help and check various sections to confirm complete translation.

### Switching back to English

Refer to the section [Setting the language](#) for steps to switch TSW back to English. Apply the following settings:

1. Change the **Culture** back to en-US.
2. Ensure the **Double-byte** checkbox is unchecked.
3. The **Enable Merge** setting depends on whether the job has default point labels that were previously translated.
4. Change the extension back to `.ENG`.
5. To switch the `mapcid.csv` and the TSW online help files back to English, the original backed-up files must be restored.