String Extraction and Translation Toolkit for National Instruments LabVIEW® 2013

User Manual

Created By: Ryan Porter

Preface

I often have to translate my code from English to French before release and then sometimes back to English once the client has changed their mind. In LabVIEW this can be quite a laborious task because some of the text is on the front panel, some is on the block diagram, some is hidden away in property dialogs and subVIs. Although LabVIEW does provide a tool to export/import strings from a VI, I have found it to have too many limitations to be considered practical:

- Strings are exported to a pseudo-XML format which makes it difficult for a non-technical person to translate.
- String Import/Export is done on a per-VI basis. There is no tool to apply a language to an entire project.
- Too many strings are extracted. For example, I don't need to translate the descriptions for every control in the hidden error in/out clusters.
- If you modify the VI, you can no longer successfully import your translated strings.

For this reason, as well as taking the opportunity to learn LVOOP and the Actor Framework, I have developed the SET Toolkit for LabView. All development has been on my own time and I would like to share the result with the LAVA and NI community as a gesture of gratitude for all that I have learned from them over the years. If you have any questions, comments or would like to contribute to this project, contact Porter via PM on lavag.org.

About

The String Extraction and Translation Toolkit has been developed to provide edit-time project-level localization support for LabView 2013.

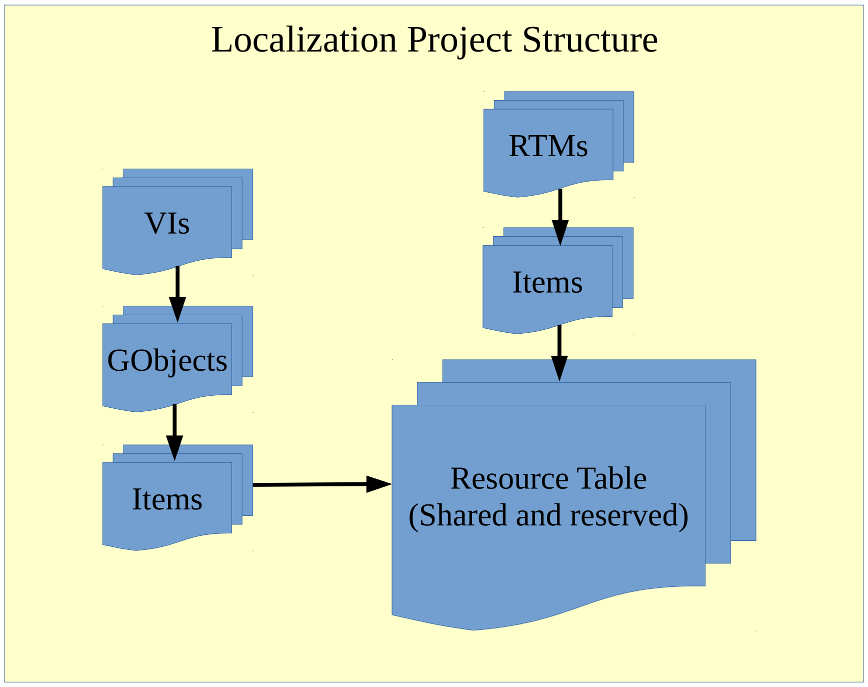
The key features currently included are:

- Wizard for exporting all UI text from a LabView project.
- Wizard for importing translated UI text to a LabView project.
- Export text to a CSV file format for translation by a third party.
- Store text in UTF-16LE to support most languages.
- Support for run-time menus.
- Support for multiple code pages (experimental).
- GUI for managing exported and translated UI text.
- Ability to define shared text that can be assigned to multiple UI elements.
- Ability to re-sync the localization data with the VI after the VI has been modified.

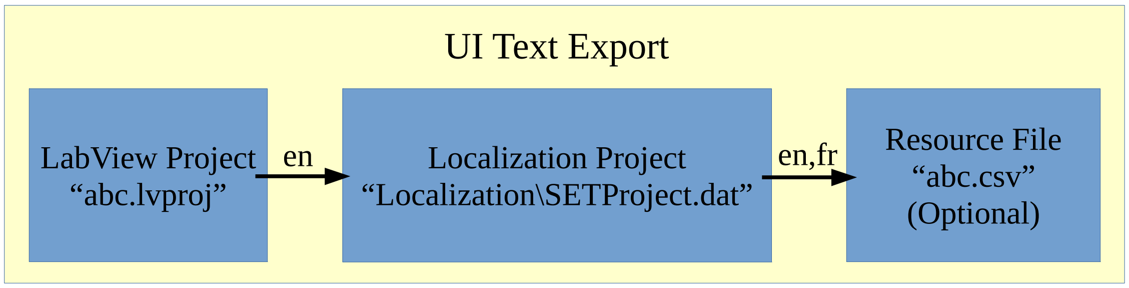
- Ability to preview translated UI and run-time menus.

How It Works

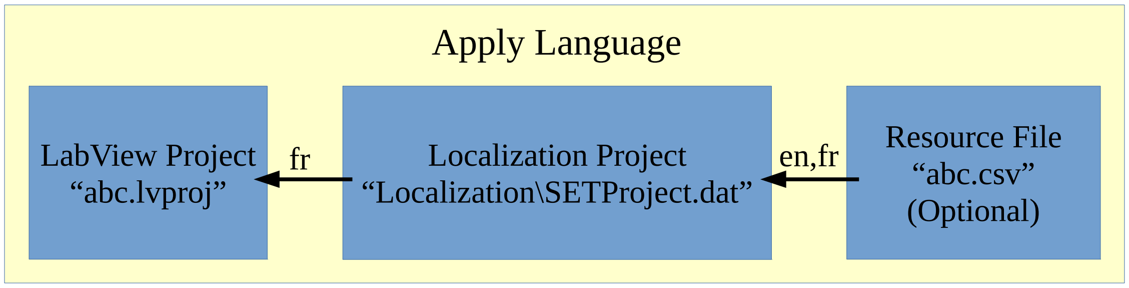
The SET Toolkit creates a Localization Project that is linked to the existing LabVIEW project that you want to localize. The Localization Project contains a global Resource Table (all UI text) as well as the meta-data required to locate the UI text in the LabVIEW project. See Appendix B for the file format details.



During creation, you are prompted to specify the source language. UI text that is extracted from the LabVIEW Project will be assumed to be in the source language. Once all UI text has been extracted from the selected VIs and RTMs of the LabVIEW project, target languages can be added.


UI Text Export
LabView Project "abc.lvproj"
en
Localization Project "Localization\SETProject.dat"
en,fr
Resource File "abc.csv" (Optional)

Optionally, the UI text can be exported to a Resource File for translation by a third-party. The Resource File is a CSV file with UTF-16LE character encoding. See Appendix A for the file format details. If the developer is performing the translation, they can use the SET Project Editor to translate UI text to the target languages.


Apply Language
LabView Project "abc.lvproj"
fr
Localization Project "Localization\SETProject.dat"
en,fr
Resource File "abc.csv" (Optional)

Applying a language to the LabVIEW project is also done through the SET Project Editor. Using the specified target language UI text and the meta-data from the Localization Project file, UI text is updated in the selected VIs and RTMs.

Getting Started

Requirements

- Microsoft Windows operating system
- LabView 2013 development environment

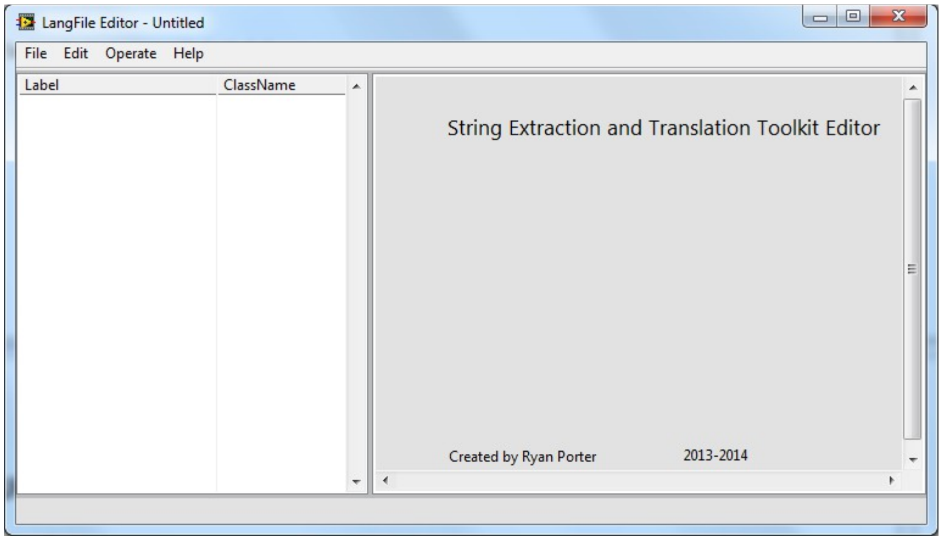
Installation

1. Un-zip the SET Toolkit to any folder.
2. Enable Unicode support by adding "UseUnicode=True" to the LabVIEW.ini file.

Using the SET Project Editor

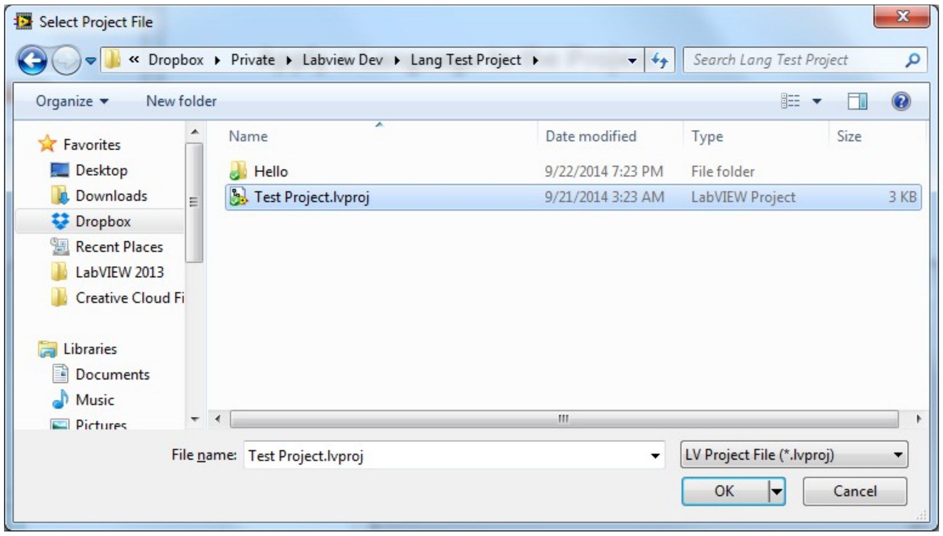
Start-up

1. Open “SET Project Editor.vi” located in the installation directory.
2. If it doesn't automatically run, press the run button.


LangFile Editor - Untitled
File Edit Operate Help
Label
ClassName
String Extraction and Translation Toolkit Editor
Created by Ryan Porter
2013-2014

Create a New Localization Project

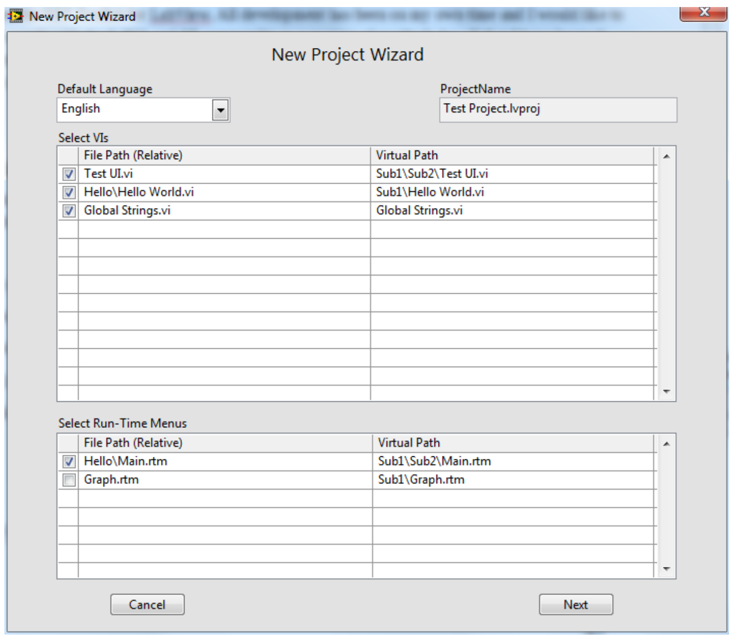
1. Select “File→New...” from the main menu.
2. Specify an existing LabView project by selecting its “.lvproj” file.



Name	Date modified	Type	Size
Hello	9/22/2014 7:23 PM	File folder	
Test Project.lvproj	9/21/2014 3:23 AM	LabVIEW Project	3 KB

File name: Test Project.lvproj
LV Project File (*.lvproj)

3. Select the language that the project is currently localized to from the “Default Language” list.
4. Select the VIs and RTM files that you would like to include in the localization project. Note that the paths displayed are relative to the directory of the “lvproj” file. The virtual path is the file's location as it would appear in the project explorer.



New Project Wizard

Default Language: English

ProjectName: Test Project.lvproj

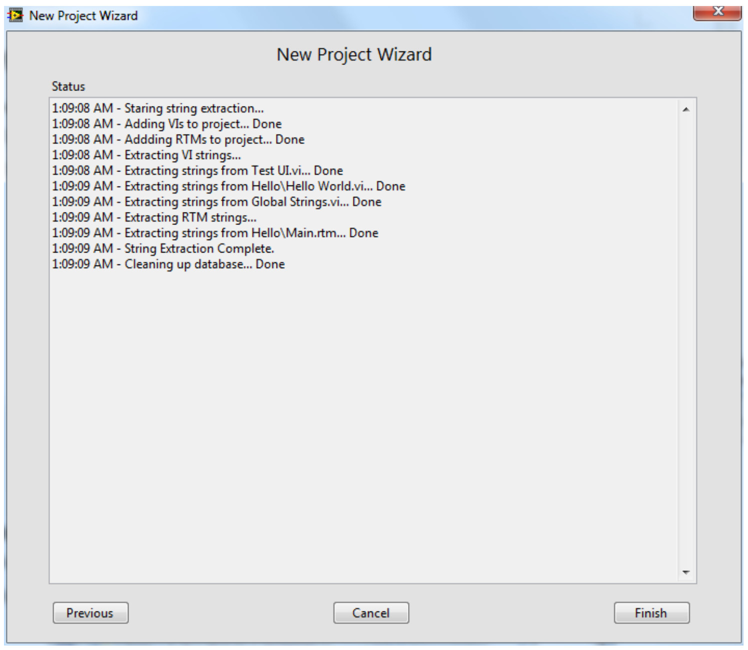
Select VIs

	File Path (Relative)	Virtual Path
✓	Test UI.vi	Sub1\Sub2\Test UI.vi
✓	Hello\Hello World.vi	Sub1\Hello World.vi
✓	Global Strings.vi	Global Strings.vi

Select Run-Time Menus

	File Path (Relative)	Virtual Path
✓	Hello\Main.rtm	Sub1\Sub2\Main.rtm
☐	Graph.rtm	Sub1\Graph.rtm

5. Press Next to extract the text from the selected items. Note that the wizard will extract UI text based on the rules outlined in Appendix C.



New Project Wizard

Status

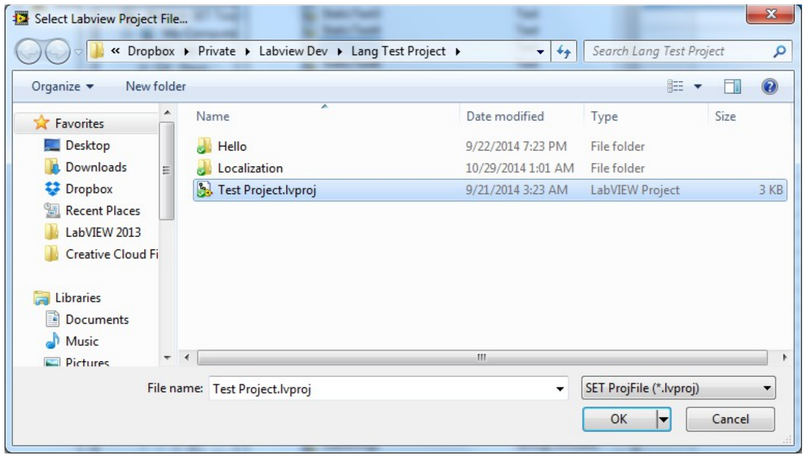
```
1:09:08 AM - Staring string extraction...
1:09:08 AM - Adding VIs to project... Done
1:09:08 AM - Addding RTMs to project... Done
1:09:08 AM - Extracting VI strings...
1:09:08 AM - Extracting strings from Test UI.vi... Done
1:09:09 AM - Extracting strings from Hello\Hello World.vi... Done
1:09:09 AM - Extracting strings from Global Strings.vi... Done
1:09:09 AM - Extracting RTM strings...
1:09:09 AM - Extracting strings from Hello\Main.rtm... Done
1:09:09 AM - String Extraction Complete.
1:09:09 AM - Cleaning up database... Done
```

6. Press Finish to load the new project to the editor.

7. Save the new project by selecting “File → Save” from the main menu or pressing “Ctrl+S”. Note that this automatically creates a sub-directory called “Localization” in the directory that contains the lvproj file. The localization project file is saved to this sub-directory under the name “SETProject.dat”.

Open a Localization Project

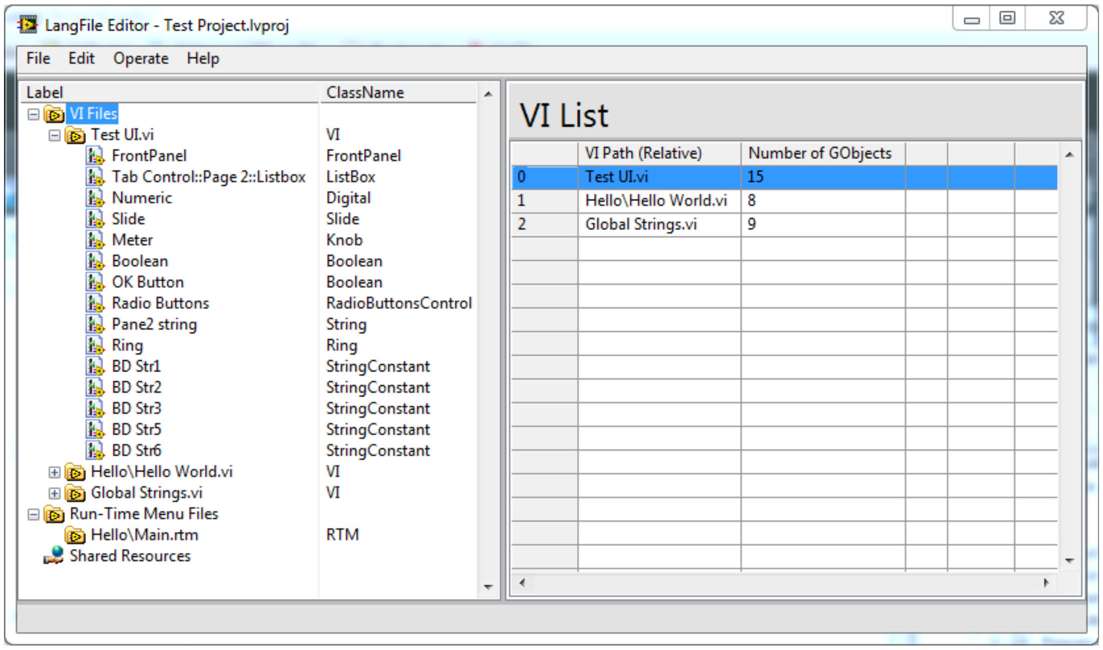
To open an existing localization project select “File → Open...” from the main menu or press “Ctrl+O”. Select the LabVIEW Project's “.lvproj” file. The project editor assumes that the localization project file is located within the Localization sub-directory.



Name	Date modified	Type	Size
Hello	9/22/2014 7:23 PM	File folder	
Localization	10/29/2014 1:01 AM	File folder	
Test Project.lvproj	9/21/2014 3:23 AM	LabVIEW Project	3 KB

File name: Test Project.lvproj
SET ProjFile (*.lvproj)

Project Editor UI



Label	ClassName
VI Files	
Test UI.vi	VI
FrontPanel	FrontPanel
Tab Control::Page 2::Listbox	ListBox
Numeric	Digital
Slide	Slide
Meter	Knob
Boolean	Boolean
OK Button	Boolean
Radio Buttons	RadioButtonsControl
Pane2 string	String
Ring	Ring
BD Str1	StringConstant
BD Str2	StringConstant
BD Str3	StringConstant
BD Str5	StringConstant
BD Str6	StringConstant
Hello\Hello World.vi	VI
Global Strings.vi	VI
Run-Time Menu Files	
Hello\Main.rtm	RTM
Shared Resources	

	VI Path (Relative)	Number of GObjects
0	Test UI.vi	15
1	Hello\Hello World.vi	8
2	Global Strings.vi	9

The SET Project Editor allows you to operate on as well as visualize the localization project. Navigate through the project items using the project explorer panel to the left. Each item, when selected, will display its properties in the view panel to the right.

On the project explorer panel, the localization project is broken into 3 categories; VI Files, RTM Files and Shared Resources.

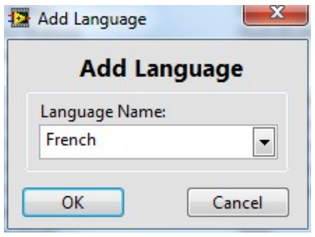
1. VI Files are identified by their relative path. Each VI File has a list of “GObjects”. These are UI elements that have been discovered. Each GObject has a list of items. These are parts of the

GObject that contain UI text. For example, a boolean control is considered to be a GObject and it has its caption, tip strip and boolean text as items.

2. RTM files are identified by their relative path. Each RTM file has a list of its menu entries. These correspond the user items of the RTM file.
3. Shared Resources is a list of resources from the resource table that have been marked as shared. That means they can be referenced by multiple RTM entries or GObject Items.

Add a Language

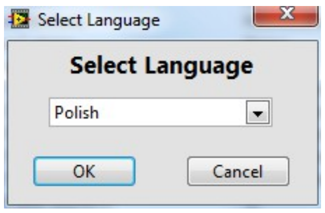
To add a target language, select "Edit → Add Language..." from the main menu then select the desired language from the list.



Language Name: French

Remove a Language

To remove a target language, select "Edit → Remove Language..." from the main menu then select the language from the list.

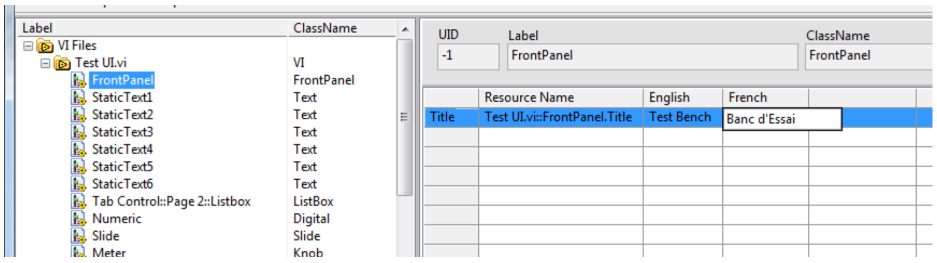


Polish

Note that there is no undo function for this action. If you save the project, you will loose all data associated with that language.

Edit Resource Text

UI text associated with any Gobject, RTM or Shared Resource can be edited directly within the view panel to the right of the project explorer. Select the Gobject, RTM or Shared Resources entry in the project explorer. The associated view is loaded to the view panel. Click on the text to start editing. Text input is in UTF16-LE character encoding.



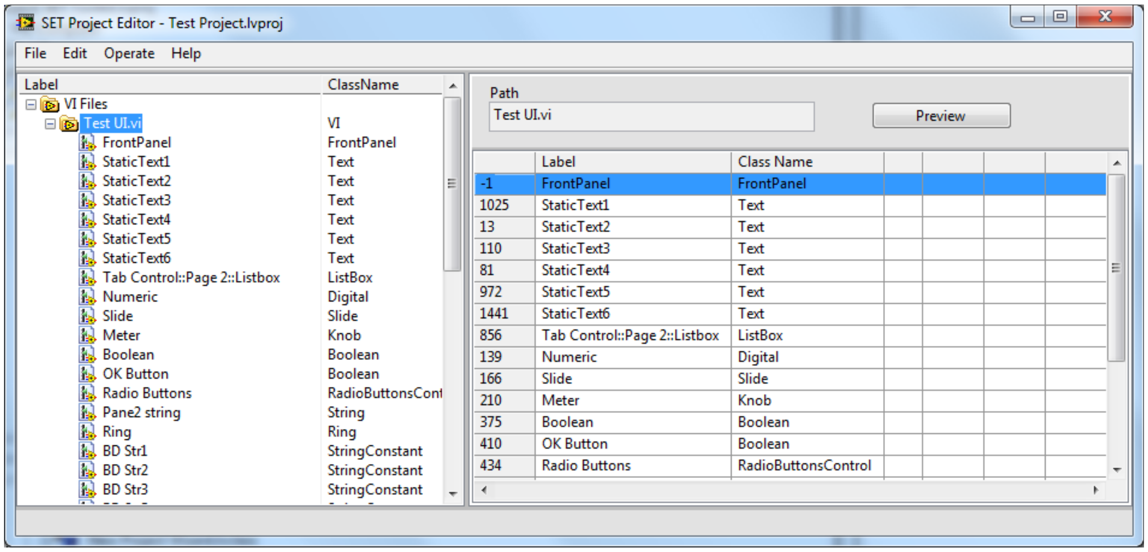
Label	ClassName
VI Files	
Test UI.vi	VI
FrontPanel	FrontPanel
StaticText1	Text
StaticText2	Text
StaticText3	Text
StaticText4	Text
StaticText5	Text
StaticText6	Text
Tab Control::Page 2::Listbox	ListBox
Numeric	Digital
Slide	Slide
Meter	Knob

UID	Label	ClassName
-1	FrontPanel	FrontPanel

	Resource Name	English	French
Title	Test UI.vi::FrontPanel.Title	Test Bench	Banc d'Essai

Preview VIs

To preview the front panel of a VI in a target language, select the VI in the project explorer then click the Preview button. Select the target language from the dialog box. The Selected VI will be opened with localized UI text.


SET Project Editor - Test Project.lvproj

File Edit Operate Help

Label	ClassName
VI Files	
Test UI.vi	VI
FrontPanel	FrontPanel
StaticText1	Text
StaticText2	Text
StaticText3	Text
StaticText4	Text
StaticText5	Text
StaticText6	Text
Tab Control::Page 2::Listbox	ListBox
Numeric	Digital
Slide	Slide
Meter	Knob
Boolean	Boolean
OK Button	Boolean
Radio Buttons	RadioButtonsCont
Pane2 string	String
Ring	Ring
BD Str1	StringConstant
BD Str2	StringConstant
BD Str3	StringConstant

Path

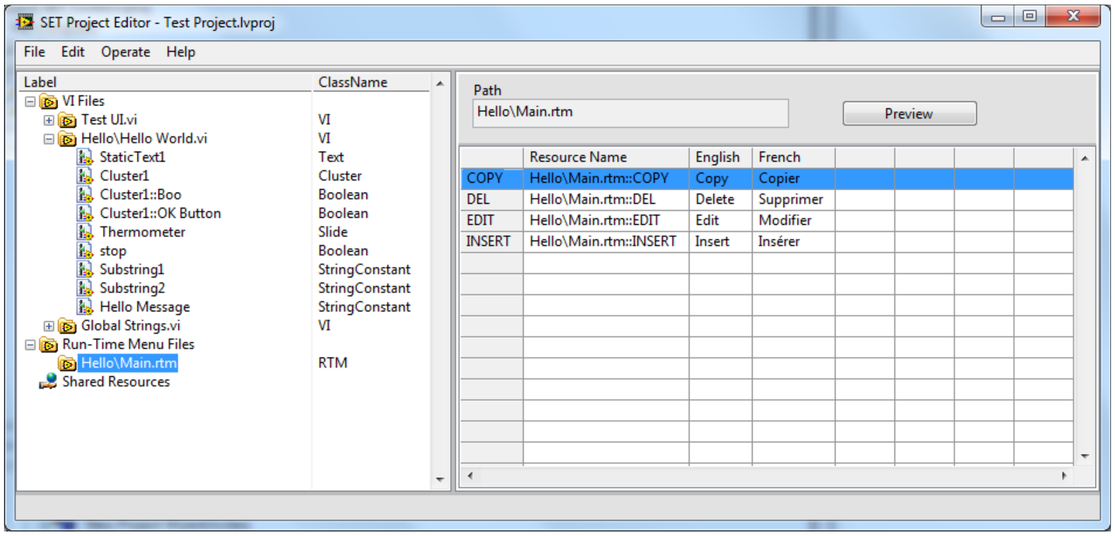
Test UI.vi

Preview

	Label	Class Name
-1	FrontPanel	FrontPanel
1025	StaticText1	Text
13	StaticText2	Text
110	StaticText3	Text
81	StaticText4	Text
972	StaticText5	Text
1441	StaticText6	Text
856	Tab Control::Page 2::Listbox	ListBox
139	Numeric	Digital
166	Slide	Slide
210	Meter	Knob
375	Boolean	Boolean
410	OK Button	Boolean
434	Radio Buttons	RadioButtonsControl

Preview RTMs

To preview a Run-Time Menu in a target language, select the RTM in the project explorer then click the Preview button. Select the target language from the dialog box. The Selected RTM will be localized and loaded to the RTM Preview dialog box's main menu.


SET Project Editor - Test Project.lvproj

File Edit Operate Help

Label	ClassName
VI Files	
Test UI.vi	VI
Hello\Hello World.vi	VI
StaticText1	Text
Cluster1	Cluster
Cluster1::Boo	Boolean
Cluster1::OK Button	Boolean
Thermometer	Slide
stop	Boolean
Substring1	StringConstant
Substring2	StringConstant
Hello Message	StringConstant
Global Strings.vi	VI
Run-Time Menu Files	
Hello\Main.rtm	RTM
Shared Resources	

Path

Hello\Main.rtm

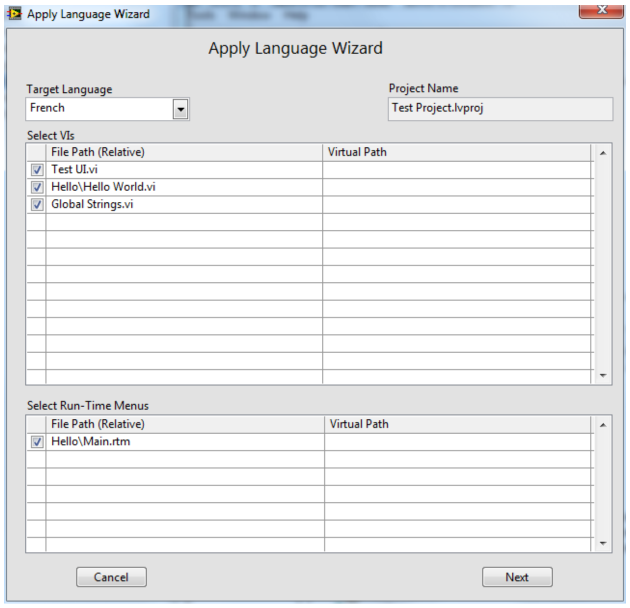
Preview

	Resource Name	English	French
COPY	Hello\Main.rtm::COPY	Copy	Copier
DEL	Hello\Main.rtm::DEL	Delete	Supprimer
EDIT	Hello\Main.rtm::EDIT	Edit	Modifier
INSERT	Hello\Main.rtm::INSERT	Insert	Insérer

Apply a Target Language to LabVIEW Project

Applying a language to the LabVIEW Project will set all UI text of selected project items to the target language.

1. Select "Operate → Apply Language to Project..." from the main menu.
2. Select the target language.
3. Select the VIs and RTMs to be localized. Then click Next...


Apply Language Wizard

Target Language: French

Project Name: Test Project.lvproj

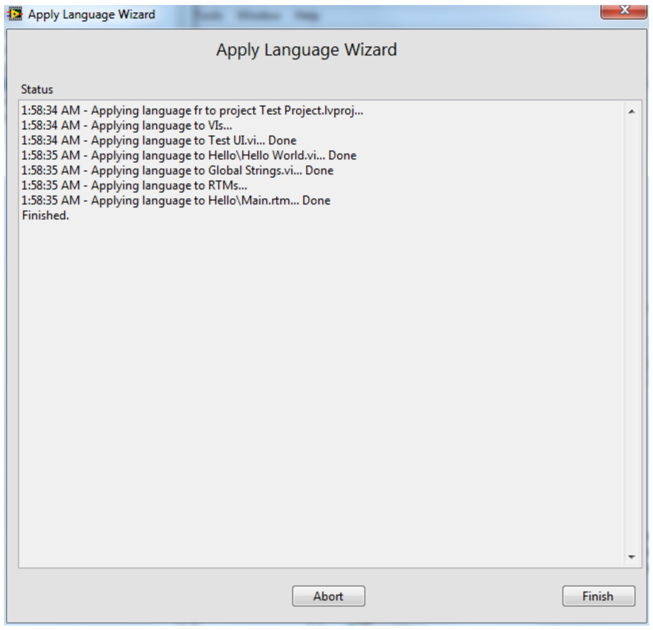
Select VIs

	File Path (Relative)	Virtual Path
✓	Test UI.vi	
✓	Hello\Hello World.vi	
✓	Global Strings.vi	

Select Run-Time Menus

	File Path (Relative)	Virtual Path
✓	Hello\Main.rtm	

4. Click Next to apply the language to selected items.


Apply Language Wizard

Status

```
1:58:34 AM - Applying language fr to project Test Project.lvproj...
1:58:34 AM - Applying language to VIs...
1:58:34 AM - Applying language to Test UI.vi... Done
1:58:35 AM - Applying language to Hello\Hello World.vi... Done
1:58:35 AM - Applying language to Global Strings.vi... Done
1:58:35 AM - Applying language to RTMs...
1:58:35 AM - Applying language to Hello\Main.rtm... Done
Finished.
```

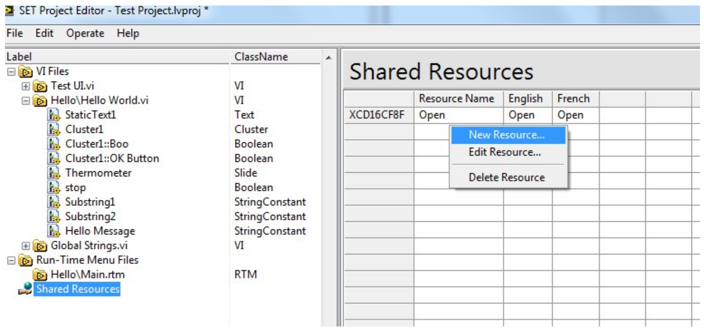
Shared Resources

To avoid translating the same text multiple times, you can define a shared resource. This is a resource that can be assigned to multiple UI text items.

To define a shared resource:

1. Select “Shared Resources” in the project explorer.

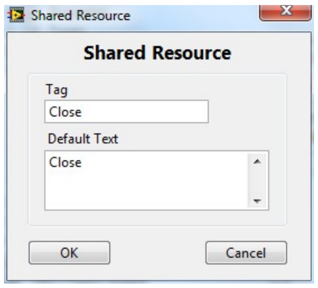
2. Right-click on the shared resource table to the right and select “New Resource...”.



Label	ClassName
VI Files	
Test UI.vi	VI
Hello\Hello World.vi	VI
StaticText1	Text
Cluster1	Cluster
Cluster1::Boo	Boolean
Cluster1::OK Button	Boolean
Thermometer	Slide
stop	Boolean
Substring1	StringConstant
Substring2	StringConstant
Hello Message	StringConstant
Global Strings.vi	VI
Run-Time Menu Files	
Hello\Main.rtm	RTM
Shared Resources	

	Resource Name	English	French
XCD16CF8F	Open	Open	Open

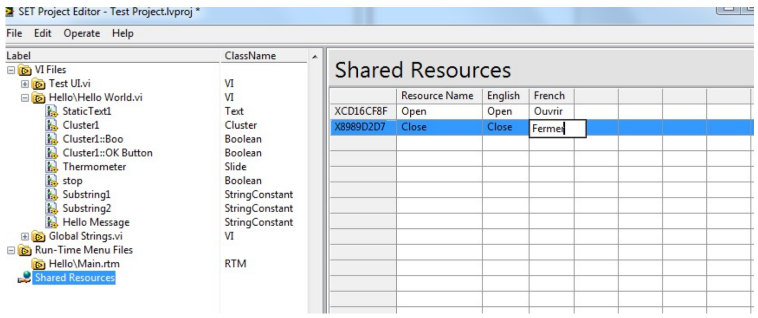
3. Enter a descriptive Tag and default UI text for the new shared resource.



Tag: Close

Default Text: Close

4. Translate the text for each target language (optional).



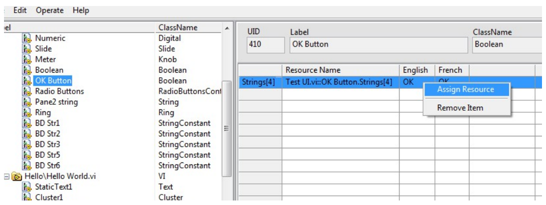
Label	ClassName
VI Files	
Test UI.vi	VI
Hello\Hello World.vi	VI
StaticText1	Text
Cluster1	Cluster
Cluster1::Boo	Boolean
Cluster1::OK Button	Boolean
Thermometer	Slide
stop	Boolean
Substring1	StringConstant
Substring2	StringConstant
Hello Message	StringConstant
Global Strings.vi	VI
Run-Time Menu Files	
Hello\Main.rtm	RTM
Shared Resources	

	Resource Name	English	French
XCD16CF8F	Open	Open	Ouvrir
X8989D2D7	Close	Close	Fermer

This shared resource can now be assigned to existing items:

1. Select the GObject or RTM that contains the item to assign the resource to.

2. Right-Click on the item and select “Assign Resource”.

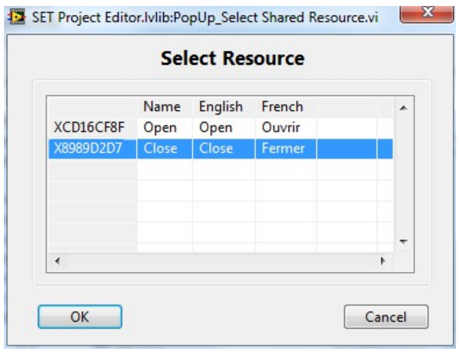


Label	ClassName
Numeric	Digital
Slide	Slide
Meter	Knob
Boolean	Boolean
OK Button	Boolean
Radio Buttons	RadioButtonsCont
Pane2 string	String
Ring	Ring
BD Str1	StringConstant
BD Str2	StringConstant
BD Str3	StringConstant
BD Str5	StringConstant
BD Str6	StringConstant
Hello\Hello World.vi	VI
StaticText1	Text
Cluster1	Cluster

UID	Label	ClassName
410	OK Button	Boolean

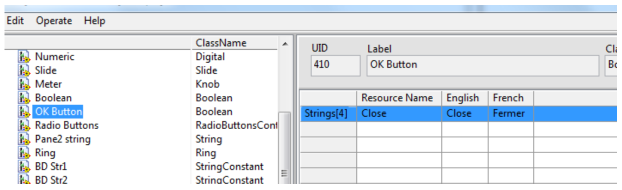
	Resource Name	English	French
Strings[4]	Test UI.vi::OK Button.Strings[4]	OK	OK

3. Select the Shared Resource to assign.



	Name	English	French
XCD16CF8F	Open	Open	Ouvrir
X8989D2D7	Close	Close	Fermer

4. Press OK.



	ClassName
Numeric	Digital
Slide	Slide
Meter	Knob
Boolean	Boolean
OK Button	Boolean
Radio Buttons	RadioButtonsCont
Pane2 string	String
Ring	Ring
BD Str1	StringConstant
BD Str2	StringConstant

UID: 410
Label: OK Button
Cl: Bo

	Resource Name	English	French
Strings[4]	Close	Close	Fermer

The item now points to the shared resource instead of its original reserved resource. The reserved resource has been deleted since no other items are referencing it.

Export resources to CSV

To export all resources to a CSV formatted text file:

1. Select “Edit → Export Resources to CSV...” from the main menu.
2. Specify the file path to export to.

The resulting file can be edited using most spreadsheet programs or text editors. For more information about the file format, refer to Appendix A.

Import Resources from CSV

Once the resources of a CSV file have been translated, they can be imported back to the Localization Project. To import:

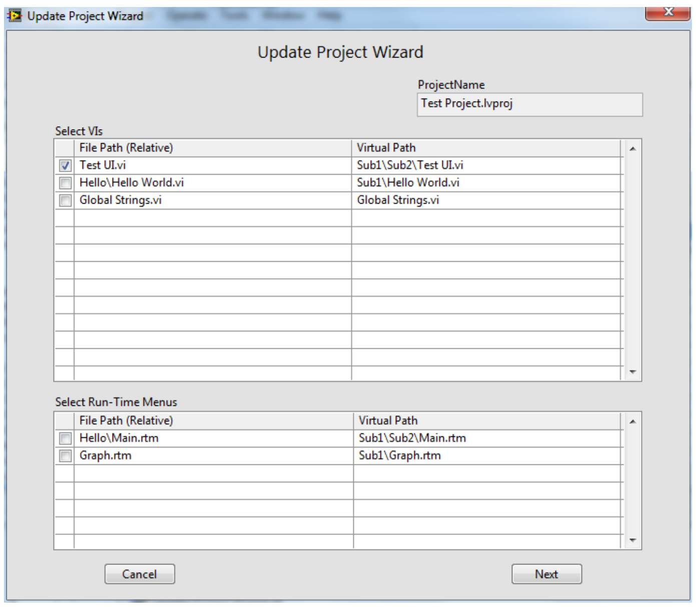
1. Select “Edit → Import Resource from CSV...” from the main menu.
2. Specify the CSV file path.

Note that any new resources in the CSV file will be ignored.

Update Localization Project Data

If parts of the LabVIEW Project are modified after the Localization Project has been created, they can be re-synced by running the Update Project Wizard:

1. Select “Operate → Update Data from Project...” from the main menu.



Update Project Wizard

ProjectName: Test Project.lvproj

Select VIs

	File Path (Relative)	Virtual Path
✓	Test UI.vi	Sub1\Sub2\Test UI.vi
☐	Hello\Hello World.vi	Sub1\Hello World.vi
☐	Global Strings.vi	Global Strings.vi

Select Run-Time Menus

	File Path (Relative)	Virtual Path
☐	Hello\Main.rtm	Sub1\Sub2\Main.rtm
☐	Graph.rtm	Sub1\Graph.rtm

2. Select the VIs and RTMs that need to be re-synced or added.

3. Click Next to start the UI text extraction.

Appendix A - Resource File Format

File Type: CSV file as per RFC-4180 (see http://tools.ietf.org/html/rfc4180)

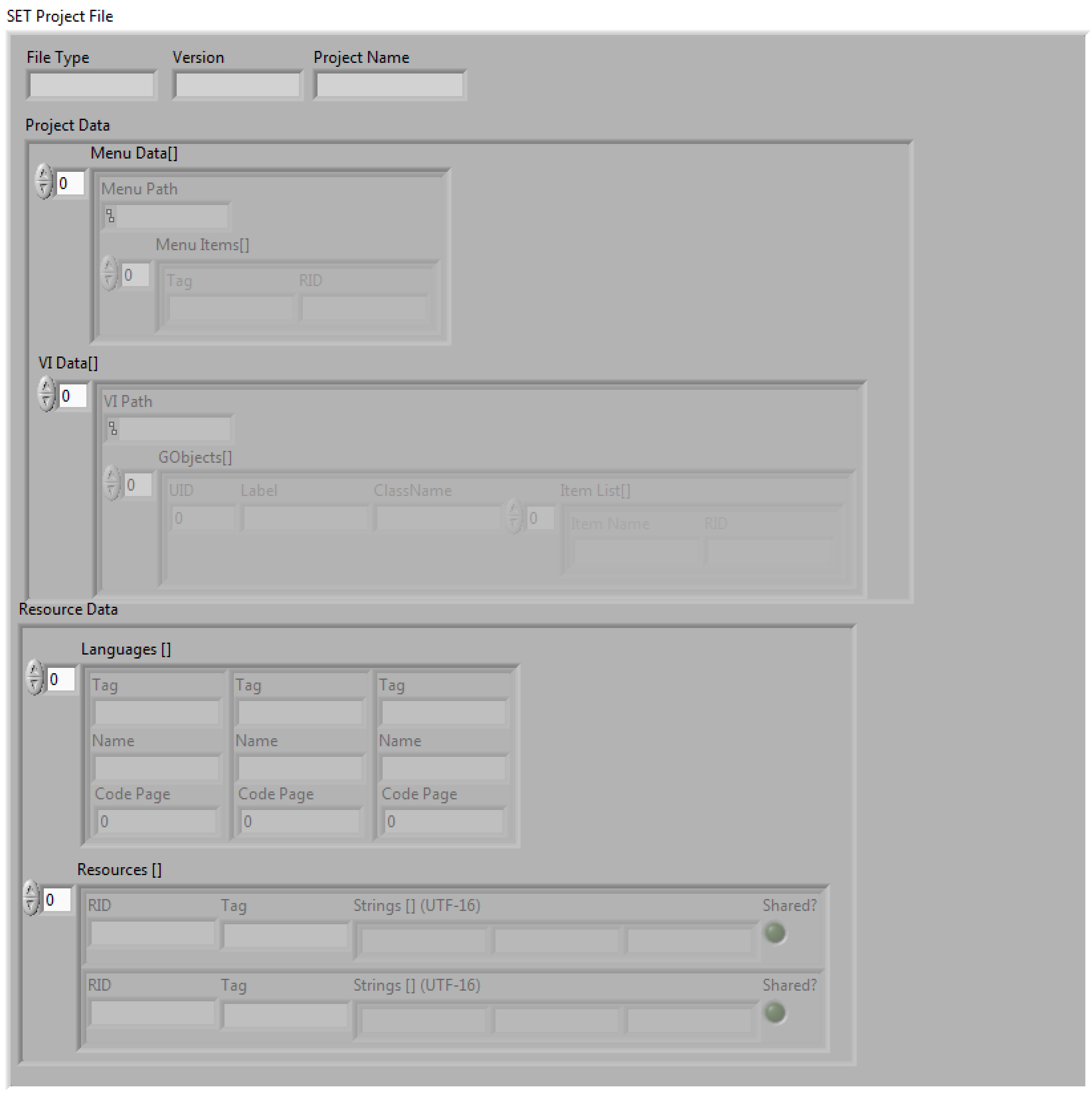
Character Encoding: UTF-16LE, First character of file is the BOM (0xFFFE)

File Header (first line): RID,Tag,<Lang1>,<Lang2>...
<Lang> = Language name in English with ISO 639-1, 2-letter language code.

Details:

- Cells are separated using a comma.
- Multi-line cells are allowed.
- Commas within cells are allowed.

Appendix B - Localization Project File Format



Binary file with the following sections:

1. File Type: String, (Must equal “SETPROJFILE” to be considered valid)
2. Version: String, (Currently at “1.0.0”)
3. Project Name: LabView poject file name (including the “.lvproj” extension)
4. Project Data: Cluster containing Menu Data[] and VI Data[]
5. Resource Data: Cluster containing Language list and Resource table

Menu Data Type Information:

1. Menu Path: Path, RTM's path relative to the LV Project File. This is considered to be the unique identifier for the RTM.
2. Menu Items[]: Array of clusters containing 2 fields – “Item Name” and “RID”.
   - Item Name: String, menu user item name.
   - RID: String, Resource identifier. Refers to an entry in the resource table.

VI Data Type Information:

1. VI Path: Path, VI's Path relative to the LV Project File. This is considered to be the unique identifier for the VI.
2. GObjects[]: Array of clusters of type GObject Data

GObject Data Type Information:

1. UID: I32, Unique identifier of the GObject within the VI.
2. Label: String, GObject's label.
3. ClassName: String, GObject's class name as reported by “ClassName” property node.
4. Item List[]: Array of clusters containing 2 fields – “Item Name” and “RID”.
   - Item Name: String, name of the part of the GObject.
   - RID: String, Resource identifier. Refers to an entry in the resource table.

Note: GObject refers to controls, indicators, block diagram strings, front panel static text etc...

Languages Type Information:

1. Tag: String, Language Code (ISO 639-1, 2-character).
2. Name: String, Language name (in English).
3. Code Page: U16, Windows (ANSI) Code Page Identifier.

Resources Type Information:

1. RID: String, Unique Resource Identifier.
2. Tag: String, Resource label.
3. Strings[] (UTF-16): Array of strings, Resource's UI text for each language of the Language Table. Stored with UTF-16LE character encoding.
4. Shared?: Boolean, Flag to identify resource as a shared resource.

Appendix C - UI Text Extraction Rules

If an item does not meet the extraction condition, it will be skipped. If all items of a GObject are skipped, the Gobject will be skipped.

GObject	ClassName	Items	Condition
Front Panel	FrontPanel	Title	Must be custom title
BD String Constant	StringConstant	Text	Label visible & not within a structure
BD Cluster Constant	ClusterConstant	StringConstant(s)	Label visible & not within a structure
BD Array Constant	ArrayConstant	Data	Label visible & Not within a structure
FP Decoration Text	Text	Text	None
FP Control*	Control	Caption TipStrip	Caption visible or Tip Strip not empty
->FP Boolean Control	Boolean	Caption TipStrip Strings[4]	Caption visible or Tip Strip not empty or Boolean text visible
->FP Ring Control	Ring	Caption TipStrip Strings[]	Caption visible or Tip Strip not empty or Items list not empty
->FP String Control	String	Caption TipStrip DefVal	Caption visible or Tip Strip not empty or Default Value not empty
->FP Tab Control	TabControl	Caption TipStrip TabCaptions[]	Caption visible or Tip Strip not empty or Tabs visible
->FP Graph/Chart**	GraphChart	Caption TipStrip XScales[] YScales[]	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label
-->FP Waveform Chart	WaveformChart	Caption TipStrip XScales[] YScales[] Plots[]	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label or At least one Plot Label
-->FP Intensity Chart	IntensityChart	Caption TipStrip XScales[] YScales[] ZScale	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label or ZScale Label not empty

-->FP Intensity Graph	IntensityGraph	Caption TipStrip XScales[] YScales[] ZScale Cursors[]	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label or ZScale Label not empty or At least one Cursor Name
-->FP Mixed Signal Graph	MixedSignalGraph	Caption TipStrip XScales[] YScales[]	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label
–>FP Waveform Graph	WaveformGraph	Caption TipStrip XScales[] YScales[] Plots[] Cursors[]	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label or At least one Plot Label or At least one Cursor Name
–>FP Digital Graph	DigitalGraph	Caption TipStrip XScales[] YScales[] Plots[] Cursors[]	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label or At least one Plot Label or At least one Cursor Name
-->FP XY Graph	XYGraph	Caption TipStrip XScales[] YScales[] Plots[] Cursors[]	Caption visible or Tip Strip not empty or At least one XScale Label or At least one YScale Label or At least one Plot Label or At least one Cursor Name
Runtime Menu	RTM	Menu Item(s)	Item Tag does not start with APP

*FP Control items are inherited by all child classes of front panel controls.

**FP Graph/Chart items are inherited by all child classes of GraphChart.

Useful References

1. RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files, http://http://tools.ietf.org/html/rfc4180
2. Localize LabVIEW Applications in Multiple Languages - National Instruments, http://http://www.ni.com/newsletter/50099/en/
3. Localization Configuration Editor Reference Example - National Instruments, http://www.ni.com/example/31257/en/
4. Passa Mak – LAVA, http://lavag.org/files/file/98-passa-mak/
5. Community: A List of Tips and Tools for using Unicode in LabVIEW, https://decibel.ni.com/content/docs/DOC-10153
6. List of ISO 639-1 codes - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes
7. Israel Science and Technology Homepage - Windows Locale Codes - Sortable list, http://www.science.co.il/Language/Locale-codes.asp
8. Code Page Identifiers (Windows), http://msdn.microsoft.com/en-us/library/dd317756
9. Wikipedia: Code Page, http://en.wikipedia.org/wiki/Code_page
10. MSDN: MultiByteToWideChar function: http://msdn.microsoft.com/en-us/library/windows/desktop/dd319072(v=vs.85).aspx
11. MSDN: WideCharToMultiByte function, http://msdn.microsoft.com/en-us/library/windows/desktop/dd374130(v=vs.85).aspx
12. A Few of the Gotchas of MultiByteToWideChar, http://www.siao2.com/2005/04/19/409566.aspx
13.