
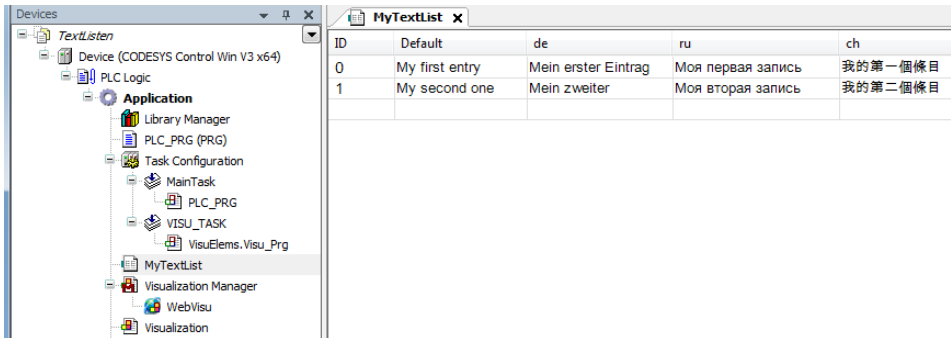


Access to Text Lists from the IEC Code

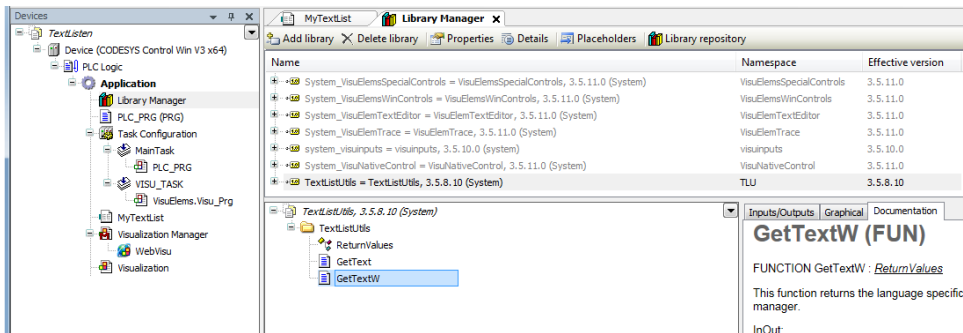
 The requirement for programmatic access to text lists is that a visualization is present in the project.

- Example of a text list:



ID	Default	de	ru	ch
0	My first entry	Mein erster Eintrag	Моя первая запись	我的第一個條目
1	My second one	Mein zweiter	Моя вторая запись	我的第二個條目

- Open the *Library Manager* and add the *TextListUtils* library.

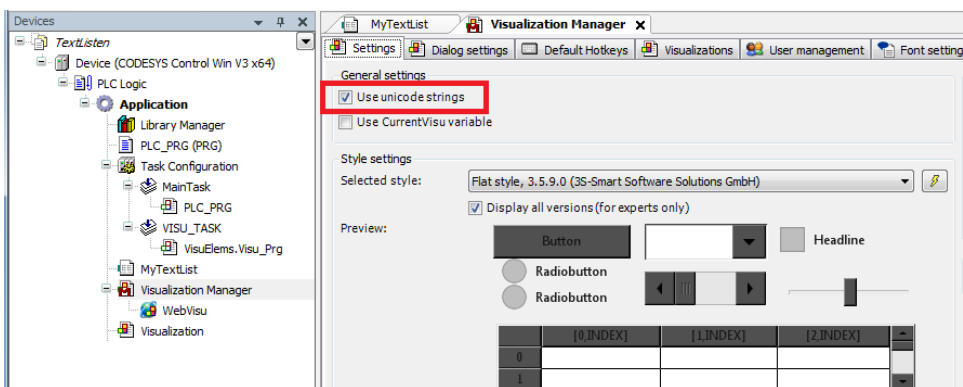


The screenshot shows the Library Manager window with the 'TextListUtils' library selected. The library is located under the 'TextListUtils, 3.5.8.10 (System)' folder. The 'GetTextW' function is highlighted in the list of functions.

Name	Namespace	Effective version
System_VisuElemSpecialControls = VisuElemSpecialControls, 3.5.11.0 (System)	VisuElemSpecialControls	3.5.11.0
System_VisuElemWinControls = VisuElemWinControls, 3.5.11.0 (System)	VisuElemWinControls	3.5.11.0
System_VisuElemTextEditor = VisuElemTextEditor, 3.5.11.0 (System)	VisuElemTextEditor	3.5.11.0
System_VisuElemTrace = VisuElemTrace, 3.5.11.0 (System)	VisuElemTrace	3.5.11.0
system_visuinputs = visuinputs, 3.5.10.0 (system)	visuinputs	3.5.10.0
System_VisuNativeControl = VisuNativeControl, 3.5.11.0 (System)	VisuNativeControl	3.5.11.0
TextListUtils = TextListUtils, 3.5.8.10 (System)	TLU	3.5.8.10

GetTextW (FUN)
FUNCTION GetTextW : ReturnValues
This function returns the language specific manager.
InOut:

Both of the functions **GetText** and **GetTextW** are differentiated only by the data type that is processed. If Unicode characters, such as Russian or Chinese, are used in the project, then you have to work with the **WSTRING** type. Please note that the use of Unicode characters has to be activated explicitly in the visualization manager



The screenshot shows the Visualization Manager window with the 'Use unicode strings' checkbox checked under the 'General settings' tab. The 'Style settings' section shows the 'Selected style' as 'Flat style, 3.5.9.0 (3S-Smart Software Solutions GmbH)' and the 'Display all versions (for experts only)' checkbox checked. The 'Preview' section shows a visual representation of the text list with columns for {0,INDEX}, {1,INDEX}, and {2,INDEX}.

- The language first has to be determined in the code before the text is read.

This is done here by means of the variable *udiLanguage* which can be changed from the visu, for example by means of a combo box.

Devices

Device [connected] (connected) Control Win V3 v1.0

PLC Logic

Application Manager

PLC Manager (PMS)

Task Configuration

MapTask

PLC Manager

Visual Task

Visual Demo Visu_Prog

MyTextList

Visualization Manager

Visual Demo

Visualization

Device Application: PLC_PMS

Expression	Type	Value	Prepared value	Address	Comment
udiLanguage	LOGINT	2			
id	STRING	"I"			
stResult	STRING	"not defined"			
valResult	WSTRING	"Missed parameter"			
retVal	RETURNVALUES	ERR_OK			

11

```
1  CASE udiLanguage OF
2    1: // English
3      OpDynamicTest.DynamicTestChangeLanguage(stLanguage := "Default");
4    1: // German
5      OpDynamicTest.DynamicTestChangeLanguage(stLanguage := "de");
6    2: // Russian
7      OpDynamicTest.DynamicTestChangeLanguage(stLanguage := "ru");
8    3: // Chinese
9      OpDynamicTest.DynamicTestChangeLanguage(stLanguage := "zh");
10  END_CASE
11 retVal := TIO.GetText(ipwText := ADDR(valResult), dilFile := SIBDF(valResult), stTextList := stTextList, stTextIndex := stIndex, stIndex := stIndex)
```