ANV Language Support Toolkit

by ANV s.r.o.

User Manual

For more of our products, see: https://www.anv-tech.com/en/software-development/

To contact our online support, please send an email to: support@anv-tech.com

Table of Contents

Change History

Date	Modified by	Changes
19.11.04	E. Tisovský	Initial document creation
19.11.11	E. Tisovský	Updated document
19.11.13	E. Tisovský	Updated document
19.11.18	E. Tisovský	Updated document
20.01.07	E. Tisovský	Added licensing info

About This Manual

This manual serves as thorough introduction for usage of ANV Language Support Toolkit. Users of the toolkit are recommended to carefully inspect this manual before using the toolkit.

Information in this manual is accurate for the current version of toolkit to the best knowledge of the authors. We reserve rights for any errors which might appear in this manual.

Abbreviations

API – Application programming interface

JSON – JavaScript Object Notation

LST – Language Support Toolkit

VIPM – VI Package Manager

Requirements

These are the minimum software requirements for proper functionality of toolkit:

Windows 10 32 – bit.

LabVIEW 2015 SP 1 or higher.

Following 3rd party VI packages available through JKI VI package manager are used by the toolkit:

1. MGI Application Control >= Version 1.1.1.10
2. OpenG Application Control Library >= Version 4.1.0.7
3. OpenG Array Library >= Version 4.1.1.14
4. OpenG Error Library >= Version 4.2.0.23
5. OpenG File Library >= Version 4.0.1.22
6. OpenG LabVIEW Data Library >= Version 4.2.0.21
7. OpenG String Library >= Version 4.1.0.12
8. Third Party Licensing & Activation Toolkit >= Version 15.0.3.43

All packages mentioned above are included in toolkit installer by default.

Licensing and activation

IMPORTANT: During activation of the toolkit, it is necessary that LabVIEW is run with administrator rights. Otherwise, activation process uses your license without activating the toolkit.

ANV LST is available with two licensing options – development and deployment licensing. Development license allows users to use all options, which toolkit offers. After the toolkit is installed through VIPM, it comes with 30-day evaluation period. Note that with evaluation version of LST, user cannot build LabVIEW executables – this requires activation of the toolkit. You will be prompted about license status when you launch LabVIEW, unless your license is already activated.

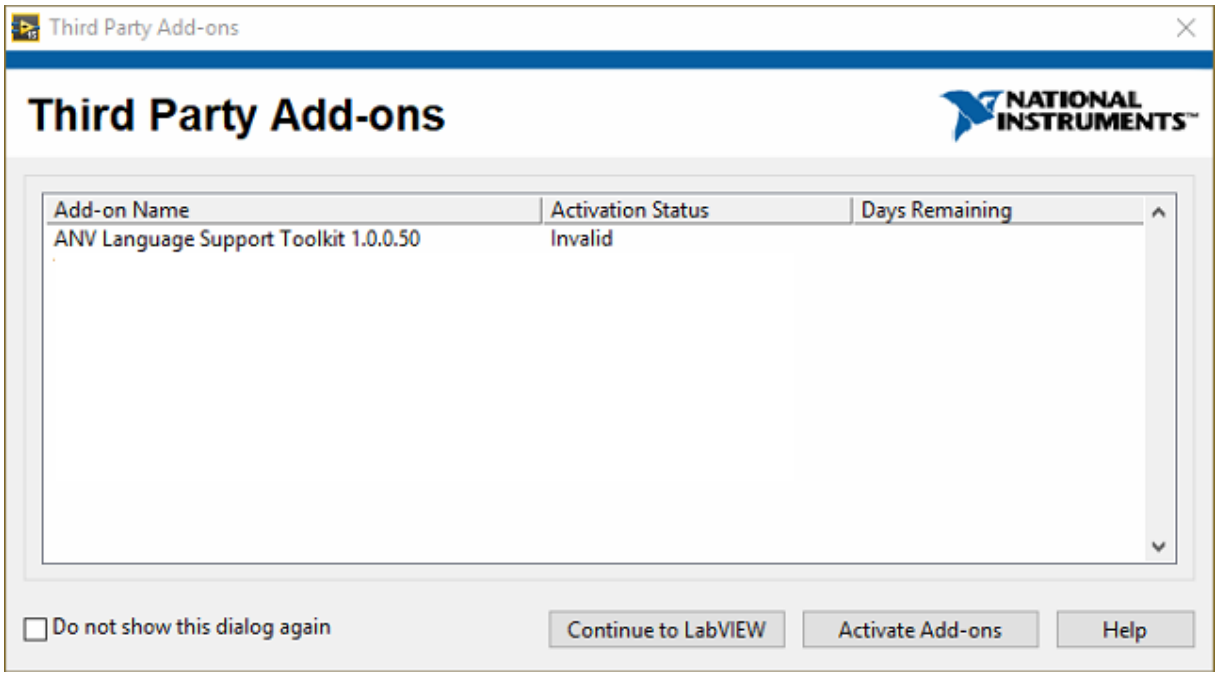


Fig. 1: LabVIEW third party Add-on window

To activate development license, either click Activate Add-ons button, or use LabVIEW menu. Select Help =>Activate Add-ons from LabVIEW Toolbar. Dialog box with the list of available toolkits appears. Select ANV Language Support Toolkit from the list and follow the activation wizard in order to activate toolkit.

You have two options to activate your license – either automatically or manually. Automatic activation is done online. You enter License ID and Password you received when you purchased license into appropriate cells and click activate. If the data you entered are

Manual activation is used, when your machine cannot connect to the internet. When you choose manual activation, user codes appear in dialog. On another machine, open browser and access https://secure.softwarekey.com/solo/unlock. Enter License ID and password that you received when you purchased your license and submit them. Then, enter user codes from the dialog of the machine, where you want to activate the license, into appropriate columns. You will then receive activation code, which you may enter below user code fields inside "Activation code 1" cell. Click next. Your license should be activated now.

IMPORTANT: Make sure that user codes you entered online are correct and do not cancel activation dialog until you enter activation code. Your license is used when you enter it online, so if you entered incorrect user codes online, or you cancelled the dialog, you will not be able to activate workstation with the license.

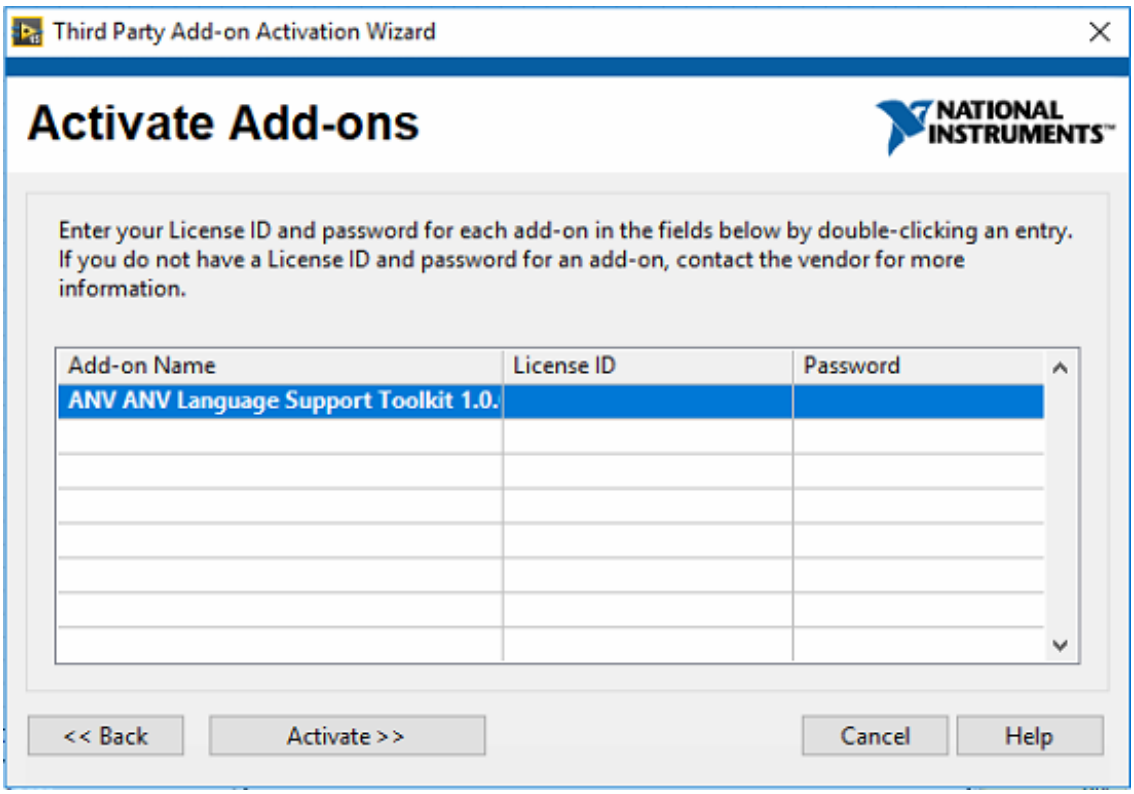


Fig. 2: Automatic license activation dialog

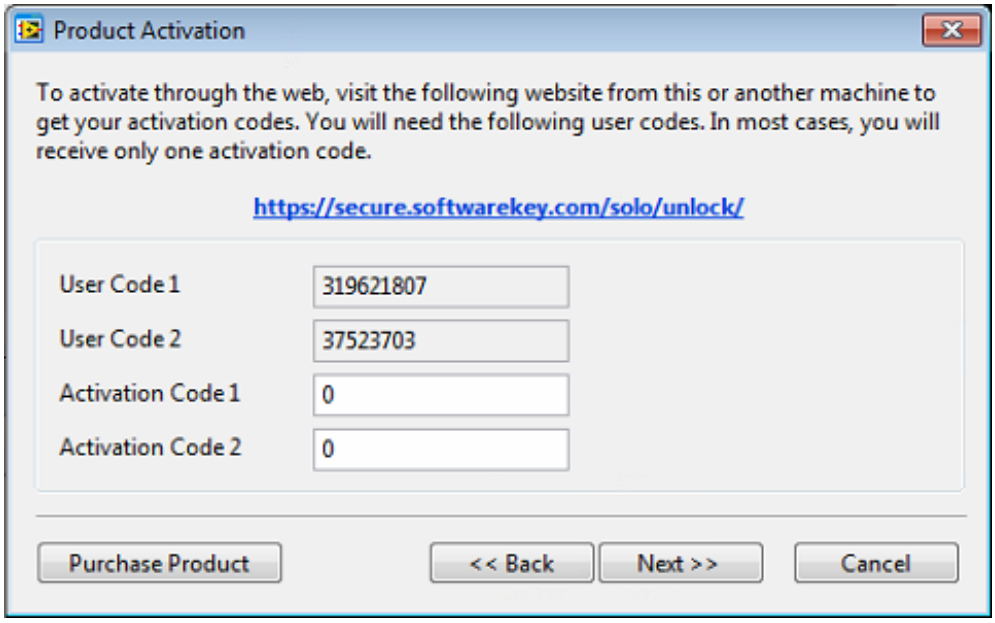


Fig. 3: Manual license activation dialog

Deployment license is used by applications, which use toolkit's API. This license comes with 7-day evaluation period. Until you activate toolkit, you will be prompted to do so each time you initialize toolkit API. For activation of deployment license, select Tools=>ANV Language Support toolkit =>Activate Runtime license and follow activation wizard in order to activate license. Alternatively, you will be prompted to activate license when you run API which is not activated yet. Activation dialog is similar to activating development license, with both manual and automatic activation available to user. Before activation of deployment license, make sure you have run executable installer (See Toolkit API section, creating executable with API).

Installation

Toolkit is installed as package via JKI VI package manager. You can download it either through VIPM or from our website. After installation, 30-day evaluation period begins. Please see our website for download links.

During the installation/un-installation process, both LabVIEW and VIPM should be launched with administrator rights, otherwise installation may not work.

Toolkit Setup

LabVIEW has limited support for Unicode strings in the front panel controls and indicators on Windows operating system. Instead, it uses Multibyte Character Strings (MBCS). More information about this can be found here: https://forums.ni.com/t5/Reference-Design-Content/LabVIEW-Unicode-Programming-Tools/ta-p/3493021.

For proper functionality of the toolkit, language for non-Unicode programs of PC, on which Language Support Toolkit is used needs to be configured. Following guide describes, how to achieve this on PC with Windows 10 operating system.

1. Go to control panel and select "Clock and Region"

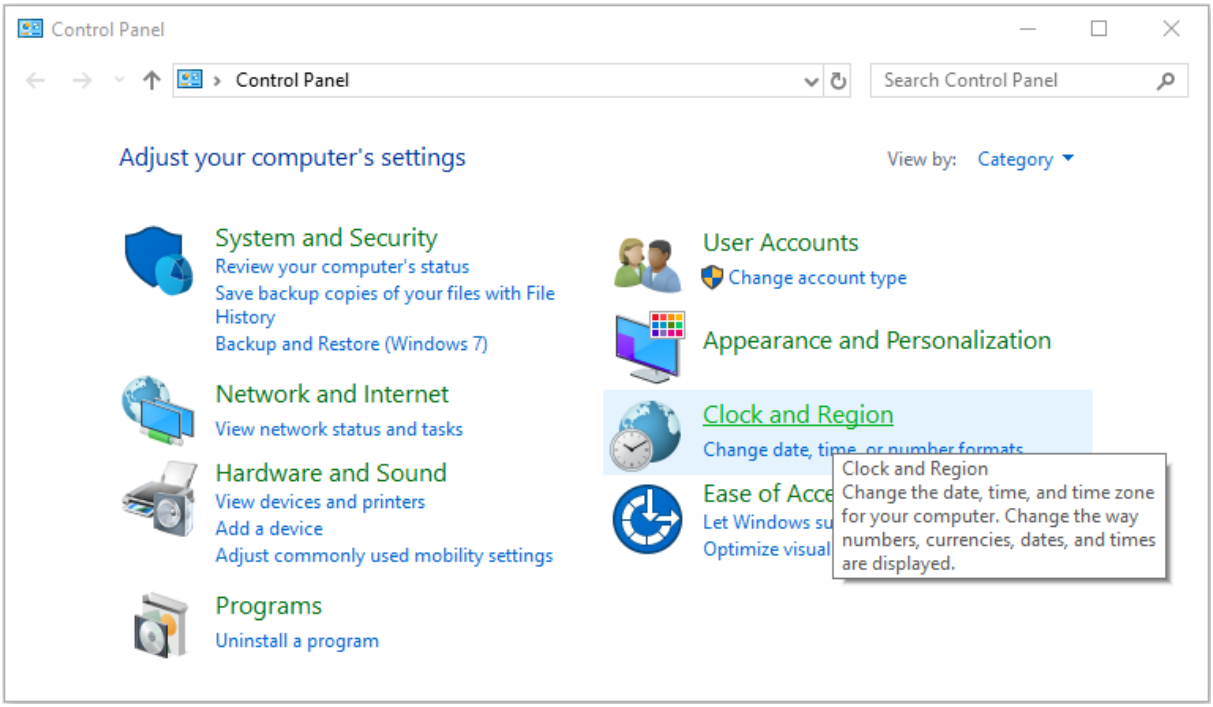


Fig. 4: Windows Control Panel

2. Select "Region". In the popup window which appears, select "Administrative" tab. Here, click on "Change system locale".

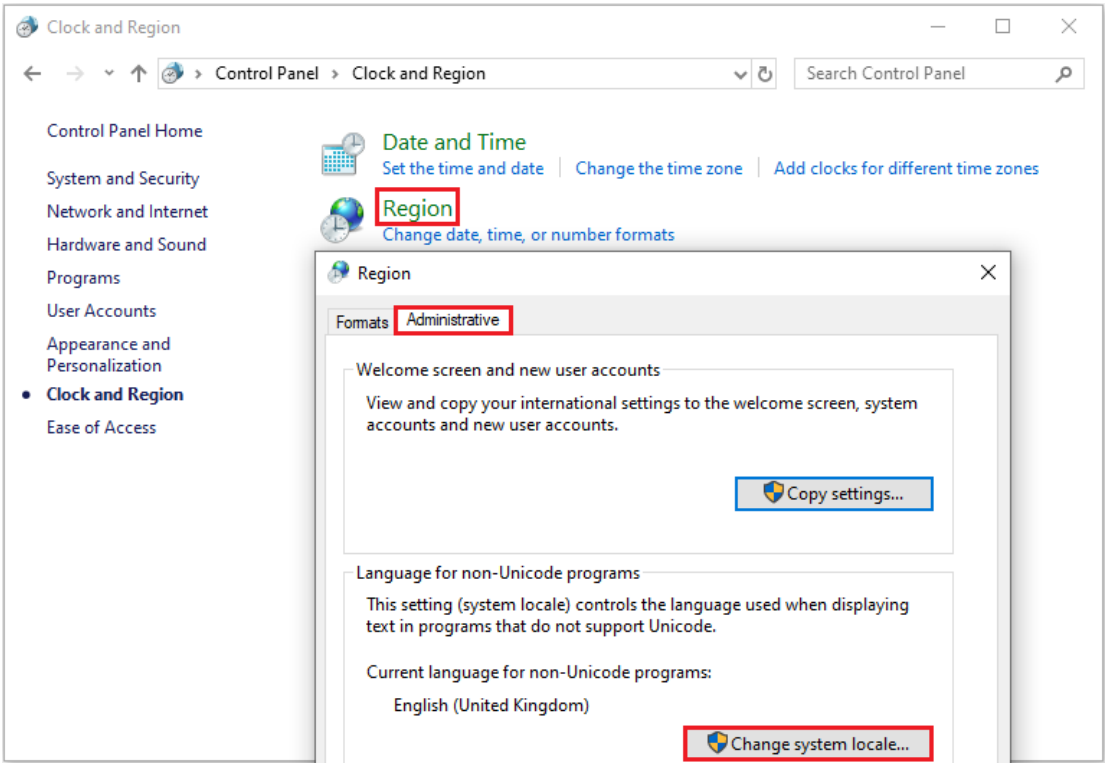


Fig. 5: Control Panel Clock and Region settings

3. Another popup window appears. Here, select the language, which you would like to use for translation. Also, mark checkbox at the bottom of popup window.

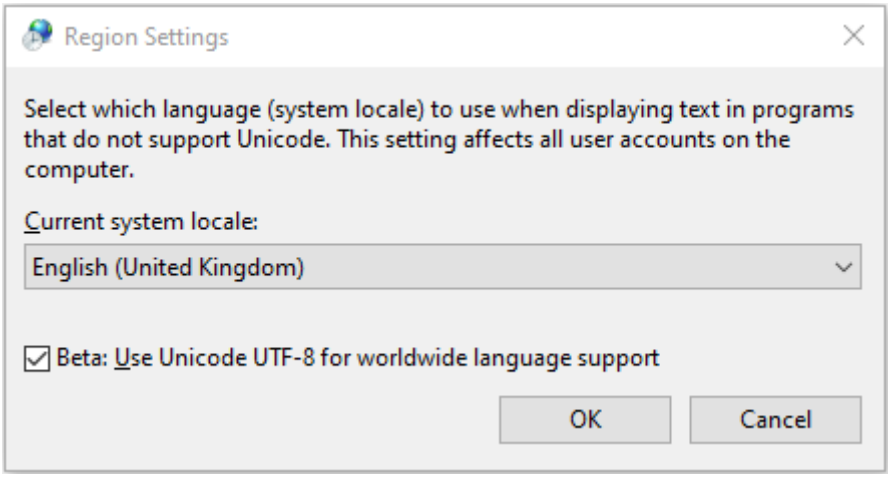


Fig. 6: Changing system settings for non-unicode programs

4. Restart your computer to apply changes

Toolkit Description

Introduction

ANV Language Support Toolkit (ANV LST) is toolkit for localizing LabVIEW applications by translating user interface text to different languages. Toolkit is used as LabVIEW add-on, and consists of two main parts: configuration part (accessible from LabVIEW Tools menu) and API part (used for integration into source code). Additionally, sample project is provided as both example and project template, to demonstrate toolkit API to user. Also, this documentation can be accessed through Help => ANV Language Support Toolkit… => Manual.

Configuration

Toolkit has the following menu options:

1) Choose LST folder
2) Languages Settings
3) Create Dictionary for VI
4) Translate this VI
5) Validation
6) Manual
7) Activate Runtime License

These options are available from Tools menu of projects or a VIs.

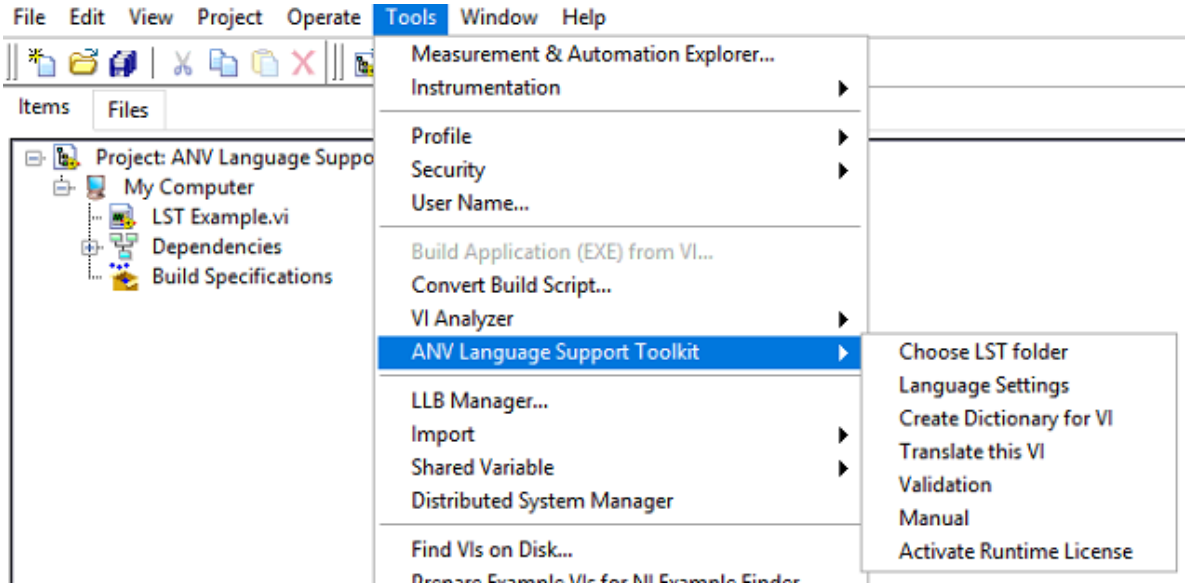


Fig. 7: LabVIEW tools menu with the toolkit installed

Choose LST folder

Each project, which is subject to translation has its own configuration folder. If there is no folder assigned to project which is currently being handled by ANV LST, user should perform this option first. Additionally, user can use this option to change folder which will be used for translation.

Language Settings

Translation of project can be performed in various languages. Language settings option allows user to define names of these languages. Additionally, this option allows user to select which of these languages is default language of the project (language of texts on user interface by default).

Create Dictionary for VI

After languages, which will be used for translation are chosen, user can create translation for VIs of choice by using this menu option. Texts, which are located on Front Panel of VI will be assigned to default language translation, while other languages will be left blank. User can fill the translation for all controls which are desired to be translated.

Translate This VI

After the tranlation for VI is created, user can verify whether it satisfies his requirements by using *Translate this VI* option. It will translate the VI into any of the languages, which were defined for the project. This option works only if selected from VI´s tools menu.

Validation

For ensuring that the translation is done correctly, user can validate all files responsible for translation in project. This includes validation of translatable VIs and configuration folders. After the project is validated, list of issues with their description is displayed. Any particular issue can be solved by taking appropriate action. User can select from these actions by right-clicking issue in list and choosing from among available actions.

Manual

Opens manual for the toolkit (this file)

Activate Runtime License

Launches activation wizard for deployment license. Please make sure that LabVIEW is running with administrator rights before activating deployment license, otherwise the activation will not work. For activation of development license, use Help => Activate Add-ons instead.

Toolkit settings

Choose LST Folder

This setting prompts user in File dialog window to select folder, which will be used as parent folder for "LST" folder. This selection is afterwards saved into Configuration file for LST folder location.

Language Settings

Language Settings UI

Language configuration window allows user to modify Language Configuration File. It provides user with the following options:

- Add language
- Edit language name
- Delete language
- Set default language
- Save configuration

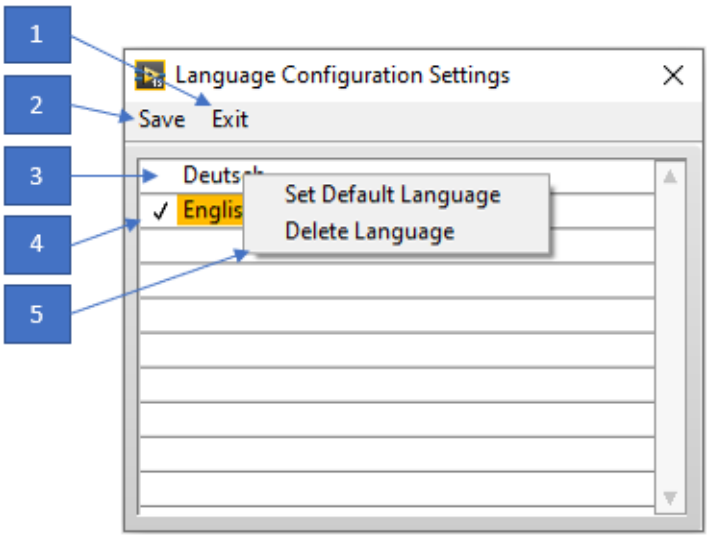


Fig. 8: Language Configuration Settings window

1. Exit option
2. Save option
3. Languages – languages, which will be used for translation.

4. Default language – checkmark sign besides a language indicates, that this language is default language of project.
5. Right-click context menu window – contains specific options based on selected item.

Add Language

Allows to create new language for configuration.

Edit Language

Allows to edit existing language. All language properties could be configurable. In order to keep consistency of the settings (after alias rename), language ID could be used.

Delete Language

Removes language from configuration file. If any translation files exist for given language, they become inactive. To delete a language from list, right-click given language and choose option "Delete Language" from menu.

Set Default Language

Sets language as default in Configuration file. This indicates to LST, that this language is default language in which Front Panel text of VI is written. To set a language as default, right-click given language and choose option "Set Default Language" from menu.

Save

This option saves all changes done to languages. It is performed by clicking corresponding menu item.

Exit

Close language configuration window without saving performed changes. To avoid accidentally exiting without saving and losing data, user is prompted to save changes when choosing this option.

Create Dictionary for VI

Translation Configuration UI

Configuration is started by call of "Create Dictionary for VI" menu item. This window is modal, in order to avoid simultaneous editing of configuration, and VI itself. VI is marked with bookmark tag (*#translatableVI*) in VI description automatically after saving configuration. This tag is used afterwards for configuration validation.

Features

Selecting VI for Translation

When configuration window is called by VI, it automatically loads data for translation from VI and from configuration files for VI, if such files exist.

In case when there is no configuration for VI, configuration file will be created for each language which is used translation.

In case when configuration file already exists, then information will be displayed correspondently based on current VI state/configuration file content.

Displaying Configuration in Hierarchical View

All objects and their properties in Configuration Window which are currently active are displayed hierarchically in Tree indicator.

Translating Items

Creating translation for a property is achieved by editing text in the corresponding row. Each column in Tree view corresponds to separate language, hence translation should be added to every column, which has language name in header.

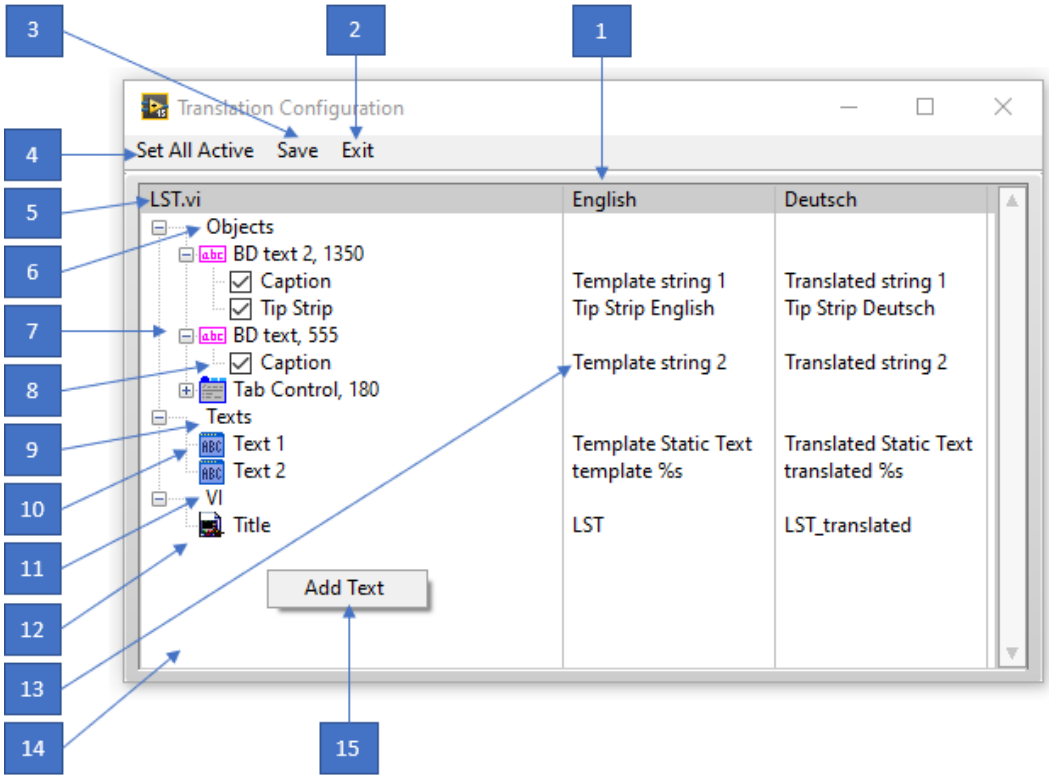


Fig.9: Translation configuration window

1. Language for translation - this column contains translations of properties to language in column header.
2. Exit – closes Translation Configuration window without saving changes.
3. Save – saves current configuration to all Translation files for VI which is being configured.
4. Set all active – sets all objects/properties active – adds all inactive objects to configuration window, to prevent user from forgetting to translate an object/property.
5. VI name – name of the VI, for which the translation is currently configured.
6. Objects – this section contains all front panel objects and their properties, which are currently set active.
7. Object – object, which is currently set as active is displayed here.
8. Property – a property for an object, which is currently set as active.
9. Texts – this section includes all Block Diagram text translations.
10. Text – particular Block Diagram text translation.
11. VI – this section includes VI specific translations.
12. VI item – translation of VI item.
13. Translation – translation text of item located in first column in the corresponding row into language located in column header of corresponding column.
14. Empty space below translations – right-clicking here allows to activate disabled objects/properties and add Block Diagram texts to translation.
15. Right-click context menu window – contains specific options based on selected item.

Activating/Disabling Translatable Properties

Configuration window allows to enable/disable objects and properties from translation. When an item is disabled, it is removed from Tree view and won´t be used for translation anymore. Conversely, activating an item adds it to tree and the toolkit will afterwards use given item for future translations.

To disable item, right-click item in Tree view and select disable option from available options.

To enable item, right-click empty area below the last tree item and select appropriate item from list.

By default, when VI´s translation is loaded for the first time, only common for objects properties are active, which are caption and tip strip.

Blacklisted Objects

Some control/indicator types are usually not interesting for translation; therefore, they are set as inactive by default (based on labels). Currently, class controls and all clusters with label containing "error" are blacklisted.

Block Diagram Text Translation Configuration

For each VI, it is possible to add custom Text Object for translation.

Each Text Object has custom ID (By default "Text X", where X is smallest integer index which is not already used as ID). User may change this ID. All translations are empty for newly added Text Object by default. After the translation is properly defined, it can be used for translation of text messages via API.

In case that text also consists of dynamical parameters which are not supposed to be translated (e.g. numbers), placeholders "%s" should be used for them. During translation, these placeholders are not translated, only text around them.

Examples of how this feature works can be found in ANV Language Toolkit Sample Project.

VI Title Translation

This is a separate property available for each VI. By default, VI title does not appear among active items, it is added in the same way as other items, by right-clicking empty area below the last item and selecting appropriate option from the list.

Modifications of VI

This UI does not modify the VI for which the translation is created, with the only exception of adding *#translatableVI* tag to the description. Therefore, changes which need to be performed in order to make translation visible to user (setting captions visible for indicators, allowing VI to use different Title bar during runtime) should be done manually.

Translate This VI

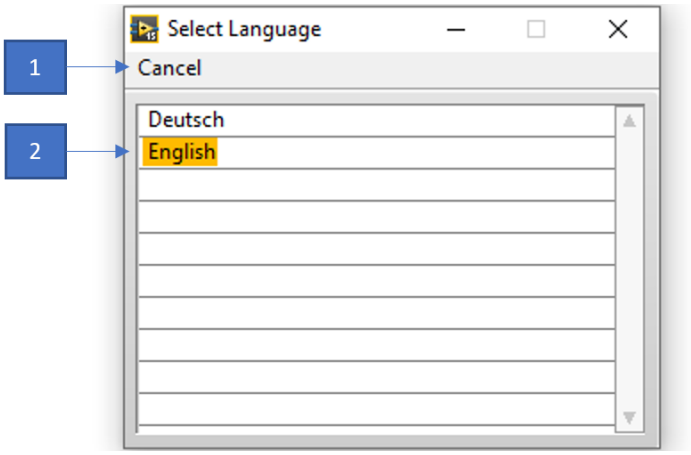


Fig. 10: Window for language selection in Translate this VI option

1. Cancel – closes Select language window.
2. List of languages – languages, which are available for translation.

This option is used for testing of translation for VIs, without API usage. It is called by selecting of "Translate this VI" menu item. All languages available for translation are listed here and when particular language is selected, VI is translated.

When VI is translated, changes are tracked and later undo-ed so that testing doesn´t modify VI in any way.

This window is modal in order to avoid simultaneous editing of VI itself and translating.

Validation Report UI

Validation is started by calling "Validation" menu item. It is project based, which means that all VIs marked as *#translatableVI* are being validated, as well as all configuration files located in folder for translation of the project.

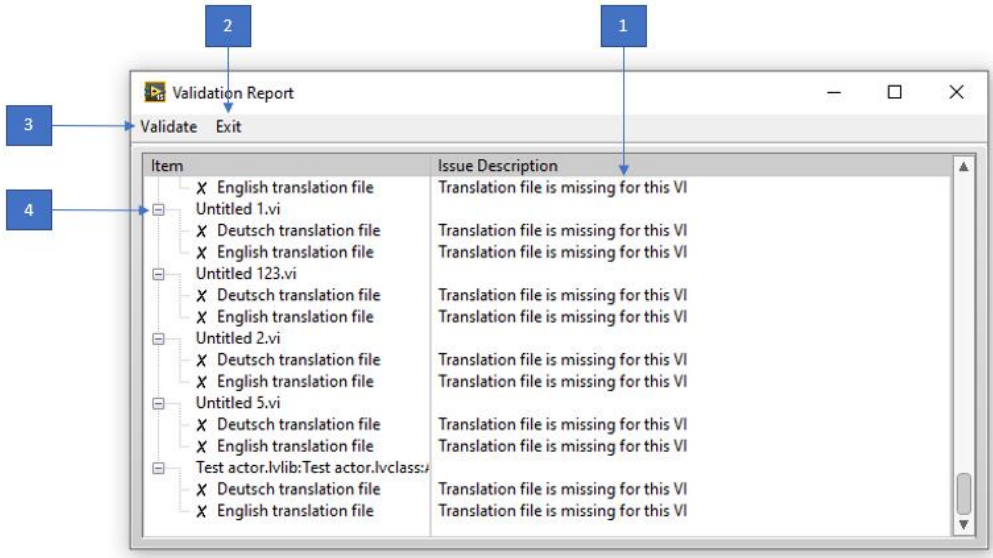


Fig. 11: Validation Report window

1. Issue description – describes details of issue which has occurred.
2. Exit – close validation window.
3. Validate – performs project-wide validation of LST files.
4. Item with issue – hierarchically displayed items, which are source of translation issues.

Context-Based Right-Click Menu

Right-click menu is context-based - it allows options which are suitable for fix of the particular issue.

Progress Bar

In case of projects with large number of files, validation process can take considerable time. Therefore, progress bar is provided, to give user approximate idea about the duration of validation process.

Validation Criteria

The following table contains possible issues which can be detected by Validation. Each of these issues has assigned methods for solution of issues, which are described in a separate part of document.

Table 1: Description of validation issues with possible methods for solution

Issue	Reported	Methods for solution
Missing configuration file	In case of missing Language Configuration File.json file.	Language Settings
No languages found in Configuration file	In case that Language Configuration File.json file exists but contains no languages.	Language Settings
No Active Language found in Configuration file	If there is no *Active language* present in Language Configuration File.json file.	Language Settings
No Default Language found in Configuration file	If there is no *Default language* present in Language Configuration File.json file.	Language Settings
Folder for language is missing	Reported if language is present in Language Configuration File.json, but folder for that language does not exist.	Create Folder
Traslation file is not in JSON format	If the .json file for translation of particular VI is not in .json format.	Open affected file(s) or folder(s), Delete Cfg File
Missing object on front panel	In case when object from translation list is missing on front panel.	Remove Object from Cfg, Open affected file(s) or folder(s), Open Editor
Configuration file exists, VI is not found	In case, when VI was translated and configuration file was created, but VI can't be found. It might happen after VI is renamed or moved.	Delete Cfg File, Relink Cfg File, Open affected file(s) or folder(s)
Translation file is missing for this VI	In case, when VI is marked by *#translatableVI* tag (in VI description), but configuration file is missing for the VI.	Open Editor
Translation for object (property) is missing for X language	When object (property) is present on Front panel of VI translation file but is not present in translation file.	Open Editor
Object (property) is in translation file for X language, but wasn´t found on Front Panel of VI	When object (property) is present in translation file but is not present on Front panel of given VI.	Open Editor
Translation is empty for X language	Property is found in configuration file, set as active, but its translation is empty, meaning that user forgot to translate particular property.	Open editor
Active flag varies between translation files.	In case, when active flag for object/property varies in configuration files in different languages.	Open editor

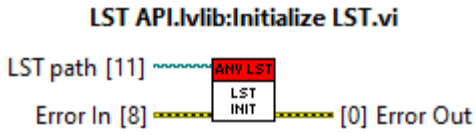
Activate Runtime license

This option is used for activation of deployment license on machines which will use executables with API VIs from the LST. It is called by clicking "Activate runtime license" menu item. Detailed information about activation of deployment license can be found in the "Licensing and activation" section of this manual.

Toolkit API

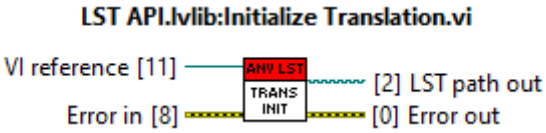
Toolkit API is used to programmatically translate VIs with existing translation configuration. After toolkit installation, API can be used from LabVIEW Functions palette located at ANV/ANV Language Support Toolkit. Following VIs are available to user:

Initialize LST



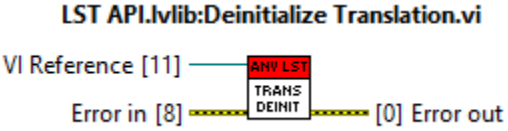
Initializes toolkit (path to LST configuration folder for given project), should be called before any other API VI. If no LST path is chosen, path of calling project will be taken as the default location for LST folder (\project folder\LST).

Initialize Translation



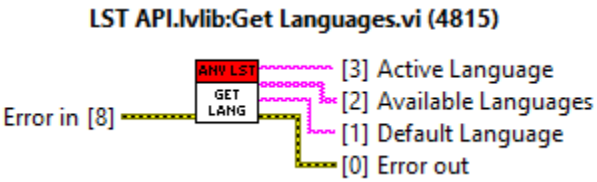
Initializes translation for a VI, should be called before translation for VI is desired (at VI start). If VI reference is wired, it initializes translation for given VI, otherwise, calling VI´s reference is used.

Deinitialize Translation



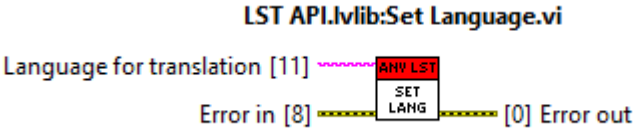
De-initializes translation for a VI, should be called when translation for VI is no longer desired (usually before closing VI). If VI reference is wired, it de-initializes translation for given VI, otherwise, calling VI´s reference is used.

Get Languages



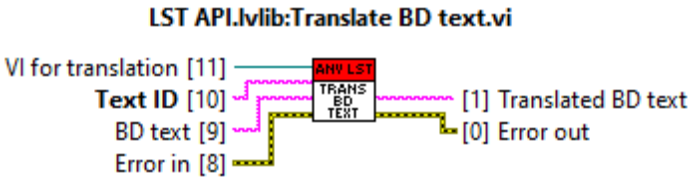
Returns list of languages, default and currently active language from configuration file.

Set Language



Sets active language of the project from all languages available for project. All running VIs, which have initialized translation, are automatically translated to this language.

Translate BD Text



Translates text on block diagram. Can be used for translation of texts, which are displayed in user interface, for example string indicators. As the inputs VI receives:

1. String with text ID, which is used for configuration in Dictionary configuration window (as described in the correspondent section).
2. Actual default BD text to translate.
3. As the output, VI has translated BD text.

If text consists of dynamic parameters (for example, there is used Format Into String function which will insert into string numbers, etc.) then these parameters should be marked as placeholders during configuration in Dictionary Configuration window by "%s".

If text is not found in dictionary configuration, or any errors occur during translation, text will be displayed without translation.

Creating executable with API

If the API should be used inside LabVIEW executable, activation of development license for the toolkit is necessary. Also, you need to distribute LST folder with all files and subfolders with the executable – it is up to the person creating executable, where this folder will be located. Make sure that this path corresponds with the one, which is used with "Initialize LST.vi" (see Toolkit API/Initialize LST section for more info).

To make API work inside executable, it is necessary to install deployment files. This is done with the use of runtime installer. After you install executable on your machine, perform following steps to ensure correct working of toolkit API in executable. Please note that is you have several applications using the API on the same machine, installation needs to be performed for every application separately:

1. Go to "National Instruments\LabVIEW 20xx\vi.lib\ANV\ANV Language Support Toolkit" folder on PC where LST is installed. (Full path may be for example "C:\Program Files (x86)\National Instruments\LabVIEW 2015\vi.lib\ANV\ANV Language Support Toolkit" for LabVIEW 2015).
2. Copy folder named "Executable" to machine which will use deployment license.
3. Run "LST runtime installer.exe". Dialog prompt appears. You are prompted to select application (executable) which uses toolkit API.
4. After the .exe is finished, all files necessary for the LST should be copied to your machine. Do not remove any of the installed files, otherwise your application may not work. Also, if you move your application into another folder, you may need to run the installer again.

Sample Project

Sample project is provided for the toolkit, to demonstrate API usage to user. Sample project is available from among Project Templates after installing toolkit for selected LabVIEW version. This Project contains single VI (LST.vi). Language configuration for this project is located inside vi.lib/ANV/ANV Language Support Toolkit/LST. Translation of this project is similar to real translations user can create with this toolkit.

Examples

Example project is accessible through LabVIEW example finder. Search for "language" and select "ANV Language Support Toolkit Example Project.lvproj" from the list of matches found. This project is similar to Sample project in this toolkit. Example project contains "LST example.vi", which demonstrates use of the API to user. Front panel of the VI contains instructions on how to use this example.

Configuration Files

This section contains information about configuration files used by ANV Language Support Toolkit. Purpose of configuration files is to store data related to settings and translation created by user.

Configuration Scope

Configuration relates to project. Each project requires its own configuration.

Configuration File Types

There are three types of configuration files:

Configuration file for language settings;

Configuration files for a VI;

Configuration file for LST folder location.

Configuration File Format

"Configuration files for language settings" and "Configuration files for a VI" were chosen to be in JSON format, in order to provide readable hierarchical structure of configuration files to user and also to allow them easily edit this configuration with common text editors. Configuration file for LST folder location is in INI file format.

Configuration Files Naming

Configuration file for a VI is saved under fully qualified name of VI (library name_VI name) + file format extension (.json).

If the VI name contains one of special characters such as comma (.) or asterisk (*), these are replaced by underscore (_).

Configuration file for language settings file is named as: "Language Configuration File.json".

Configuration file for LST folder location is named as: "LST paths.ini"

Configuration Files Location

The structure of configuration is hierarchical with following organization of files (example for translation with two languages and two translated VIs):

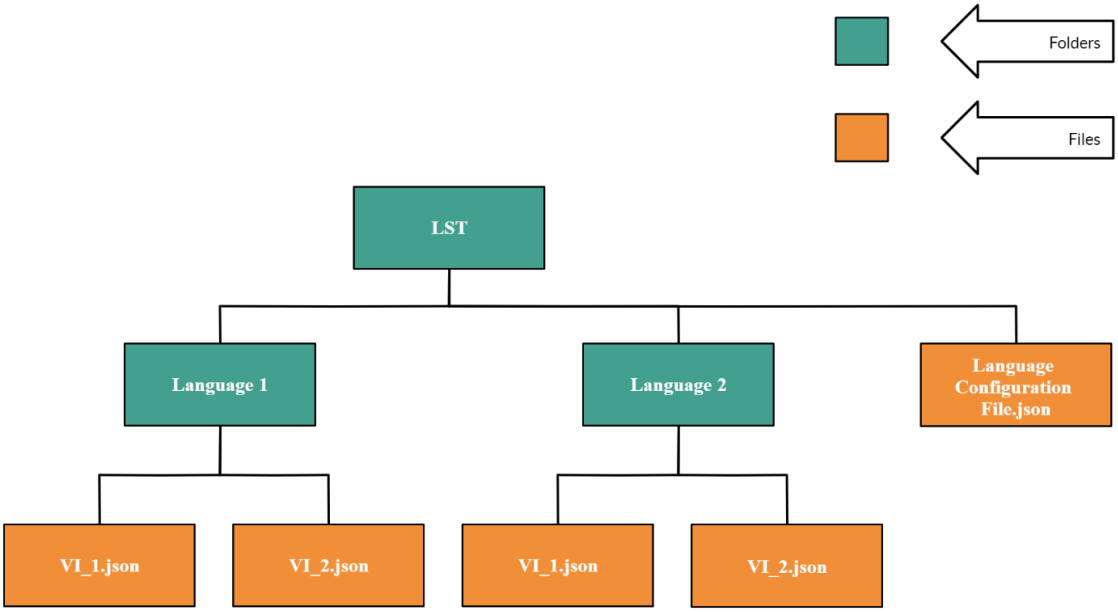


Fig. 12: LST Folder structure

Configuration files are saved in special separate folder. Folder contains subfolders for each translated language. Only exception is "Configuration file for LST folder location", which has location "LabVIEW20xx\vi.lib\ANV\ANV Language Support Toolkit\Configuration files\ LST paths.ini",

Main folder "LST" contains "Language Configuration File.json" and folders for each language, which is used in translation. Language folders contain configuration files for each VI, which is being translated.

Configuration File for LST Folder

Purpose of this configuration file is to store project names and their respective paths. When the path of LST folder is selected by "Choose LST folder" tools menu option, this path is saved into INI file as a name with the key, which is project path.

When the project is loaded later and another menu item from ANV LST tools menu palette is called, INI file is checked and if sthe project path is found by the corresponding key, LST path will be automatically assigned without the user having to manually re-select it.

This path is used only by the tools menu items (not by the API).

Language Configuration File

Contains the following settings:

1. Currently selected (active) language;
2. Default language, which is the language found in VIs by default;
3. Configuration for each language.

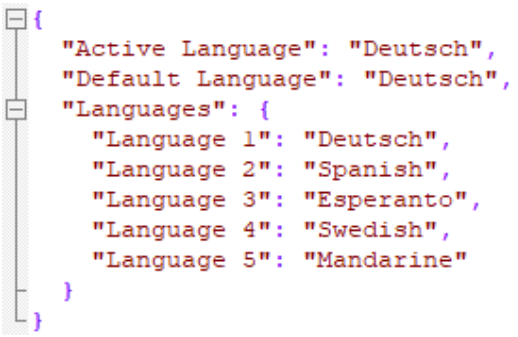


Fig. 13: Example of file with 5 languages and "Deutsch" language selected as both default and currently active language.

Configuration File for a VI

This configuration file is created for each translated VI + for each language.

There are 3 separate types of translated texts:

Objects – Front panel objects, i.e. Boolean or Numeric controls. Each object possesses several properties, which can be translated. Complete list of all translatable objects and properties can be found in separate section.

Texts – Block diagram text translation. Full description of how block diagram texts are translated can be found in separate section.

VI – VI specific texts. Currently VI Title bar translation is only item in this category.

In general, structure of configuration file is following:

Objects:

Object Name:

Translatable Property: values for different languages

Texts:

Text Index: Text for Block diagram text translation

VI:

Title: title bar text for VI

Structure of configuration file can be for example:

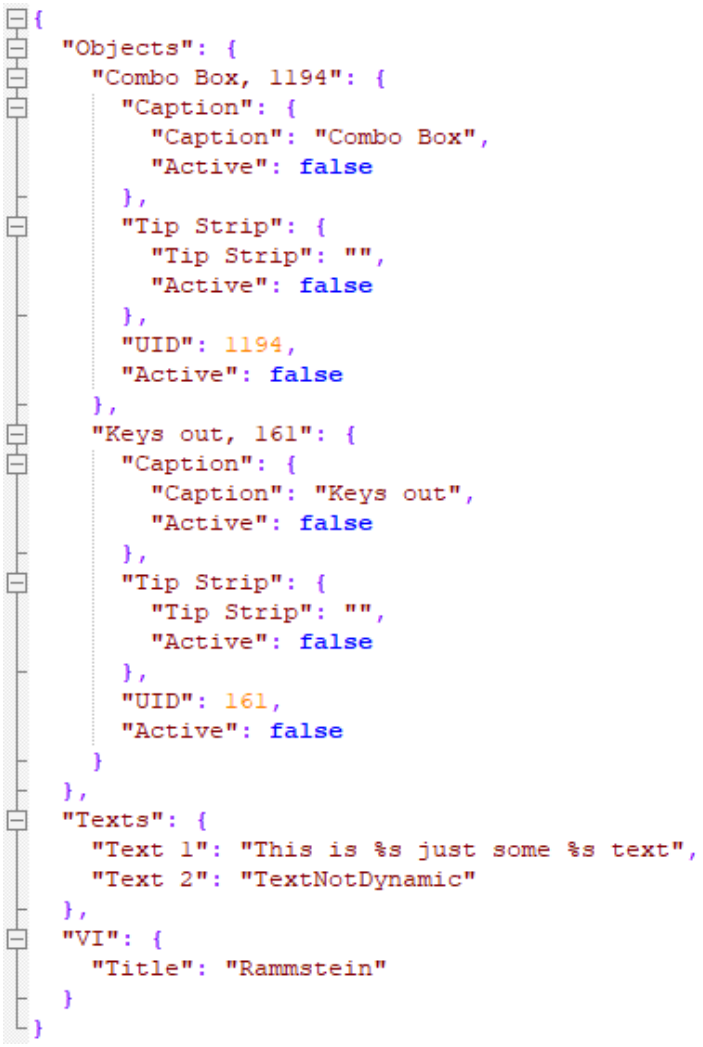


Fig. 14: Example of configuration file

Appendix

Objects and Properties Available for Translation

Following list summarizes all objects and their properties, which are available for translation in current version of ANV LST.

Table 2: Description of validation issues with possible methods for solution

Object icon in Translation Configuration window	Object name	Available properties
	Absolute Time	Caption, Tip Strip
	Array	Caption, Tip Strip
	Boolean	Caption, Tip Strip, Boolean Text
	Cluster	Caption, Tip Strip
	ComboBox	Caption, Tip Strip
	Enum	Caption, Tip Strip
	Graph	Caption, Tip Strip, Graph Axis
	Listbox	Caption, Tip Strip, Listbox Items
	Multicolumn Listbox	Caption, Tip Strip, Column Headers
	Numeric	Caption, Tip Strip
	Path	Caption, Tip Strip
	Ring	Caption, Tip Strip, Ring Item
	Slider	Caption, Tip Strip
	String	Caption, Tip Strip
	Tab	Caption, Tip Strip, Page Caption
	Table	Caption, Tip Strip, Table Column Header, Table Row Header
	Tree	Caption, Tip Strip, Tree Column Header