

# SubVIs DESCRIPTION EDITOR

---

## USER MANUAL

### Contents

0. Introduction .....	2
1. Acronyms and Abbreviations .....	2
2. System Requirements .....	2
3. Installation .....	3
4. LabVIEW Environment Requirements.....	4
5. Application Start .....	5
6. Loading Top Level VI .....	6
7. Loading VIs from Folder .....	7
8. EDITING .....	7
8.1. Filtering .....	7
8.1.1. Filter By .....	7
8.1.1.1. Filter By: Folder .....	9
8.1.1.2. Filter By: Name.....	10
8.1.1.3. Filter By: VI With Description, etc.....	12
8.1.2. Filtering Sequence.....	12
8.1.2.1. Add To Filtering Seq(uence) .....	12
8.1.2.2. Apply Filtering Seq(uence) .....	12
8.1.2.3. Clear Filtering Seq(uence).....	12
8.2. Description Editing.....	12
8.2.1. COPY .....	13
8.2.2. Send TO VI .....	13
8.2.3. Bold/Un-Bold.....	13
8.2.4. Spell Check .....	14
8.2.5. Requirement(s) Coverage .....	15
8.2.6. Export To File .....	15
8.2.7. Load From File.....	16

8.2.8. Send To SubVIs.....	16
8.3. System Settings .....	16
8.4. Interface to Controls Description Editor .....	16
8.5. Help and Manual.....	17
9. Technical Support .....	17
10. Known Issues.....	17

## 0. Introduction

SubVIs Description Editor (SVDE) application is intended to provide a LV developer with a tool to retrieve edit and deploy descriptions of SubVIs (dependencies) of a top level VI or VIs located in a selected folder.

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

It seems the easier way to make user familiar with SVDE is to explain its functionalities applying to an example VI (VIs Description Editor\_Demo\_TopVI.vi) which is located in Support folder where SVDE was installed.

SVDE 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

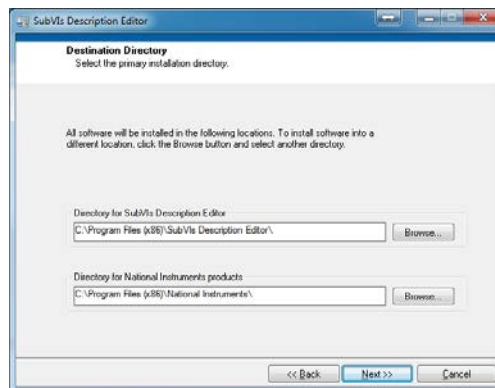
The SVDE is built for Windows XP and up to Win 8.

The SVDE has been developed In LabVIEW 2013 32 Bit version however it will be functional in editing LabVIEW 2013 64 Bit VIs (these VIs will be compiled into LabVIEW 2013 32 Bit).

### 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 SVDE:

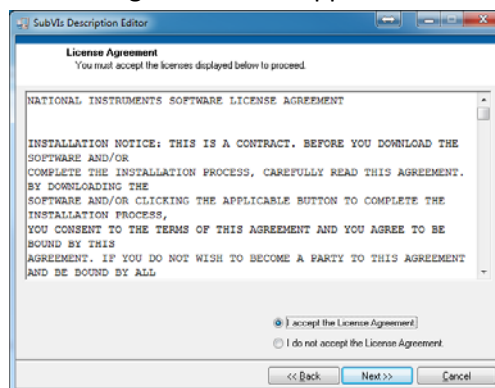
- 3.1. Download and save SubVIs Description Editor Installer.
- 3.2. Go to the VIs Description Installer Ver.1.0.00>Volume>setup.exe.
- 3.3. Launch setup.exe file. In the case of Win 7 there will be warning that changes are about to be made on your computer. Press “Yes”.
- 3.4. After a quick flash NI installation window you will see the Destination Directory pop up window:



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

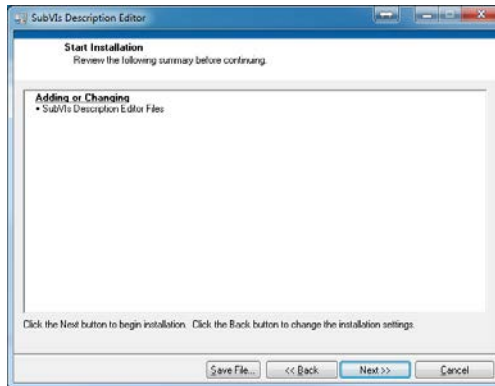
At this moment you may change the destination directories as you may wish but, please, be aware of Win 7 security restrictions.

- 3.5. The standard NI Software License Agreement will appear:



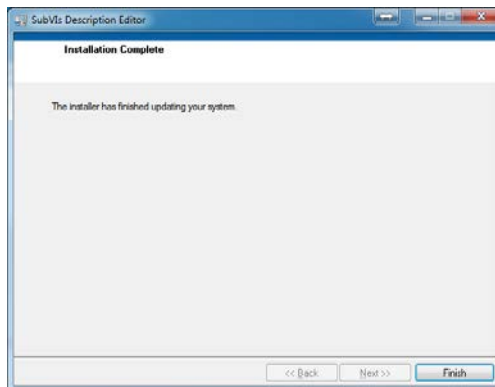
To proceed with installation you have to accept its terms and click “Next”.

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



Click “Next” to continue installation.

3.7. After Installation Complete window will pop up:

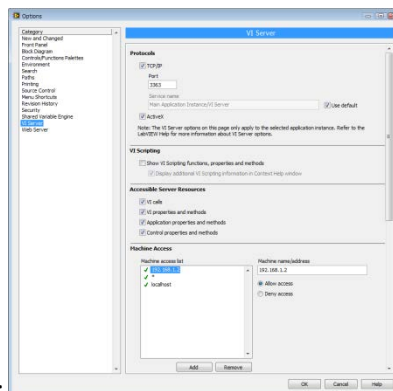


Click “Finish”.

## 4. LabVIEW Environment Requirements

SVDE 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:

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



Please use the picture below as a reference:

## 5. Application Start

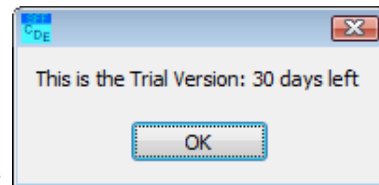
The properly installed SVDE application should create an icon on the Windows Desktop:



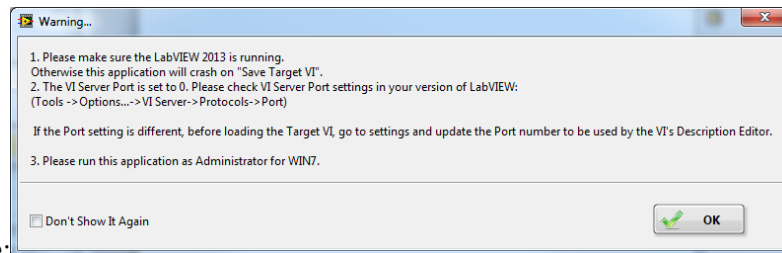
SVDE Shortcut icon.

**NOTE:** If the SubVIs Description Editor is being installed in Program Files, the INI file will be protected by Windows 7 security. Please, make sure you will be running this application with administrative rights.

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

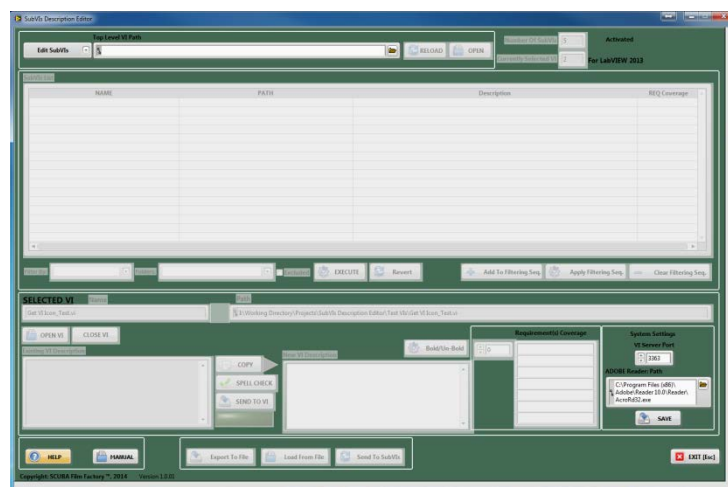


If you have installed a trial version there will be dialog window: notifying you how many days left for this version. The trial period is set to 30 days. After clicking “OK”



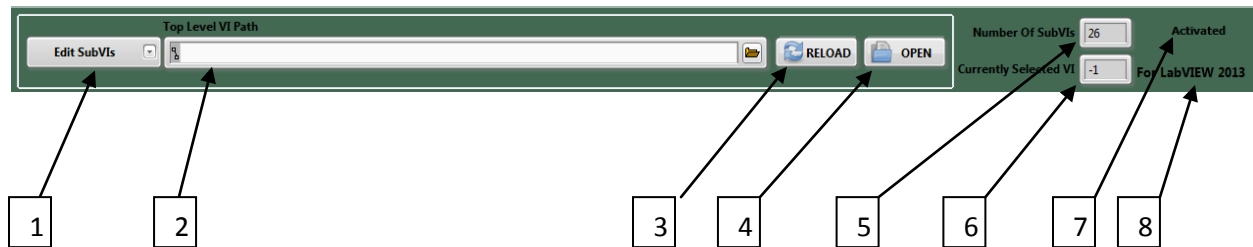
there will be another message: reminding you about required settings to be met.

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



## 6. Loading Top Level VI

The very top section of SVDE is dedicated to the loading the Top Level VI, in this case we will be editing “VIs Description Editor\_Demo\_TopVI.vi” SubVIs descriptions:



Where:

1. Editing mode selector. It can be either Top Level VI (default) or folder.
2. Top Level VI/Folder path.
3. Reload the Top Level VI/Folder button. It can be used to verify changes made and its functionality is identical as Top Level VI/Folder Path selector.
4. Opens the Top Level VI specified by Top Level VI Path selector. In the case of "Folder" editing mode, this control is disabled and grayed out.
5. Number of the SubVIs in the Top Level VI.
6. Index (+1) of the SubVI being selected in the SubVIs List.
7. Application status indicator. If SVDE activated this indicator will be empty.
8. Warning – SVDE is the LabVIEW 2013 application. If any previous LabVIEW code will be accessed by SVDE it will be re-compiled into LabVIEW 2013 which may be extremely undesirable.

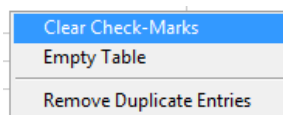
After the path to a valid Top Level VI is being selected its containing SubVIs will be loaded into SubVI List:

SubVIs List			
NAME	PATH	Description	REQ Coverage
VIs Description Editor_Demo_SubVI 1.vi	F:\Working Directory\Projects\VI's Description Editor\Support\VB:		
VIs Description Editor_Demo_SubVI 2.vi	F:\Working Directory\Projects\VI's Description Editor\Support\VB:		
VIs Description Editor_Demo_SubVI 3.vi	F:\Working Directory\Projects\VI's Description Editor\Support\VB:		

There are four columns displayed:

1. SubVIs name.
2. SubVIs absolute path.
3. SubVIs description.
4. Requirement(s) associated with SubVIs.

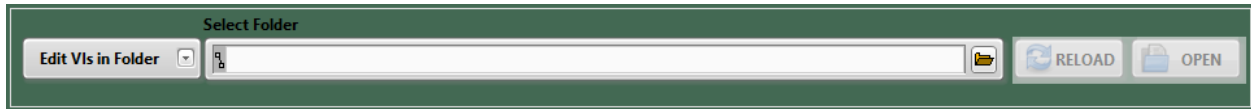
This way the user can assess at glance state of documentation of the application he/she works on.



The SubVIs List has custom run-time menu: which is self explanatory.

## 7. Loading VIs from Folder

The SVDE allows to access VIs from folder. To do this the mode selector should be set to **Edit VIs in Folder**:



It should be noted that VIs in sub-folders will be not accessed.

User can further open sub-folders and add VIs containing there into SubVIs List. However, VIs with duplicate names will be filtered out automatically.

## 8. EDITING

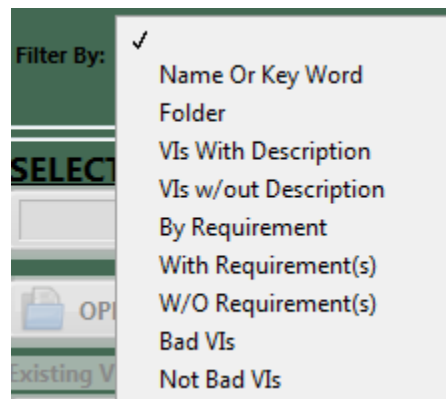
### 8.1. Filtering

SVDE has intuitive filtering facility. When the Top Level VI being loaded all SubVIs, including LabVIEW native ones will be displayed. In many cases developers use various tool sets which should be part of editing or should be edited/documentated on their own. The filtering section of the SVDE shown below:

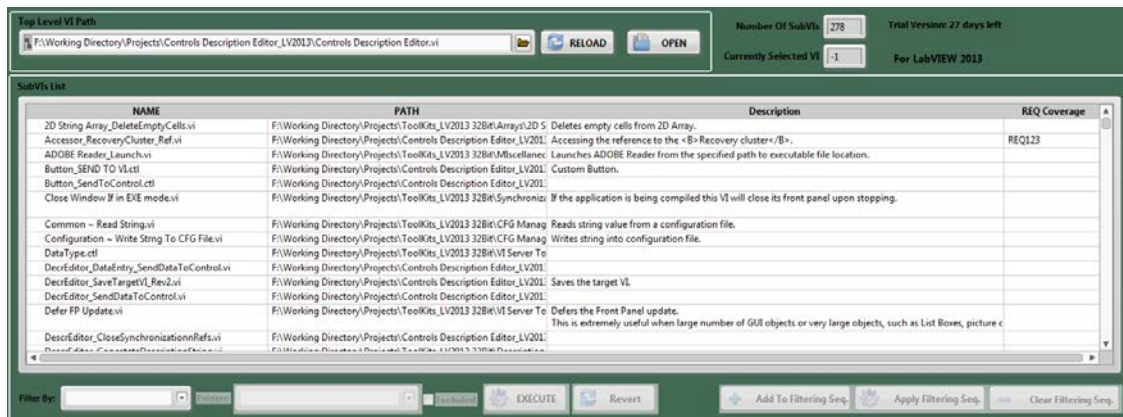


#### 8.1.1. Filter By

The Filter By selector<sup>(1)</sup> allows configuring filtering in respect:



This selection should be made first to continue filtering. As an example, there is SubVIs List of the sizable application – Controls Description Editor:

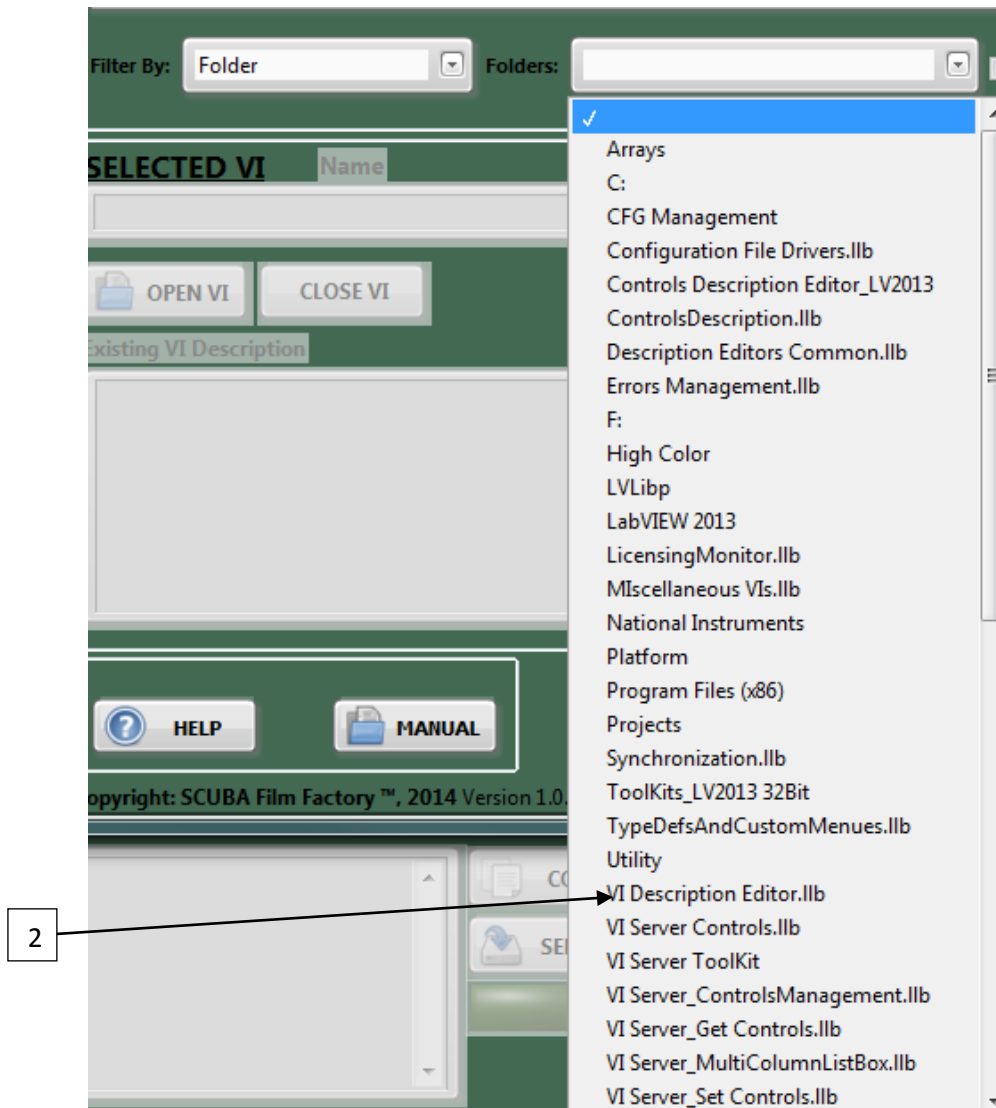


As one may see there are 278 SubVIs located in different folders, some of them have description others don't.



#### 8.1.1.1. Filter By: Folder

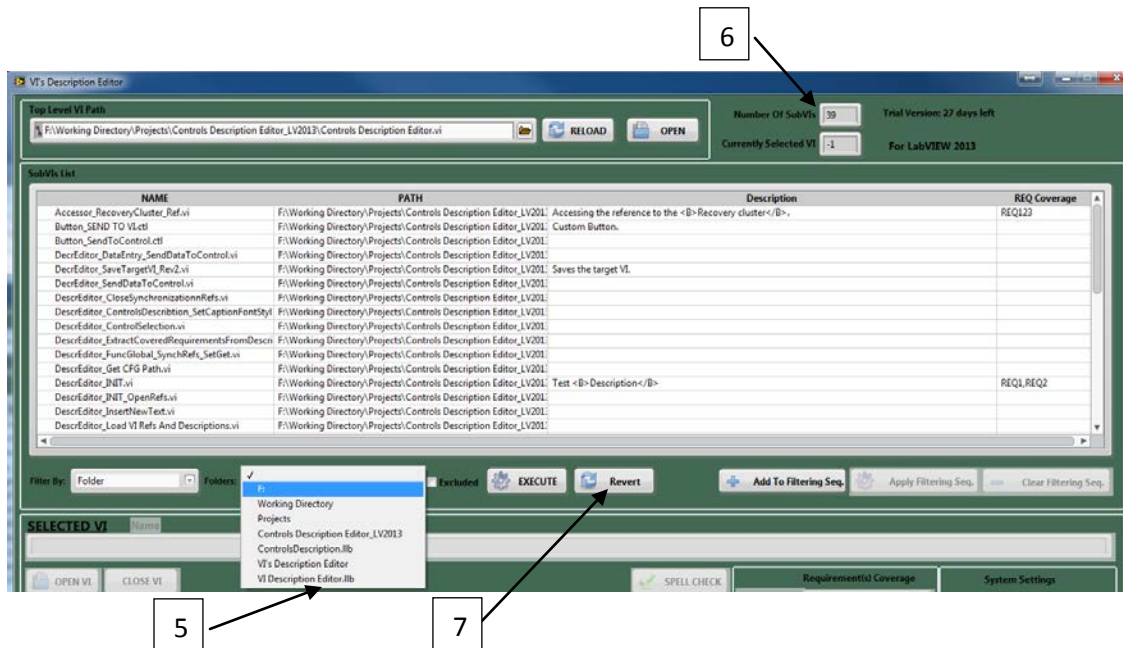
If the Filter By will be set to “Folder” the “Folders” selector will be populated:



For demo sake the folder “ToolKits\_LV2013 32Bit”<sup>(2)</sup> has to be excluded for the editing. To do that, the check-box Excluded/Only<sup>(3)</sup> should be unchecked:



Now, pressing “Execute”<sup>(4)</sup> filtering button will perform filtering:



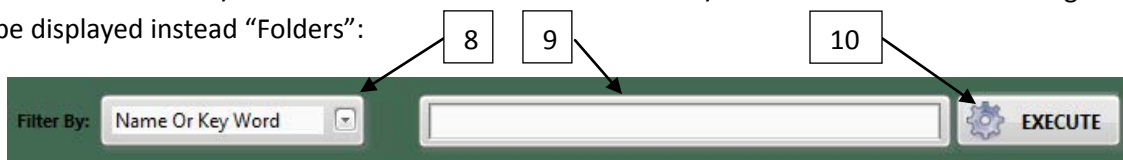
Please note following results of filtering:

1. The folder "ToolKits\_LV2013 32Bit"<sup>(5)</sup> is being removed from the folders list since there are no more SubVIs, which belongs to that folder, listed.
2. The number of SubVIs<sup>(6)</sup> is changed from 278 to 39.
3. The button "Revert"<sup>(7)</sup> becomes enabled, meaning the filtering was executed at least once.

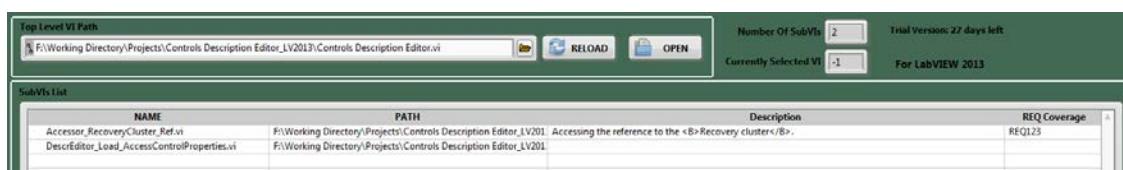
#### 8.1.1.2. Filter By: Name

The intent of Filter By Name is to extract from the SubVIs List those SubVIs which names contain keyword specified.

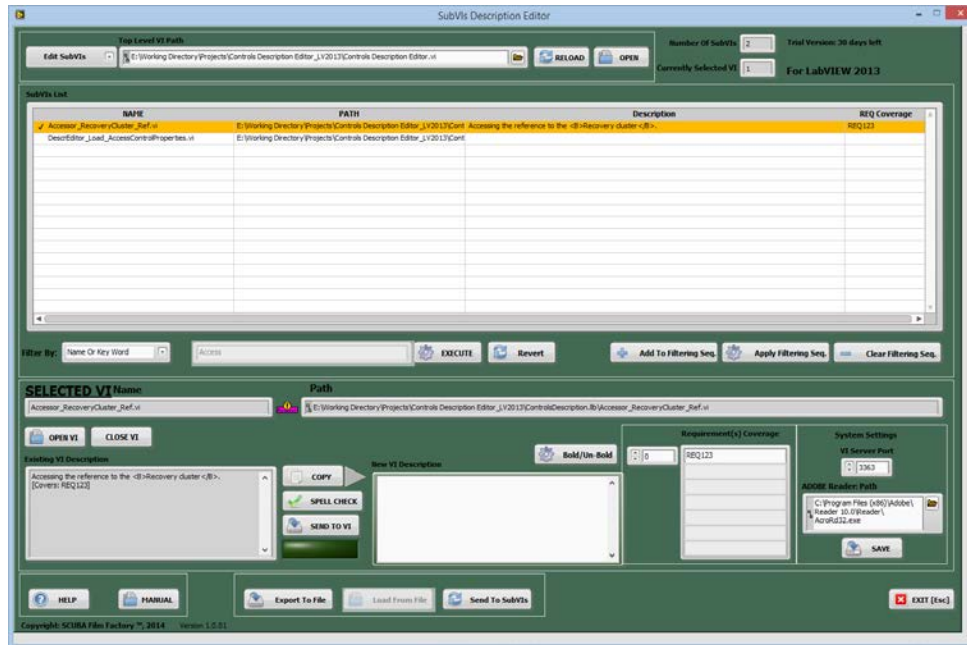
To do that the "Filter By" selector should be set to "Name or Keyword"<sup>(8)</sup>. In this case the string control<sup>(9)</sup> will be displayed instead "Folders":



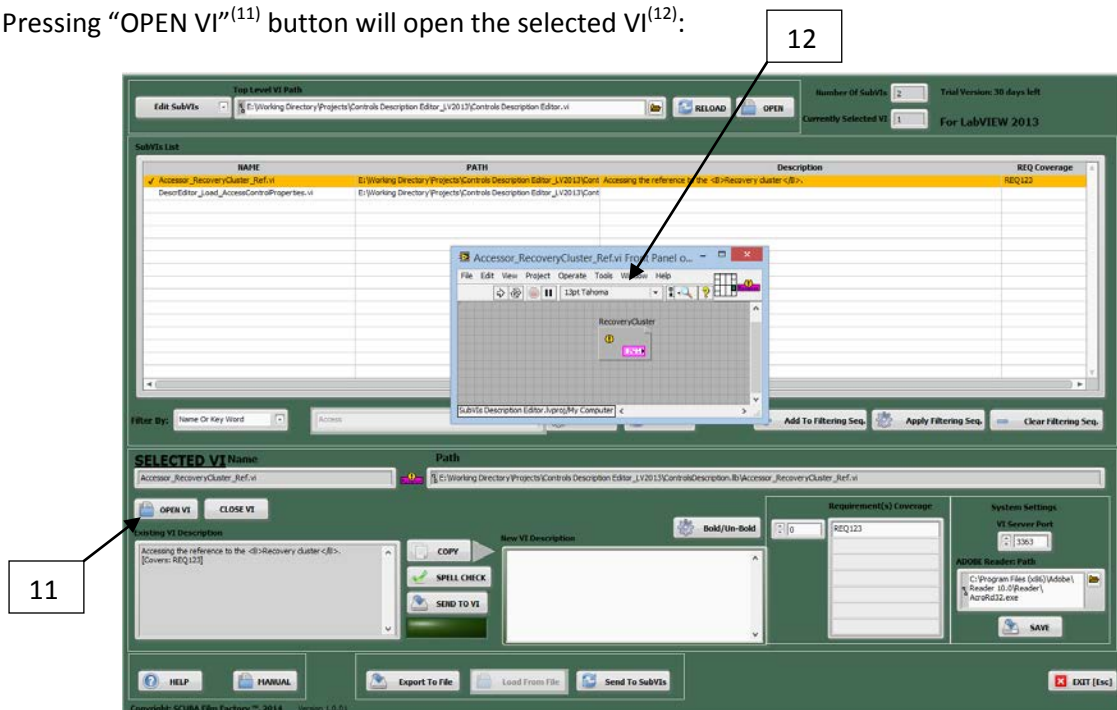
Lets' say we are in the process of developing the Controls Description Editor application and need another VI which with a similar functionality as VIs labeled Accessor... (this VIs return references of notyfiers and/or queues). Since we want to be productive and not to start from a scratch it would be reasonable just to copy one of those VIs as new one. Lets' type Access in the string control<sup>(9)</sup> and press "Execute"<sup>(10)</sup>:



Now the SubVIs List contains just two VIs which name has Access word and “Number Of SubVIs” indicator changed its value to two accordingly. As the next step, double click on the first VI named Accessor\_recoveryCluster\_Ref.vi and observe that selected VI field become enabled and populated:



Pressing “OPEN VI”<sup>(11)</sup> button will open the selected VI<sup>(12)</sup>:



And now we can work with it.

### 8.1.1.3. Filter By: VI With Description, etc.

The “Filter By” selector contains other selections which are self-explanatory.

## 8.1.2. Filtering Sequence

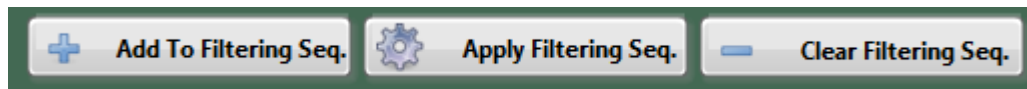
The SVDE stores the filtering settings in the file. When the top Level Vi is loaded this setting are extracted. However, please keep in mind, this filtering sequence may be not compatible with different Top Level VI since it may have different dependencies.

If at least one filtering setting was added to the sequence the file is being created and “Apply Filtering Seq.” control will be enabled. Intend of this functionality is that large applications may cause multiple iterations of documentation editing. It would be convenient to streamline SubVIs sorting to save time.

### 8.1.2.1. Add To Filtering Seq(ue)nce

Adding filtering settings to the sequence should be done before filtering was executed. The reason is that folders list may be changed if is excluded or only displayed (check-box “Excluded/Only”).

A soon first “Add To Filtering Seq(ue)nce” routine is being executed the “Apply Filtering Seq(ue)nce” and “Clear Filtering Seq(ue)nce” buttons become enabled.



### 8.1.2.2. Apply Filtering Seq(ue)nce

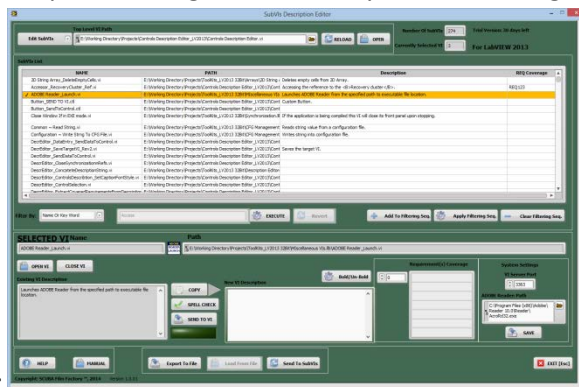
This control initiates filtering as the sequence of specified and stored filtering executions. Essentially it just replicates filtering steps executed by user.

### 8.1.2.3. Clear Filtering Seq(ue)nce.

The filtering sequence is being stored in a queue and saved into the file. If Clear Filtering Sequence routine invoked, the corresponding queue will be flashed and the file section, containing filtering data, deleted.

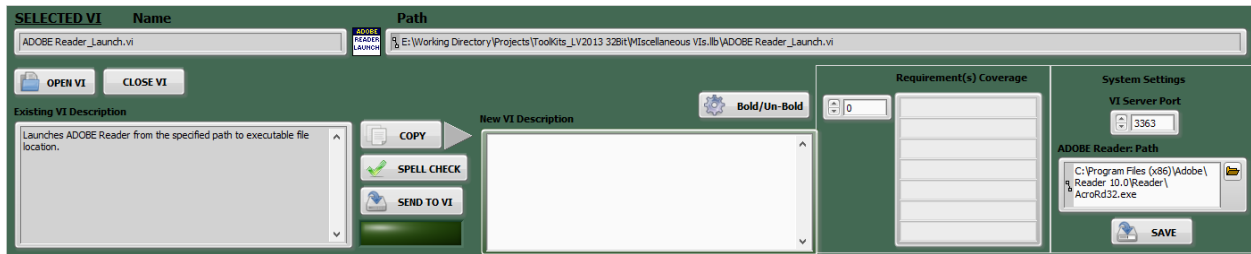
## 8.2. Description Editing

The description editing is activated by double clicking on a row of the SubVIs List, containing a target



SubVI:

In this case the corresponding controls and indicators are enabled and populated:



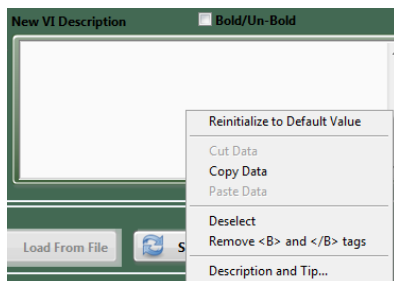
Most of these GUI objects are self-explanatory (NAME, Path, Existing VI Description, OPEN VI, CLOSE VI) and there is no reason to describe their functionality.

**NOTE:** The checkmark in the SubVIs List doesn't mean selection. Its purpose is to mark the SubVI as being accessed. The selection is indicated by highlighted row and done by double click. If another SubVI would be selected the check mark will stay for purpose of indication that associated SubVI was accessed.

### 8.2.1. COPY

The "COPY" function transfers contents of the "Existing VI Description" indicator to the "New VI Description" control. In this control text and style (Bold/Plain) can be modified and checked for spelling.

The "New VI Description" has custom run-time menu:



. The custom items are "Deselect", which clears text selection and "Remove <B> and </B> tags", which deletes from the description text all bold tags.

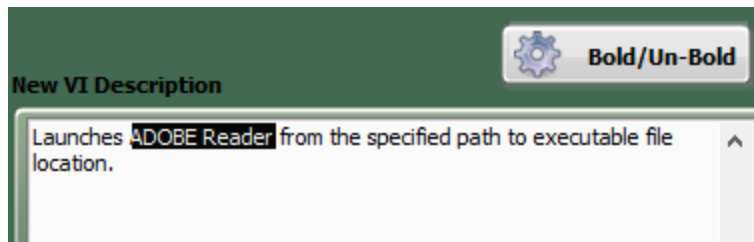
### 8.2.2. Send TO VI

Sends content of the "New VI Description" to the target SubVI, saves it and verifies correctness. If the target VI description was successfully updated, the green indicator below will be set to True:



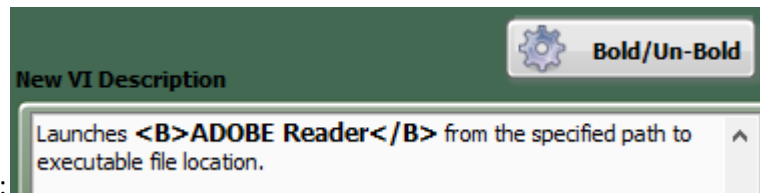
### 8.2.3. Bold/Un-Bold

This check- box has momentary mechanical action and used to insert or delete HTML tags (<B> and </B>) into description text which will be "Bolden". To do it, please, select the segment of the text to be modified in the "New VI description" control:



and press the Bold/Un-Bold button to

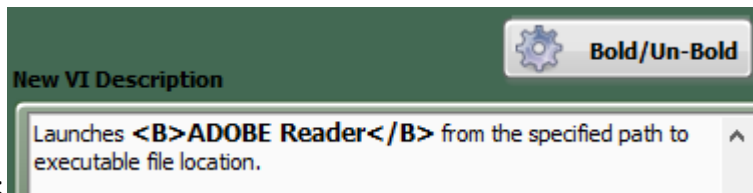
insert HTML tags:



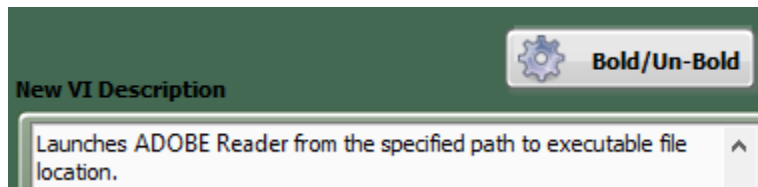
If there were no other selections made, the application holds an existing string section selection, if the Bold/Un-Bold button will be presses again it will remove HTML tags.

If there need to remove HTML tags explicitly select the string segment, including <B> and </B> tags

again:

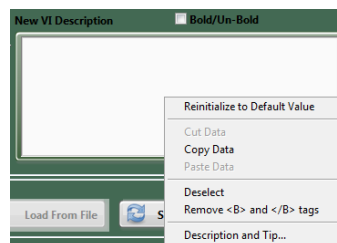


and click on the check-box again:



- the text segment becomes plain and

HTML tags are being removed. Another method to remove bold tags is to use run-time menu:

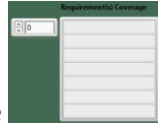


#### 8.2.4. Spell Check

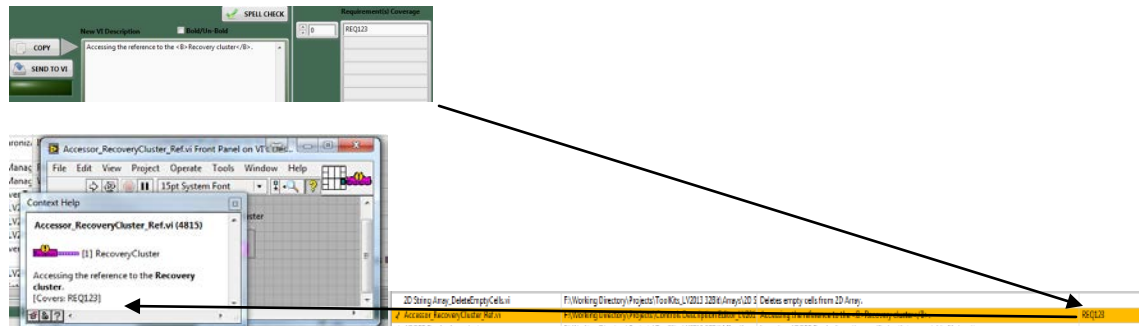
Pressing this button will initialize the spell-check within MS Word (it should be present in the development machine). If there no errors detected the MS Word screen will flash for a moment and close. If there is misspelling the MS Word spell-check window will stay open.

**NOTE:** Spell Check function opens the DummyFile.docx. When the SubVIs Description Editor getting closed, there will be an attempt to close the DummyFile.docx by closing MS Word. If there are other MS Word documents are open they will be closed too.

## 8.2.5. Requirement(s) Coverage

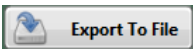


Requirements Coverage control allows to enter a list of requirements associated with the Targeted SubVI. When description will be sent to the Target SubVI the Requirement(s) Coverage content will be properly formatted and added to description:

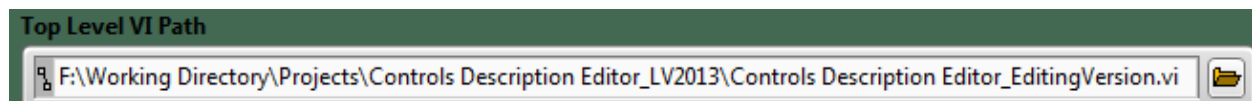


When SubVIs loaded, the description is being decomposed and requirements are separated from the description text. It is done to make editing more convenient and assure the correct formatting.

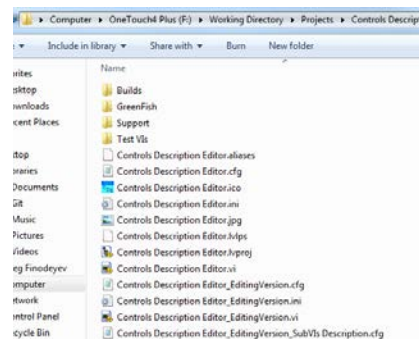
## 8.2.6. Export To File

Export To File  button writes descriptions of all SubVIs into configuration file.

In this example the Top Level VI path is:



and the SubVIs descriptions will be located in the same directory:



This file contains contents of the SubVIs List is the format below:



---

[Accessor\_RecoveryCluster\_Ref.vi]

Path = "F:\\Working Directory\\Projects\\Controls Description  
Editor\_LV2013\\ControlsDescription.llb\\Accessor\_RecoveryCluster\_Ref.vi"


Description = "Accessing the reference to the <B>Recovery cluster</B>."

REQ Coverage = "REQ123"


---

At this point the description can be edited and/or translated outside LabVIEW development environment.

#### 8.2.7. Load From File

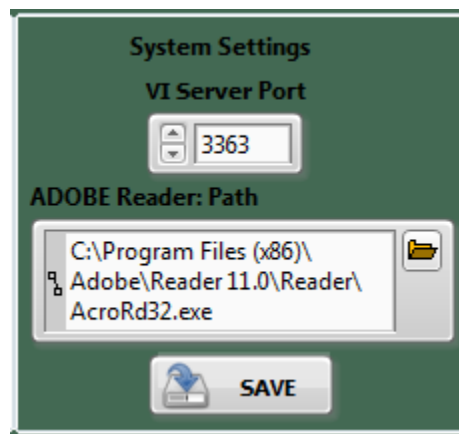
The button  transfers SubVIs description and requirement coverage from the file into SubVIs List. Now description(s) can be edited and sent to the corresponding SubVI(s).

#### 8.2.8. Send To SubVIs

The button  sends the whole contents of the SubVIs List into individual ones and saves them.


### 8.3. System Settings

There are just two settings should be adjusted to make SVDE functional:

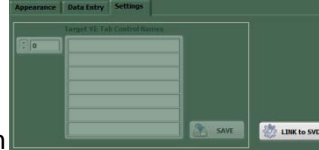


The ADOBE reader path varies from machine to machine and VI Server Port to be used should be identical to one used by LabVIEW development platform.

### 8.4. Interface to Controls Description Editor

The SubVIs Description Editor is capable to interface with Controls Description Editor. To implement this, the Controls Description Editor application should be open and the SVDE button  button to





be set to TRUE. In an addition the corresponding button should be, also, set to TRUE. In this case double-click on a raw of the SubVIs List will update corresponding controls of the Controls Description Editor and load its controls.

## 8.5. Help and Manual



These buttons functionality is self-explanatory.

## 9. Technical Support

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

## 10. Known Issues

There was observer random application crash. At the moment, the reason of this unknown. If it happened, we apologize for inconvenience. We are working to resolve it and as soon it will be done we will notify you about it.