CONTROLS DESCRIPTION EDITOR

USER MANUAL

Contents

0. Introduction

Controls Description Editor (CDE) application is intended to provide a LV developer with a tool to retrieve edit and deploy descriptions of controls and indicators commonly referred to as controls.

Specifically, CDE allows editing following controls' properties:

a. Label (text)
b. Caption (text)
c. Caption Style (font)
d. Boolean text (font)
e. Tip Strip (text)
f. Description (text, Bold/Plain)

CDE have extensive context help. It makes easier to end user to learn the applications functionalities and reduces probabilities of error.

It seems the easier way to make user familiar with CDE is to explain its functionalities applying to an example VI (LANC.vi) which is located in Support folder where CDE was installed.

CDE is written as LabVIEW 2013 32 Bit application but it works with LabVIEW 2013 64 Bit code.

Please be extra cautious: do not attempt editing 2012 Vis – it will compile them into LabVIEW 2013 version!

1. Acronyms and Abbreviations

LabVIEW - software development environment by National Instruments (NI)
VI – Virtual Instrument, the common term used to refer to segment of code or application developed with LabVIEW.
Target VI – VI on which editing process will be executed.

2. System Requirements

The CDE is build for Windows XP and up to Win 7. It was not tested for Windows 8 version thus its performance is not guaranteed.

The CDE has been developed In LabVIEW 2013 32 Bit version however it will be functional in editing LabVIEW 2013 64 Bit Vis.

3. Installation

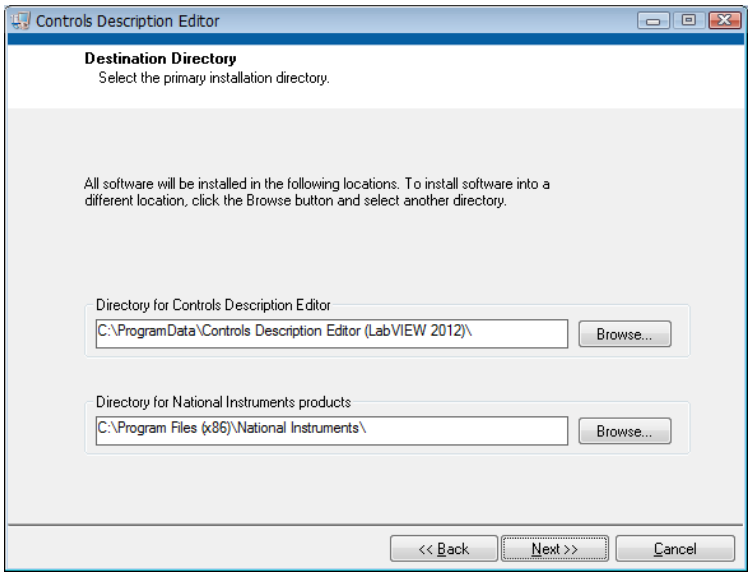
The installation process is identical to installation of other LabVIEW applications and sufficiently intuitive. However for unfamiliar users below are steps to install CDE:

3.1. Download and save Controls Description Editor Installer.

3.2. Go to the Controls Description Installer Ver.1.0.00>Volume> setup.exe.

3.3. Launch setup.exe file. In the case of Win 7 there will be warning that changes are about to be made on your computer. Press “Yes”.

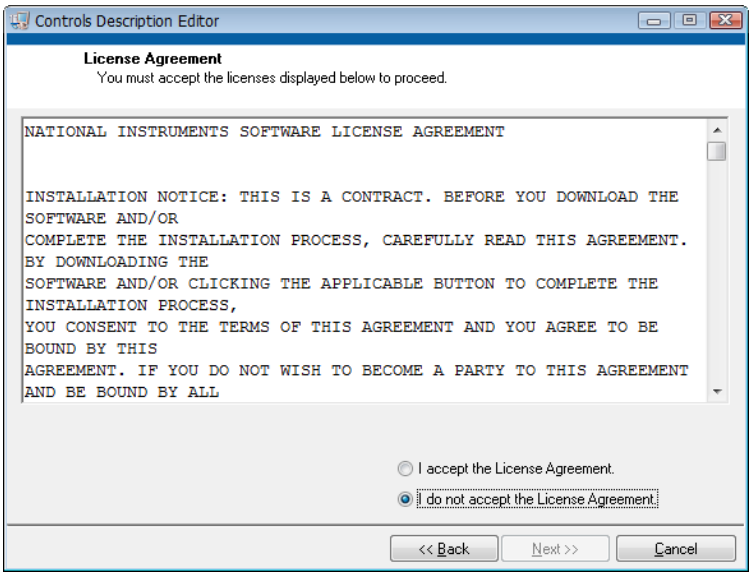
3.4. After a quick flash NI installation window you will see the Destination Directory pop up window:



Since the CDE is 32 bit application and machine is 64 bit OS the default destination directory is Program Data and for NI products “Program Files (x86)\National Instruments.

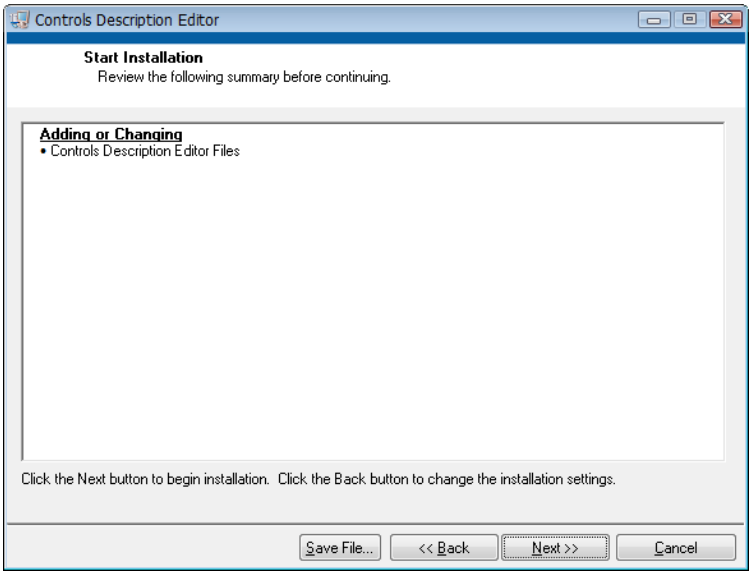
At this moment you may change the destination directories as you may wish but, please, be aware of Win 7 security restrictions.

3.5. The standard NI Software License Agreement will appear:



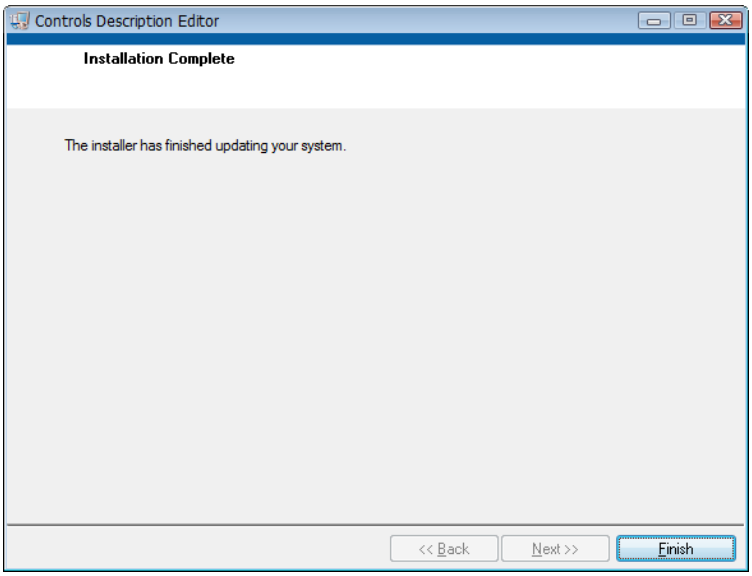
To proceed with installation you have to accept its terms and click “Next”.

3.6. After that the application is ready to be installed:



Click “Next” to continuo installation.

3.7. After Installation Complete window will pop up:



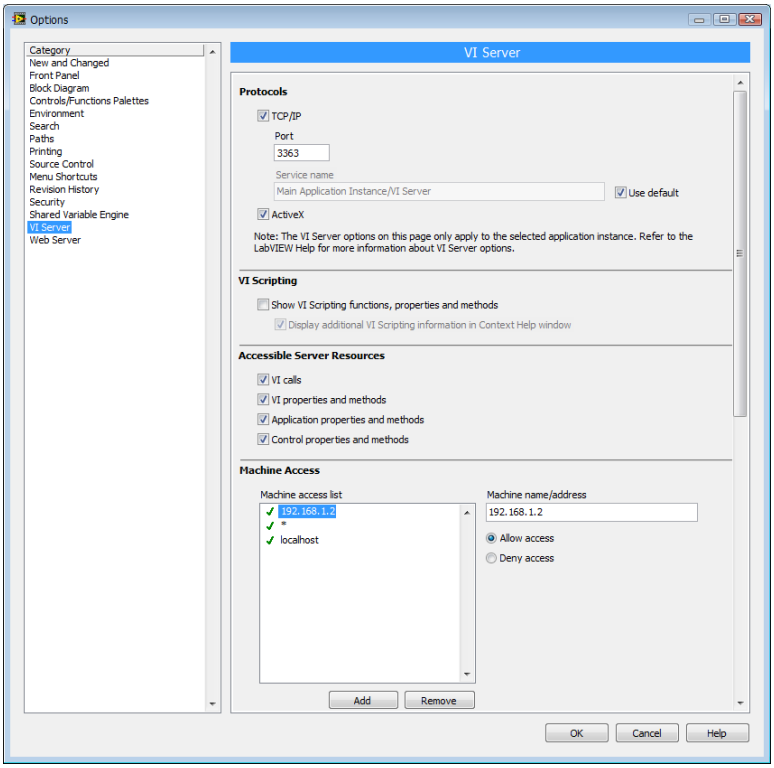
Click “Finish”.

4. LabVIEW Environment Requirements

CDE is a stand-alone application (executable) which communicates to a Target VI via VI Server utility. Because of that, VI Server should be configured per following requirements:

a. Protocols – TCP/IP and ActiveX enabled.
b. Available Server Resources - VI Calls, VI properties and methods, Application properties and methods, Control properties and methods should be enabled.
c. Machine access list should include “*” and localhost.

Please use the picture below as a reference:



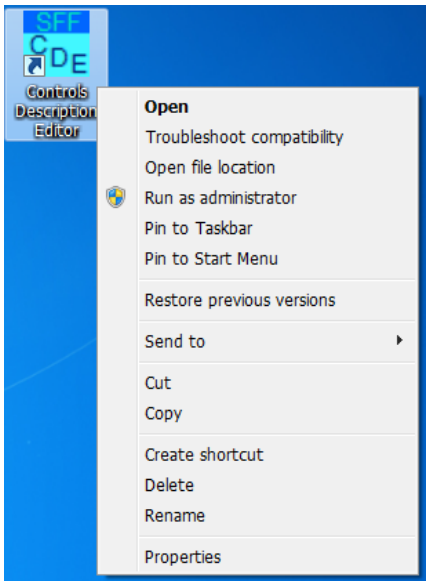
VI Server recommended configuration.

5. Application Start

The properly installed CDE application should create an icon on the Windows Desktop:

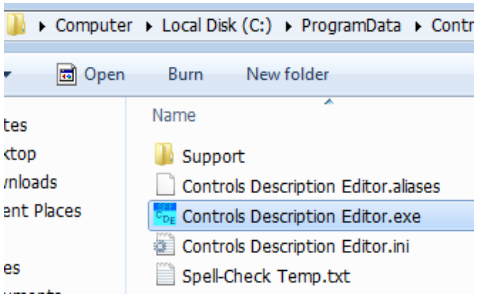
CDE Shortcut icon.

Before starting the CDE application, please, make sure LabVIEW 2013 is being opened. Right click on the icon and chose “Open file location” option:

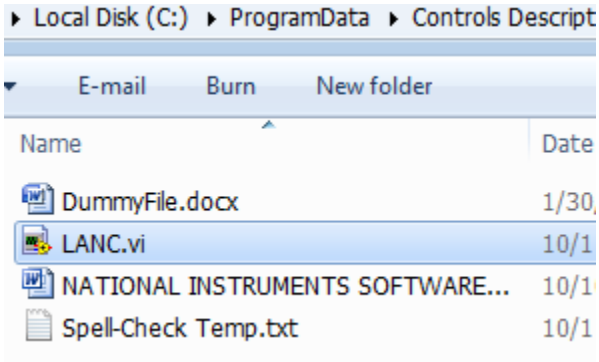


Shortcut icon options

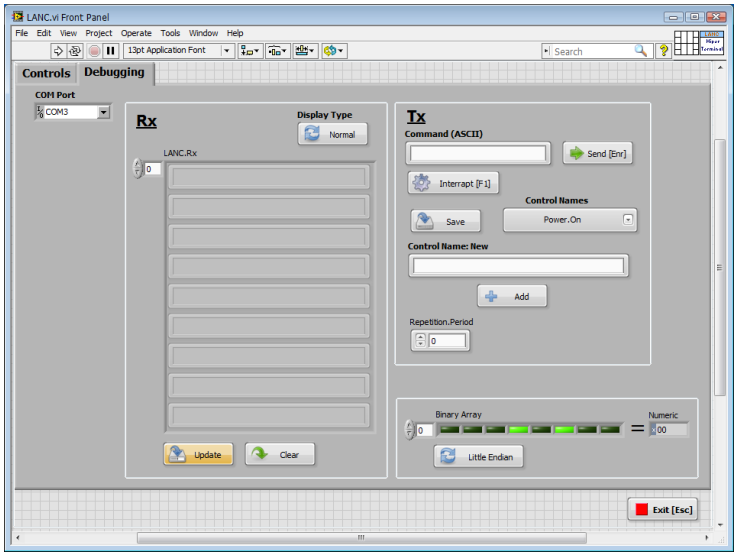
This action will open folder where CDE is installed:

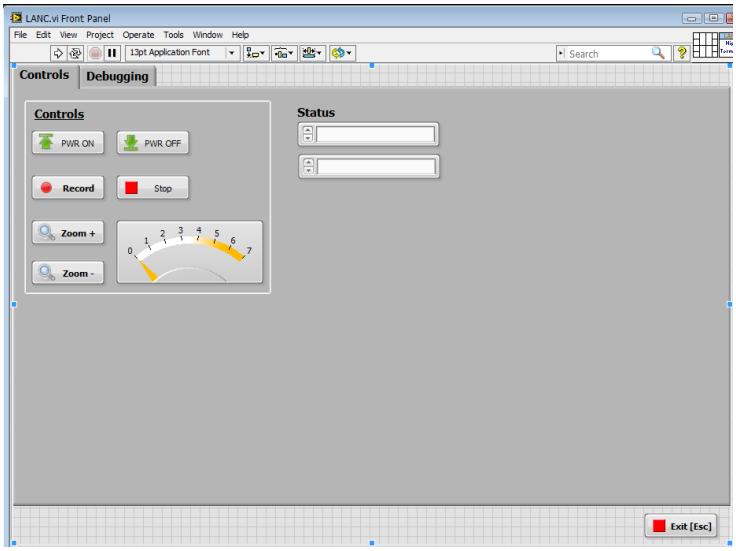


and, further, open “Support” folder:



Now, please, open LANC.vi which will be used as an example in the further discussion:

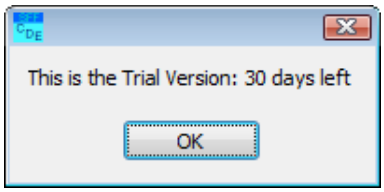




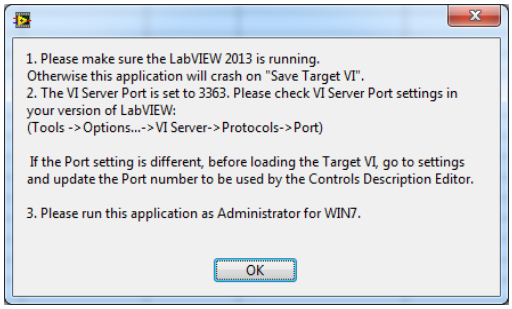
Let's examine this VI from prospective of GUI objects:

It contains Tab Control, Boolean Controls, String Controls, Numeric Controls and clusters (error cluster, in this case).

Now we are ready to open Controls Description Editor – please, click on the CDE shortcut icon.

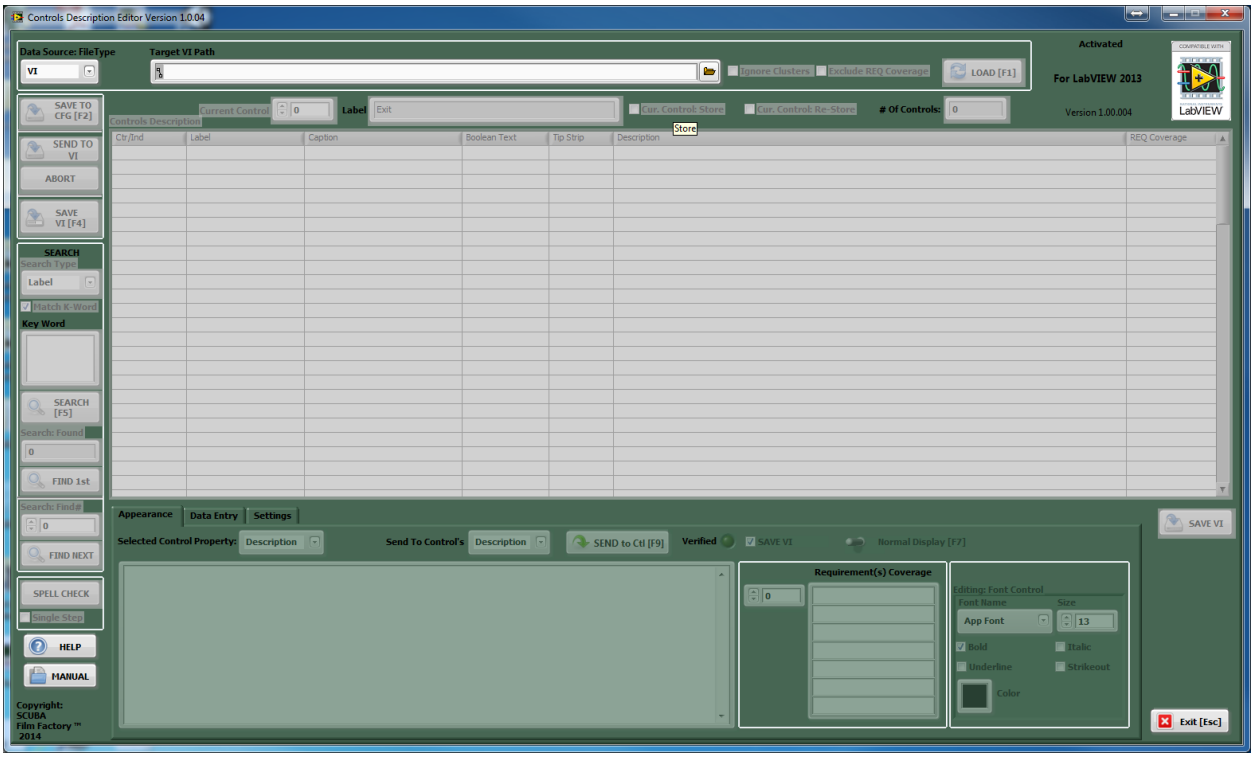
If you have installed a trail version there will be dialog window: 

notifying you how many days left for this version. The trail period is set to 30 days. After clicking "OK"

there will be another message: 

remaining you about required settings to be made. It seems to be annoying but it is better than sudden failures.

And finally the whole application will be in normal running mode:



6. Loading Target VI

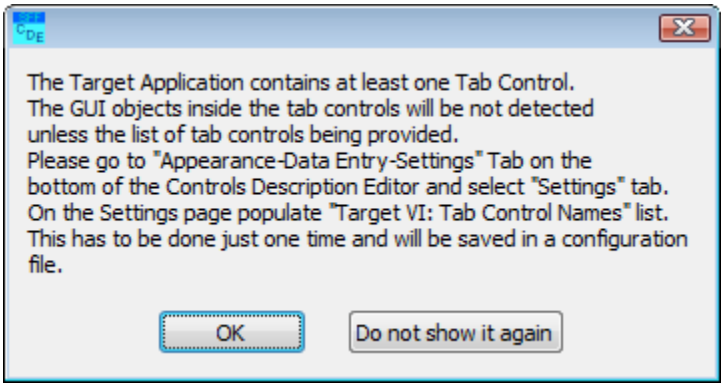
The very top section of CDE is dedicated to the loading the Target VI, in this case we will be editing LANC.vi we have opened earlier.

Please make following selections:

1. Data Source: File Type – set to VI;

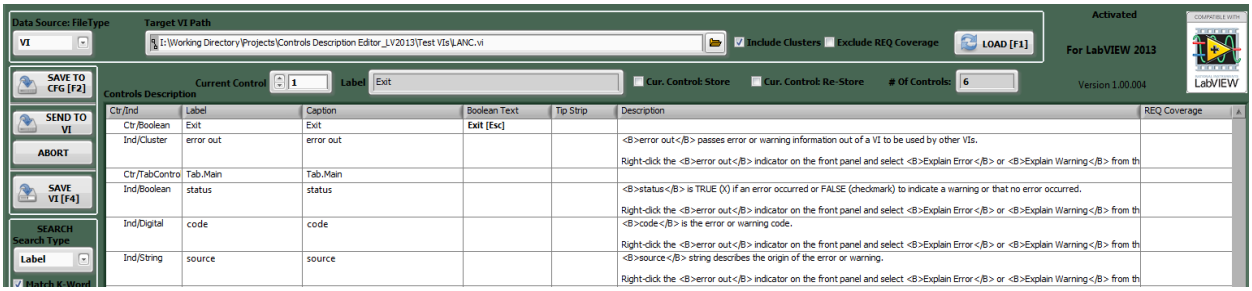
2. Data Source: Path – since the File Type is set to VI it is the path to the target VI – navigate to the LANC.vi;
3. Ignore Clusters checkbox – check it, so, it will display “Include Clusters”.
4. Exclude REQ Coverage – leave it unchecked.
5. Press “LOAD [F1]” button.

While loading, there will be message displayed:



this informs you the target VI contains at least one Tab Control. The significance of it in the fact that tab controls as well as clusters are containers and their content is hidden from VI Server unless they internal references are exploited. Please read carefully this message because there is clear instructions what should be done. But, at the moment, please, press “OK” button.

Now the Controls description List Box got populated:



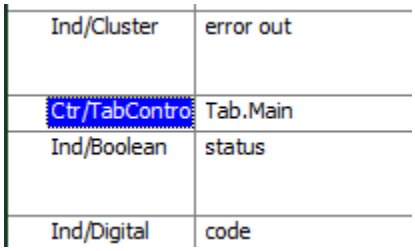
The “# Of Controls” indicator counts only 6 controls. It is obvious there is more than that. What happening is that “error out” cluster was “exploded” and contained indicators are now available for editing:

Ctr/Ind	Label	Caption
Ctr/Boolean	Exit	Exit
Ind/Cluster	error out	error out
Ctr/TabContro	Tab.Main	Tab.Main
Ind/Boolean	status	status
Ind/Digital	code	code
Ind/String	source	source

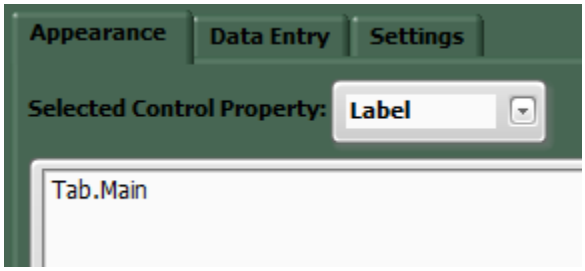
This is a valuable feature letting you edit context help and appearance not just a cluster itself but its contents as well.

Now, we have to access controls which are inside the Tab Control. Let's do following:

1. Select "Tab.Main" raw in the list box:



2. In the editing field set "Selected Control Property" – select "Label":



3. Highlight and copy (Ctl+C) text "Tab.main".
4. In the Editing Tab (the same as above) go to the page "setting" and paste (Ctl+V) copied label text into "target VI: Tab Control Names, than press "SAVE" button.
5. Uncheck "Include Clusters" checkbox since we are not interested in editing "error out" cluster and press "LOAD [F1]" button again.

   Now the Control Description list box is populated with all controls existing in the target VI:

Current Control 1 Label Tab.Main Cur. Control: Store Cur. Control: Re-Store # Of Controls: 28 Version 1.00.00

Controls Description

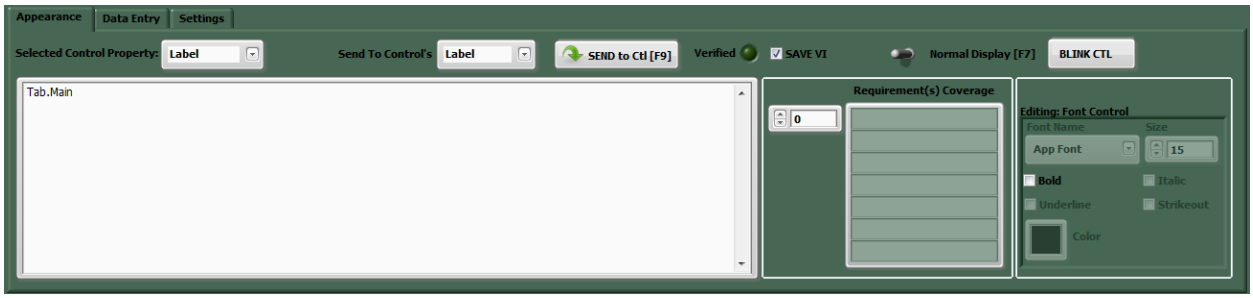
Ctr/Ind	Label	Caption	Boolean Text	Tip Strip	Description
Ctr/Boolean	Exit	Exit	**Exit [Esc]**		
Ind/Cluster	error out	error out			<B>error out</B> passes error or warning information out of a VI to be used by other VIs. Right-click the <B>error out</B> indicator on the front panel and select <B>Explain Error</B> or <B>Explain Warning</B> from th
Ctr/TabContro	Tab.Main	Tab.Main			
Ctr/Boolean	Record.Start	Record: Start	**Record**		
Ctr/Boolean	Record.Stop	Record.Stop	**Stop**		
Ctr/Boolean	Zoom.In	Zoom.In	**Zoom +**		
Ctr/Knob	Zoom.Speed	Zoom.Speed			
Ctr/Boolean	Zoom.Out	Zoom.Out	**Zoom -**		
Ctr/Boolean	Power.On	Power.On	PWR ON		
Ctr/Boolean	Power.Off	Power.Off	PWR OFF		
Ctr/Ring	Rx.Status.Byte0	**Status**			
Ctr/Ring	Rx.Status.Byte1	Status			
Ctr/Boolean	Update	Update	Update		
Ctr/String	Command.ASCII	Command (ASCII)			
Ctr/Boolean	Command.ASCII.Send	Command.ASCII.Send	Send [Enr]		
Ctr/Boolean	Clear	Clear	Clear		
Ctr/Boolean	LANC.Rx.DisplayType	Display Type	Normal		
Ctr/Array	Binary Array	Binary Array			
Ind/Digital	Binary Array.NumericVal	Numeric			
Ctr/Boolean	Interrapt	Interrapt	Interrapt [F1]		
Ctr/VISAResou	COM Port	COM Port			
Ctr/Boolean	Big/LittleEndian	Big/LittleEndian	Little Endian		
Ind/Array	LANC.Rx	LANC.Rx			

As we can see, there is only one control – "error out" cluster has description associated with and there are 28 controls altogether, excluding error out cluster contents. At this moment we are ready to edit controls of the Target VI.

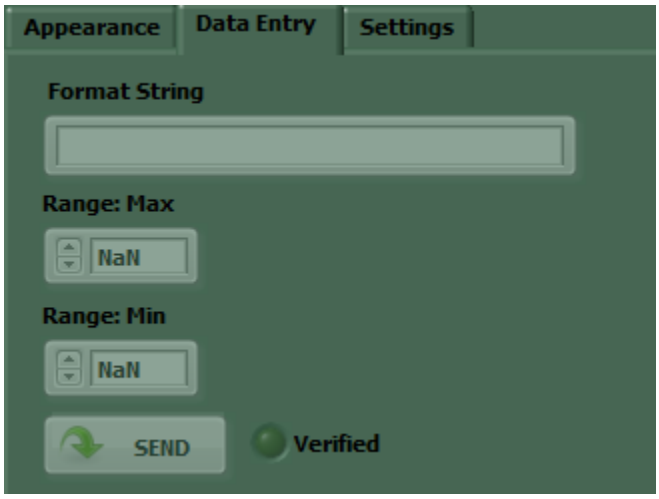
5. EDITING

The editing functions are located in two pages:

1. Appearance:



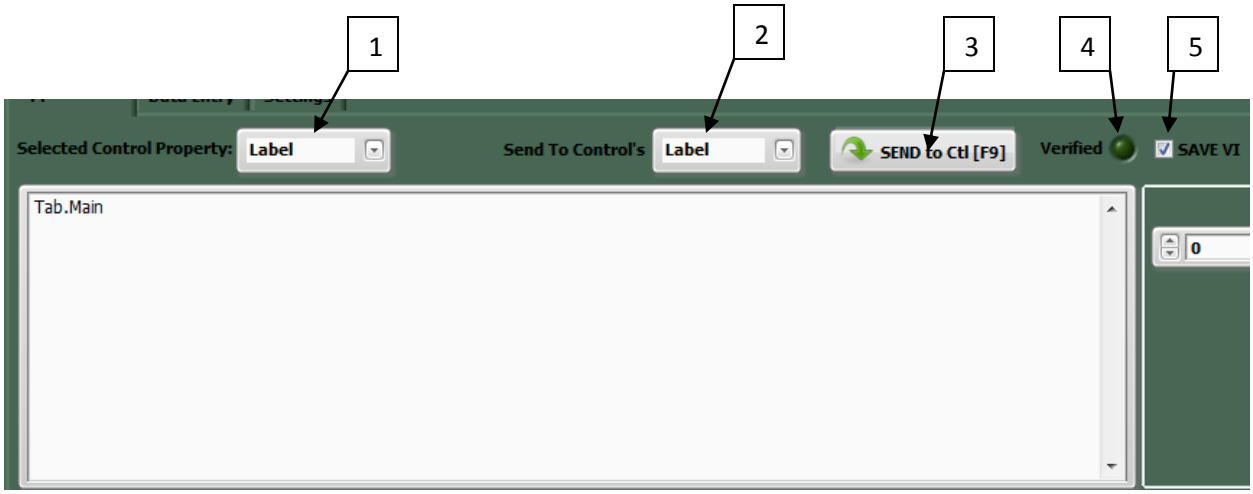
2. Data Entry:



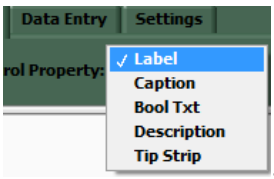
3.

5.1. Editing Appearance

The Appearance page contains following controls:

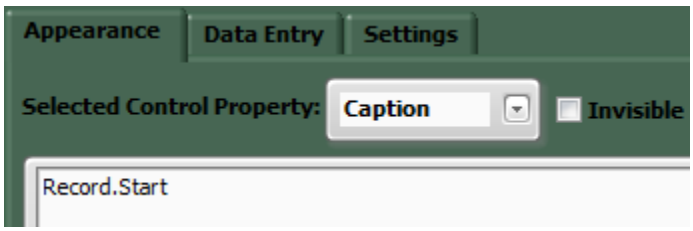


5.1.1. Selected Control Property

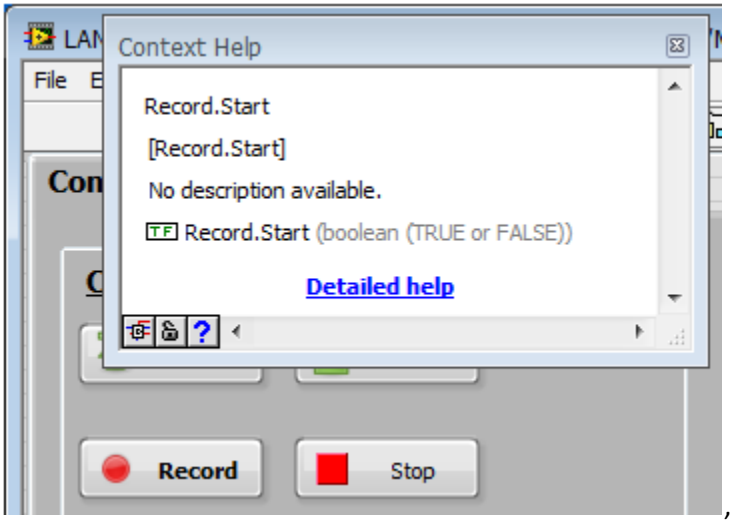
This menu ring list properties which can be edited: 

.

Please select from the Controls description list box “Record.Start” control and from the Selected Control Property list – Caption:

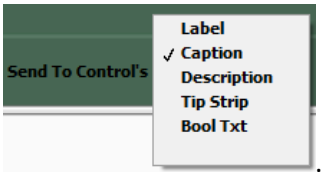


Please note, the checkbox “Invisible” appeared and it is unchecked. It means the Caption property of the selected control (Record.Start) is invisible. When application is in development mode and you will display content help (Ctl+H) there will be label and caption both displayed:



, where [Record.Start] is the label. When application is being built into executable, the only caption will be displayed. It is reasonable to make captions more readable and title-like, in a bold font style. To do that, modify the text to Record: Start.

5.1.2. Send To Control’s

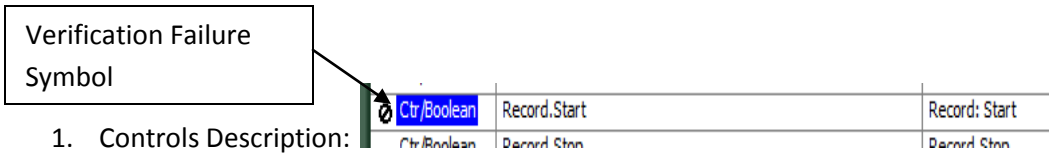
This control defines destination where text can be sent: 

. By default it duplicates the “Selected Control Property” selection but user can modify destination at will. In this demo case we will send the new caption text to control’s caption property.

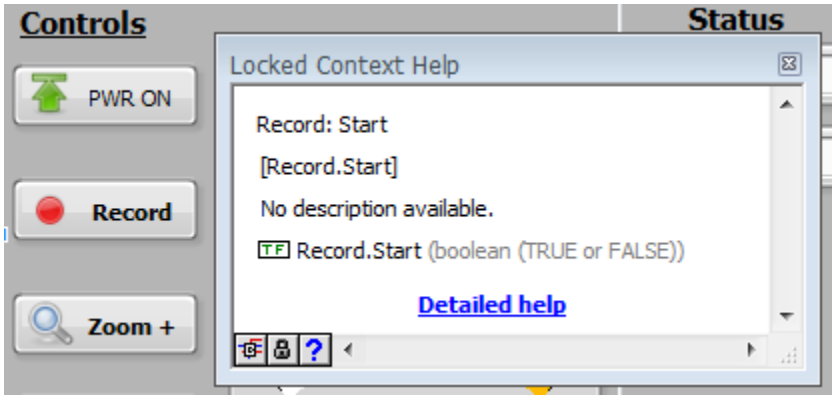
5.1.3. SEND to Ctl [F9]

Pressing this button will write a text to the selected control property.

Press this button and note following changes:

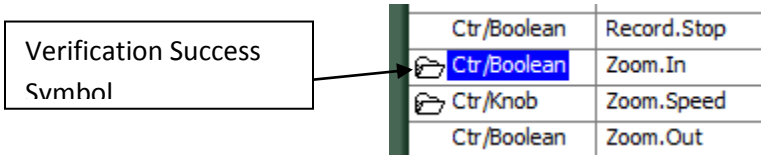


1. Controls Description:
   The caption is changed from Record.Start to Record: Start.
2. In the LANC.vi, press Gtl+H (context help) and put cursor over "Record" button, you should observe:



5.1.4. Verify

The "Verify" indicator provides information if "SEND to Ctl" operation was successful. It reads control property from the Target VI and compares with sent text.



5.1.5. Save

The Save checkbox specifies either to save the Target VI after control property was modified. The reason why save was not hardcoded is that if the Target VI is very large, the save process can be very time consuming. In this case user may be willing to save periodically by pressing "SAVE VI" rather than saving after each update.

5.1.6. Requirement(s) Coverage

This function is made specifically to assist users with integration GUI objects with NI Requirements Gateway application. For selected "Record.Start" control, select control property "Description" and enter whatever description you may want, for instance:

Starts recording images to a camera's media.

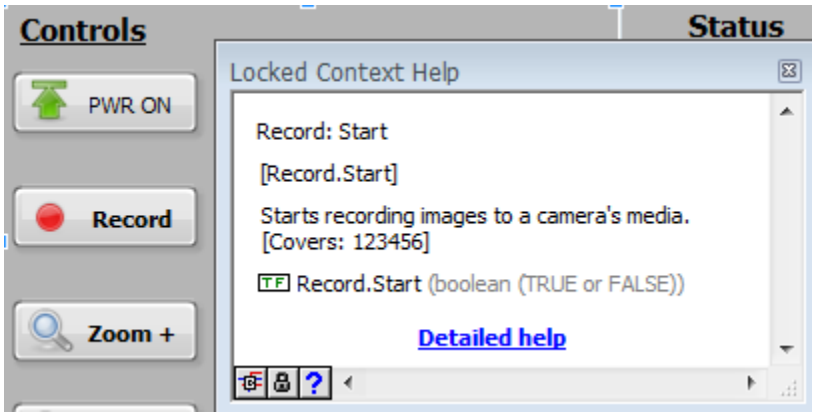
Now, enter into Requirement(s) Coverage fictitious number – 123456, for instance.

Press "SEND to Ctl [F9]" button and observe following changes:

1. Controls Description list box should display:

were in the REQ Coverage column requirement ID 123456 appeared.

2. Please cursor over "Record" button of the LANC.vi and observe:

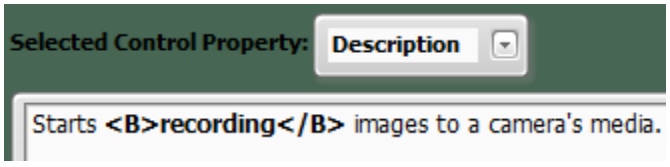


Requirement specification string "[Covers: 123456]" is being inserted into description.

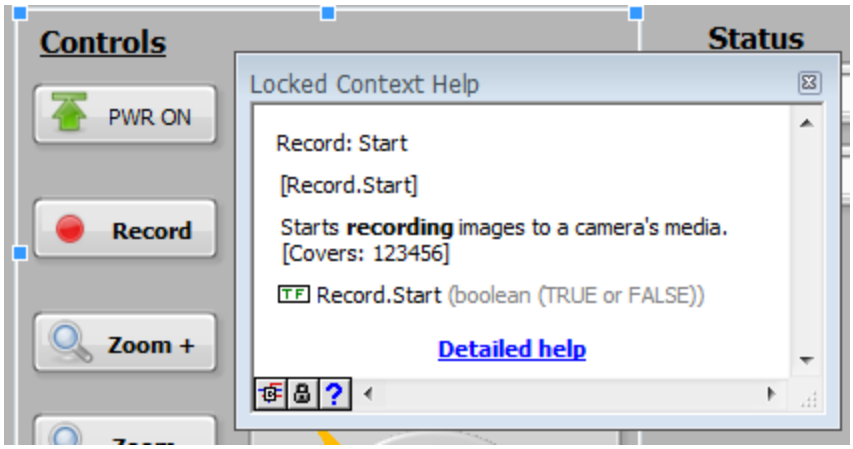
5.1.7. Font Control

Different properties accommodate different font specifications. For instance, description text or its portion can be bold or plain. The Tip Strip can be plain only. Depending from the property selected, various font options become either enabled or disabled.

To demonstrate is, highlight the world "recording" in the description test and check "Bold" box. The world will be in the bold style and HTML tags will be inserted:



Send the description to the control and observe:



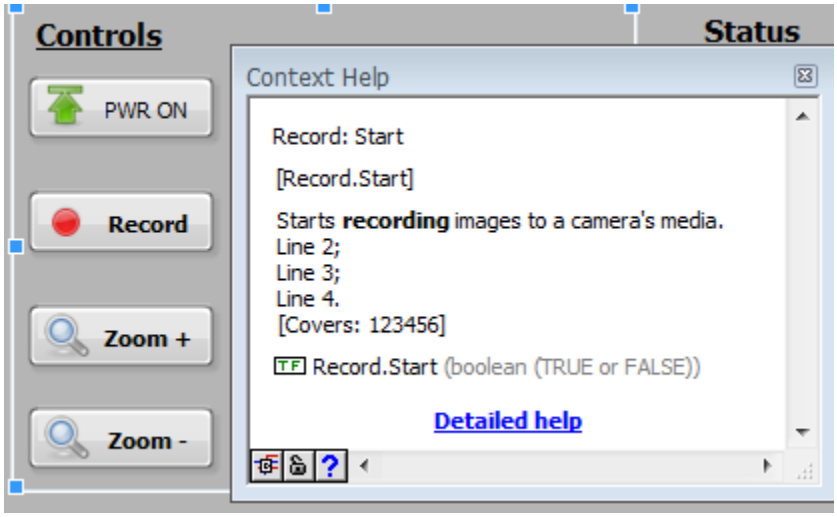
5.1.8. Multiple lines in description.

If you will try to make control's description extensive you will need to split the text into paragraphs. It is impossible within control properties – Description, unless using HTML tags.

CDE allows doing it with ease.

Modify the description text by adding extra lines:

And send it to the control:

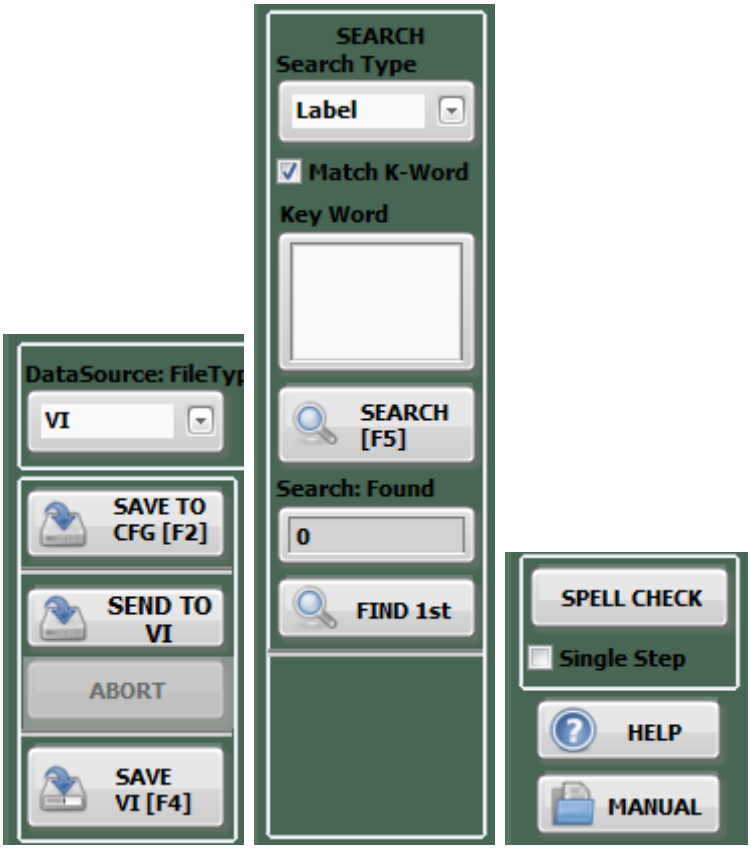


5.1.9. Data Entry page

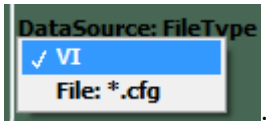
Controls on the Data Entry page are for modifying format string and few other properties of the numeric controls.

6. Additional Options

The Controls description Editor has additional option useful for the Target VI editing:

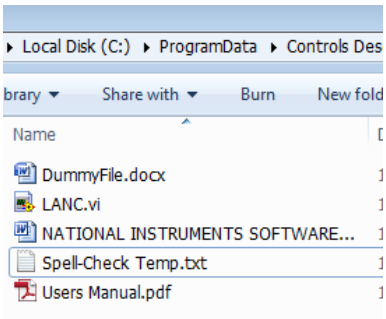


6.1. Data Source: File Type

Editable control properties can be loaded either from the Target VI or from its associated configuration file: 

.

The first time the property was sent to the Target VI, the configuration file will be created with the same name but with “cfg” extension. To illustrate this, let’s take a look on the Controls Description Editor (LabVIEW 2013)>Support directory:



Please note there is LANC.vi which was used as the Target VI in previous sections of this document.

Please perform the following steps:

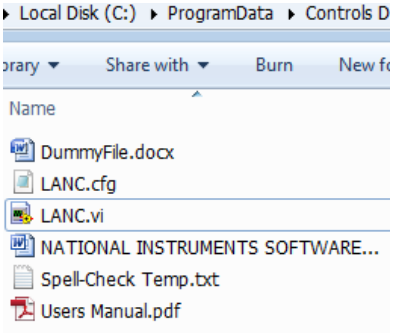
6.1.1. Select “Data Source: File Type” as VI and navigate to the LANC.vi.

6.1.2. Press the “LOAD [F1]” button and follow prompts on the screen.

6.1.3. Verify the Target VI editable properties are loaded and displayed in the “Controls Description” table.

6.1.4. Press 

button.

6.1.5. Navigate to the ‘Support” folder: 

and note there is LANC.cfg file being created.

6.1.6. Open the LANC.cfg file with the Notepad, for instance and examine its contents: In this example they will be:

```
[TabControlNames]
TabControlNames = "Tab.Main\0D\0A"

[Exit]
Ctr/Ind = "Ctr/Boolean"
Label = "Exit"
Caption = "Exit"
Boolean Text = "Exit [Esc]"
Tip Strip = ""
Description = ""
Requirements = ""
Font.Label = "\00\00\00\06Tahoma\00\0D\00\00\00\00\00\00\00\00"
Font.Caption = "\00\00\00\06Tahoma\00\0D\00\00\00\00\00\00\00\00"
Font.Boolean Text = "\00\00\00\08App Font\00\0D\01\00\00\00\00\00\00\00"

[error out]
Ctr/Ind = "Ind/Cluster"
Label = "error out"
Caption = "error out"
Boolean Text = ""
Tip Strip = ""
Description = "<B>error out</B> passes error or warning information out of a VI to be used by other VIs.\0A\0ARight-click the <B>error out</B> indicator on the
front panel and select <B>Explain Error</B> or <B>Explain Warning</B> from the shortcut menu for more information about the error."
Requirements = ""
Font.Label = "\00\00\00\06Tahoma\00\0D\00\00\00\00\00\00\00\00"
Font.Caption = "\00\00\00\06Tahoma\00\0D\00\00\00\00\00\00\00\00"
Font.Boolean Text = "\00\00\00\06Tahoma\00\0D\00\00\00\00\00\00\00\00"
```

6.1.7. Please note there are multiple section corresponding to all controls of the LANC.vi.

6.1.8. Do not close this file because it will be used in the following sections.

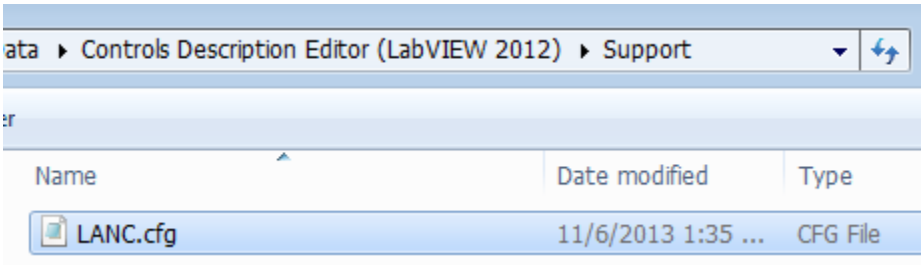
6.2. Loading properties from the configuration file.

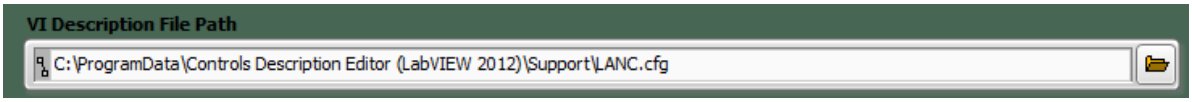
The controls properties can be loaded from the configuration file instead of the Target VI. The benefit of it is that the text file can be edited and then imported into the Target VI. To demonstrate it, please, follow the steps below:

6.2.1. Close the Controls Description Editor and re-open it again. It should be “blank”.

6.2.2. In the “Data Source: File Type” menu ring select 

and navigate to the LANC.cfg file:





6.2.3. Load the file and observe the “Controls Description” table got populated with all editable (by CDE) properties.

6.2.4. Go to the LANC.cfg file and modify some key values. For the demo purpose we will make following changes:

```
[Record.Start]
Ctr/Ind = "Ctr/Boolean"
Label = "Record.Start"
Caption = "Record.Start"
Bold = "Bold"
Boolean Text = "Record"
Tip Strip = "Starts recording"
Description = "Starts recording"
```

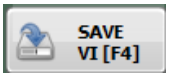
. Save the LANC.cfg file.

6.2.5. In the Controls description application press the “LOAD [F1]” button again and observe that the Tip Strip and Description of the “Record.Start” control changed to “Starts recording”.

6.3. Sending CFG file data to the Target VI

In order to import properties to the Target VI press the 

button. However, please, keep in mind the Target VI will be not saved automatically. The reason is that erroneous data can be sent and you may need to verify correctness of it before saving the Target VI.

If you are satisfied with the verification you may save the Target VI either pressing 

or 

button.

6.4. Search features

The Controls Description Editor features efficient search options. In large projects there may be difficult to locate targeted control. For this reason you can find control/controls using the whole word or key-word: 

or 

.

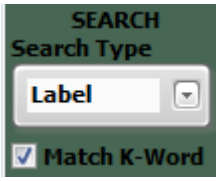
You can search for it in 

> 

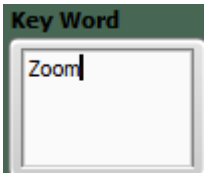
.

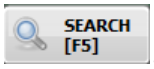
To demonstrate the search feature please follow steps below:

6.4.1. Goal: to locate controls which have 'Zoom" as the part of their label.

6.4.2. Select "Search Type" 

and check "Match K-Word".

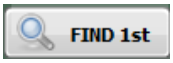
6.4.3. Enter "Zoom" into Key Word field 

and press "SEARCG [F5]" button 

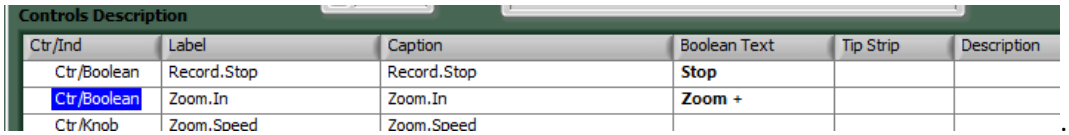
.

6.4.4. The search results will be populated: 

. There are 3 controls which label contains the word "Zoom".

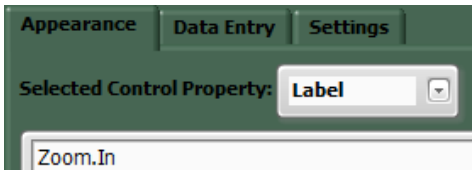
6.4.5. Press 

button and this control will be highlighted in the Controls Description table:

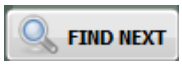


.

The properties will be transferred into “Appearance” tab

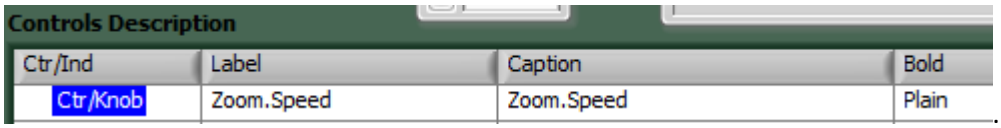


where editing can be performed.

6.4.6. Press “FIND NEXT” button 

and observe the next control which label contains the word “Zoom” will be selected in the “Controls Description” table:

->



The other search options are working in the same fashion.

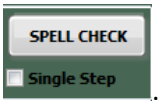
7. Spell Check

The Controls Description Editor is capable of performing the spell check. It is accomplished by executing the spell check features of the Microsoft Word.

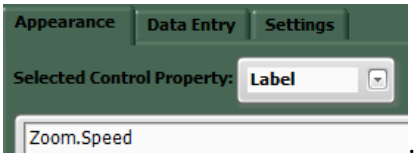
The MS Word is loaded by opening the DummyFile.docx and than passing into it contents of the editing field.

When you are running the Spell Check, please, make shore any MS Word documents are saved and closed. Upon exiting the Controls Description Editor the MS Word will be closed.

Since we were discovering the search functionalities lets run the spell check:

7.1. At the moment make sure the check box Boolean text displays “Single Step”: 

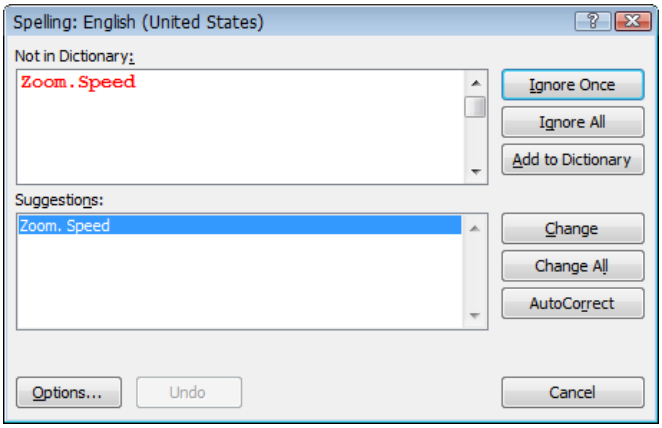
.

7.2. The “Appearance” tab shows: 

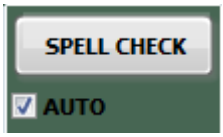
.

7.3. Press the “SPELL CHECK” button to check the “Zoom.Speed” (for instance) text.

7.4. The spelling error window will appear:



If there are no spelling errors the MS Word window will flush shortly and close.

7.5. Check the box below “SPELL CHECK” button to show 

. If this option is enabled, the spell check will be executed from the selected raw in the Controls Description table.

7.6. In this example the “Zoom.Speed” control is being selected. Press the “SPELL CHECK” button.

7.7. The spell check will step through all controls down to the last:

✓ Ctr/String	ControlName.New
✓ Ctr/Boolean	ControlName.Add
✓ Ctr/Digital	Repetition.Period

. The check marks are indicating that that item was spell-checked.

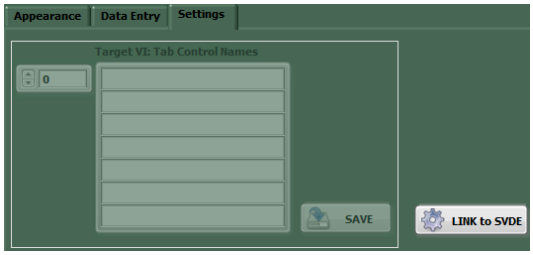
While spell check is running, the “Cancel” button on the spelling window cancels update of the Controls Description table.

The spell check routine is not updating the Target VI – it is updating the Controls Description table entries **ONLY**. You have to send it to the Target VI yourself either by pressing 

button (sends all contents of the table), or item-by-item as described in the previous sections.

8. SubVIs Description Editor Interface

The Controls Description Editor can be interfaced to SubVIs Description Editor. This interface controlled by button on the Settings page:



If **LINK to SVDE** button set to **TRUE** the **Target VI Pa**th will be updated by SubVI Path being selected in SubVIs Description Editor and loaded. This function provides a convenient option to include SubVI controls editing in addition to the SubVI description editing.

9. Technical Support

It is believed the Controls Description Editor is free of bugs. However if you will encounter any errors, please, email to cde@scubaff.com and attach as much information as deems reasonable. We will fix errors within three business days or sooner.