

Icon Editor

White Paper

Mattias Ericsson



Abstract

Managing icons in LabVIEW™ is a manual and time-consuming task. The build-in icon editor in LabVIEW™ is very simple and only edits one icon at a time. It is not possible to define a common icon style for a group of VI.

The new *Icon Editor* makes icon generation and maintenance easy and fast. Keeping a good icon style within the project is no longer a problem. The Icon Editor is a tool for generating and maintaining icons. An icon style consisting of a common header and colors for icons is defined which will be applied to all icons. It handles everything from single VIs to large structures of VIs.

Basically there are three ways of working the Icon Editor:

- Generate the entire icon (header, colors and body text)
- Only applying the header and colors
- Only applying the header

The Icon Editor consists of two main tool:

- Icon Manager – advanced tool for generating and maintaining an icon style for a group of VIs.
- Create VI Icon – simple tool for creating an icon for a VI in a fast and simple way.

System requirements are LabVIEW™ 6.0.2 or later (Professional Development System or Full Development System) and all platforms are supported.

Contents

1	Purpose.....	4
2	Introduction.....	4
3	Icon style	5
4	Icon Manager Tool.....	5
4.1	Selecting VIs.....	6
4.1.1	Public and private VIs.....	6
4.2	Edit header and colors.....	7
4.2.1	Header Editor	8
4.3	Generate icons.....	9
4.3.1	Entire icon.....	10
4.3.2	Only header and colors	10
4.3.3	Only header.....	10
4.4	Checking and modifying individual icons	10
4.4.1	VI Icon Editor	11
4.5	Applying and updating VIs	11
5	Create VI Icon Tool	12
6	System requirements	13
7	Example	13

1 Purpose

The purpose of this document is to present the *Icon Editor* and how to use it. It is assumed that the reader of this document has basic LabVIEW™ knowledge.

2 Introduction

Managing icons in LabVIEW™ development is a manual and time consuming task. In large projects there might be thousands of VIs and of course thousands of icons to handle.

As LabVIEW™ is a graphical programming language, the use of good icon style is important to make the code easy to read and understand. The build-in icon editor in LabVIEW™ is very simple and only edits one icon at a time. It is not possible to define a common icon style for a group of VI.

In many LabVIEW™ project, the header of the icon defines to which category the VI, belongs to. A category might be a module, a class, function library, component etc., anything that make a group of VI have something in common. By using headers it is really easy to understand where a VI belongs and the code is easy to read.

The use of colors might also be a good idea, in some project the header background color specifies which layer of the architecture the VI belongs to. The background color of the lower part, body background color, might symbol something else. In GOOP for example, it is quite common to make all private methods using a white body background and all public methods uses another color.

If there is many VIs in a project, usually there is no time to make a good symbolic icon for every VI. Normally the name of the VI is simple typed in the icon using a small font like “Small Fonts” size 8.

One of the most time-consuming task existing is actually changing header and color for a group of VIs. For instance, if a member of a developer team has made a mistake and not following the icon style guide for the project and needs to correct the mistake, the only option remaining is to edit all VIs manually, one by one. Normally this is not considered important enough and usually skipped.

There are as many icon styles as there are LabVIEW™ users and there certainly is a need for a tool that might help making icon generation and maintenance easier and faster. The new *Icon Editor* makes icon generation and maintenance easy and fast. Keeping a good icon style within the project is no longer a problem. As long you use headers on your icons, you probably find the *Icon Editor* a very useful tool.

The Icon Editor consists of two main applications, the *Icon Manager* and the *Create VI Icon* applications. The *Icon Manager* is used to manage and maintain a large group of VI icons, for instance when changing colors of many icons. The *Create VI Icon* is used frequently for each VI and is a simple tool to create the icon for a VI.

3 Icon style

To be able to generate icons, there must be some kind of icon style for the Icon Editor to work with. Therefore use of a *header* is the key of the Icon Editor tool. There are a number of different colors that might be set common for all icons; header background, body background, foreground and frame color. Figure 1 shows an example of a typical icon with a header.



Figure 1. An example of an icon with a header. This icon has a green header background, red body background, black foreground (text color) and black frame color. All these colors are common for all VIs.

All VIs must have a frame in the same color and 1 pixel wide. Normally the frame color is black. The header might be a text, but it is also possible to use some kind of more symbolic user-defined header if wanted. *The header is defined using a horizontal line in the same color as the frame.* Notice that the icon must be 32 x 32 pixels and the B&W and 256 colors icons must be defined for the VIs.

Basically there are three ways of working the Icon Editor:

- Generate the entire icon (header, colors and body text)
- Only applying the header and colors
- Only applying the header

If generating the entire icon, a textual description will be suggested as body text, base on the VI name. If the VIs has a common prefix in the file name this will be excluded from the suggested text.

4 Icon Manager tool

The Icon Manager is a tool for generating and maintaining icons. It handles everything from single VIs¹ to large structures of VIs. To fully benefit from all the advantages, it works best for a group of VIs that has something in common².

Working with the Icon Manager involves a number of steps that has to be performed:

¹ Type Def controls (.ctl) is handled just as well. The Icon Manager also handles (.ctt) and (.vit) templates. In this paper, where ever VIs are mentioned, this will also be applicable for ctl, ctt, and vit files.

² It is even better if all VIs in the group have a common prefix in the filename.

1. Select VIs to edit.
2. Create, find or edit the common header and colors.
3. Generate³ icons for the VIs. It is also possible to skip this step.
4. Inspect the icons and edit icons that are not good enough.
5. Apply and update the VIs with the new icons.

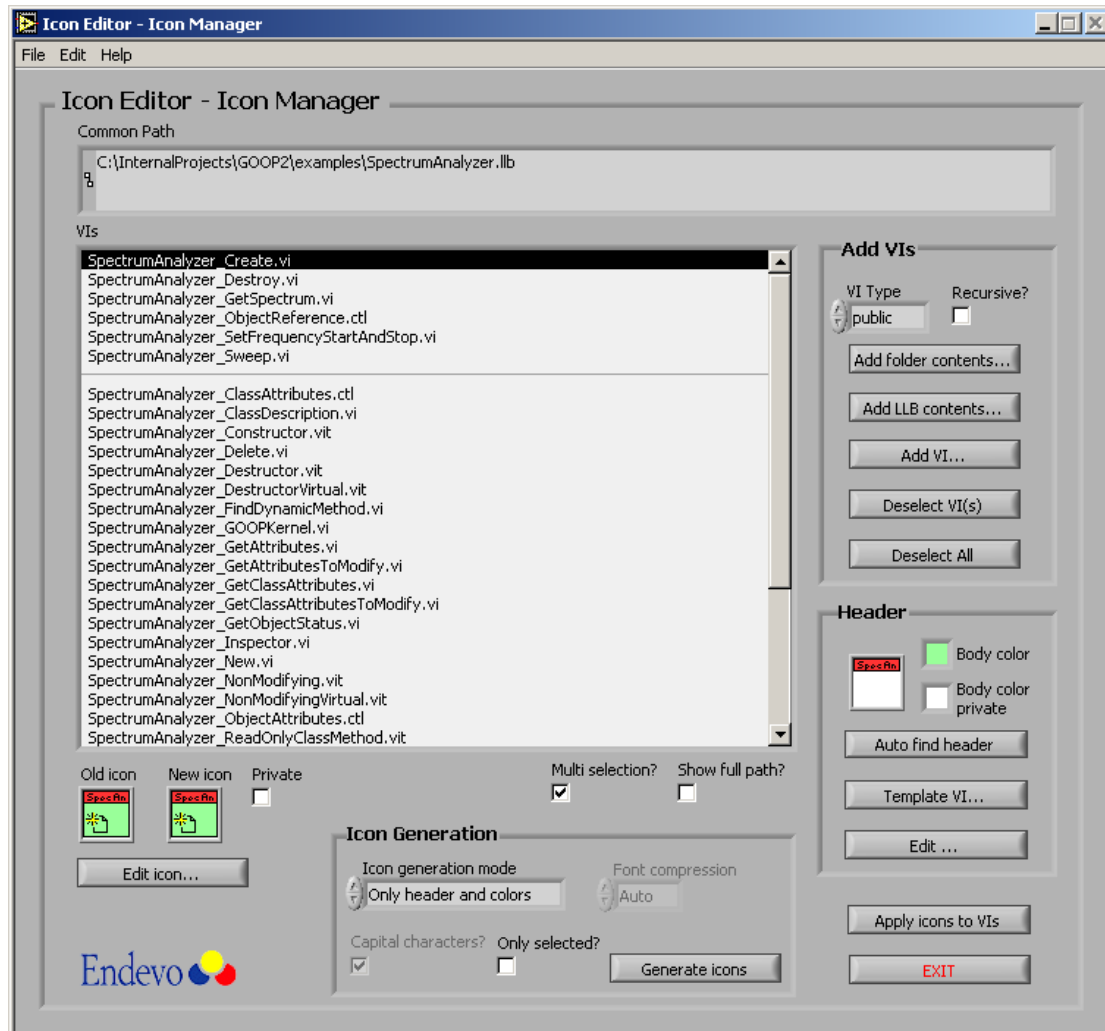


Figure 2. The front panel of the Icon Manager.

In Figure 2, the front panel of the Icon Manager is shown. The Icon Manager is started from the LabVIEW™ Tools menu -> Icon Editor – Icon Manager.

4.1 Selecting VIs

First of all, the VIs to edit must be selected and added to the VI list at the front panel. Single VI, LLB contents or folder contents could be added, folders even recursively, which means that all sub folders and LLB will be searched and added.

4.1.1 Public and private VIs

The Icon Manager uses the concept of *public and private VIs*. The difference is which background color to use. It is possible to use two background colors in the Icon

³ This will NOT apply and update the VI with the icon.

Manager. When adding a single VI or a folder contents, public or private has to be selected with the VI Type selector. Default is public. In a LLB all top-level VIs is treated as public and the rest as private.

In the Icon Manager, the public and private VIs are separated by a gray separator line in the VI list. It is possible to change if a VI is public or private in the list by just changing⁴ the checkbox indicating if the VI is public or private.

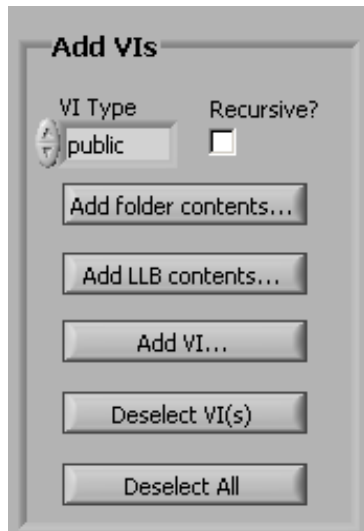


Figure 3. VIs may be added as single VIs, LLB contents or folder contents. It is also possible to add folder contents recursively. Notice that it is also possible to deselect VIs.

4.2 Edit header and colors

When all VIs to be edited are added, the next step is to define the header and colors. This could be performed in three different ways.

1. If there already exists a header along the selected VIs, this could be found by using the *Auto find header*. This will search through the VIs and try to find the header and colors. The header and background colors are shown at the front panel.
2. A header could also be defined pointing⁵ out a single VI that has the wanted header. It is quite common that one VI is designed as a template. Using the *Template VI* performs this. However, the body background colors must still be defined. One single template VI could of course not have both a public and a private background color!

⁴ This will not change the location of the file. If the file is in a LLB, it will change if it's top-level or not as it would in the GOOP Wizard.

⁵ This VI might or might not be one of the VIs in the VI list.

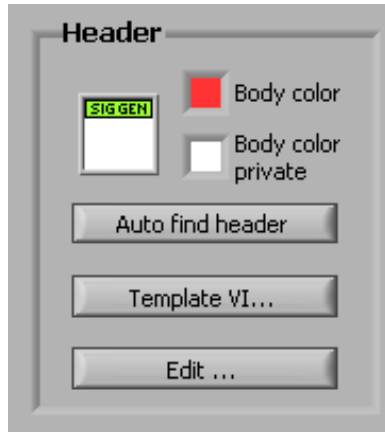


Figure 4. The header might be selected using a template VI or auto find the header. It is also possible to create a header from scratch.

3. A header could also be created from scratch. Selecting *Edit* performs this and starts the *Header Editor*, explained in chapter 4.2.1.

The header and color could be edited at any time. If a more symbolic and artistically header should be used, it is recommended to design the header in LabVIEW™ standard icon editor, then select this VI as the template VI. This way it is easy to use “good-looking” headers.

4.2.1 Header Editor

The *Header Editor* edits the header and selects colors common for all VIs. If a new text should be applied, the icon will update immediately when typing. If only colors should be edited, do not make any change in the *Header Text control*. The default text in the control is the common prefix for all VIs in the list, if no prefix is found the control will be empty.

It is possible to force using capital characters. The font compression (font width) could be set to *Auto*, *On* or *Off*. Font compression will try to force as many character as possible into one line. Figure 5 shows the front panel of the Header Editor. There is also a very simple mouse tool available where single point could be added.



Figure 5. The front panel of Header Editor where header text and colors might be changed.

Notice if the header background color is accidentally set the same as foreground color, the entire header will be “blank”. Press *Cancel* and retry if necessary.

4.3 Generate icons

When the header and colors are defined, it is time to generate icons. The generated icons are only a proposal and are not applied to the VIs. Both the original icon and the new generated are shown for the selected VI.

Icons could be generated for all VIs in the list or just the selected ones. This is controlled by the *Only selected* checkbox. As explained in chapter 3, there are three ways of generating icons; Entire icon, only header and colors and only header. The *Icon generation mode* selects this. Pressing the *Generate icons* button will generate the icons. Notice, it is always possible to regenerate the icons again if the result is not satisfying, maybe after reediting the header.

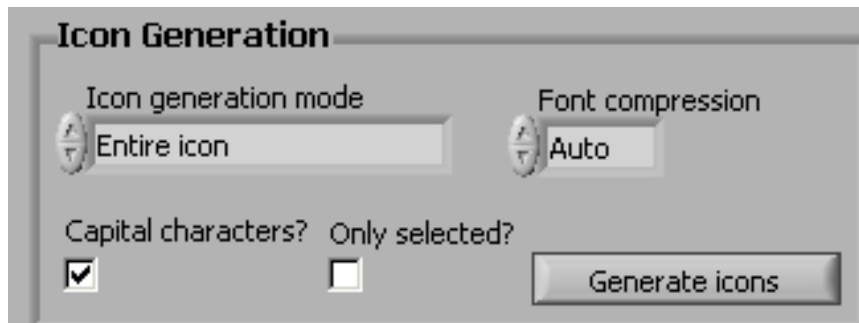


Figure 6. There are three icon generation modes; entire icon, only header and colors, or just the header.

4.3.1 Entire icon

This mode will generate the entire icon with header, colors and a text in the body of the icons. The text is based on the VI name. If a common prefix is found along the VIs, this will be removed. For example, there are three VIs:

Test_Main.vi
Test_Hello.vi
Test_Sweep.vi

The common prefix is Test_ and this will be excluded and the body texts will be Main, Hello and Sweep.

If just one of the VIs has a different prefix, no common prefix could be found! However, by checking the *Only selected*, only the selected VIs will be used to find the prefix. This way the icons may be generate in steps by selecting VIs with same prefix and generate the icons for these, until all icons are updated.

The Font compression and Capital characters controls if the body texts have as many characters as possible on one row and if only capital character should be generated.

4.3.2 Only header and colors

This mode will only paste the header and colors for the icons. The body color will be changed as well, but no body text will be applied.

4.3.3 Only header

This mode will only paste the header. Frame and body colors will be unaffected.

4.4 Checking and modifying individual icons

When the icons have been generated, then icons need to be inspected. Most of the icons will be fine with the suggested icon, but some icons, especially if the entire icon has been generated, will need to be fixed. Figure 7 shows an example of an icon that needs to be manually edited. The most common problem is VIs with long names and words that will not fit within the body of the icon.



Figure 7. An example of an icon generation that was not perfect. This is common for VIs with long names (and words) where the text will not fit within the body.

Select the VIs one by one and inspect the icons. If an icon needs to edited press the *VI icon editor* button to open the *VI Icon Editor*.

4.4.1 VI Icon Editor

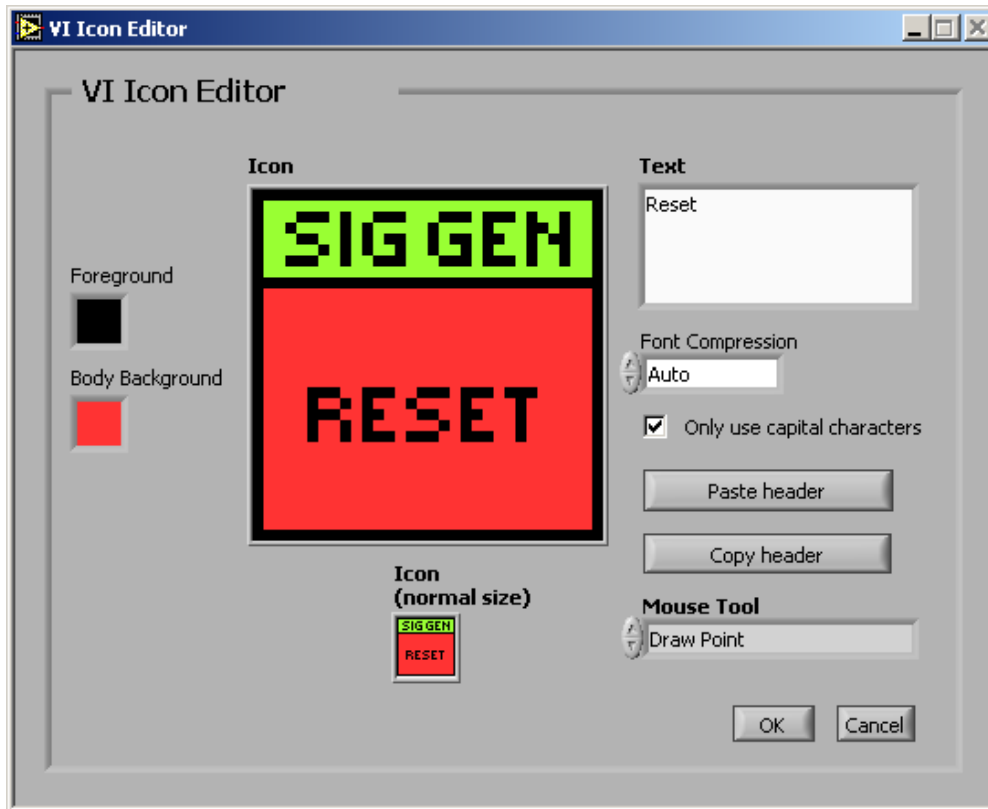


Figure 8. The VI Icon Editor. Individual icons might be updated and corrected if the auto generation was not good enough.

The *VI Icon Editor* almost works as the *Header Editor* described in chapter 4.2.1, the difference is that these changes will only affect the selected single VI and not all VIs. As soon the *Text* control is updated the change will be visible in the icon. Notice that it is possible to change body background for the VI.

There is a button called *Paste header* and *Copy header*. The *Paste header* will paste the common header to the icon; this might be good if no icon has been generated for the VI (it is possible to skip that step). The *Copy header* will copy⁶ the header from this icon and use as the new common header for all VIs. Notice, to apply this header to all other icons, icons has to be generated again.

4.5 Applying and updating VIs

When all icons are correct it is time to actually apply the icons to the VIs. Press *Apply icons to VIs* and the updating will take place. This might take a while.

⁶ Usually it better to select *Template VI* described in chapter 4.2.

5 Create VI Icon tool

The *Create VI Icon* tool is a much simpler tool than the Icon Manager. It has a different purpose and usage, while the *Icon Manager* is used to generate and maintain a group of VIs and is used occasionally, the *Create VI Icon* tool is used much more frequently for each VI. Notice that type def controls and globals are not supported by this tool.

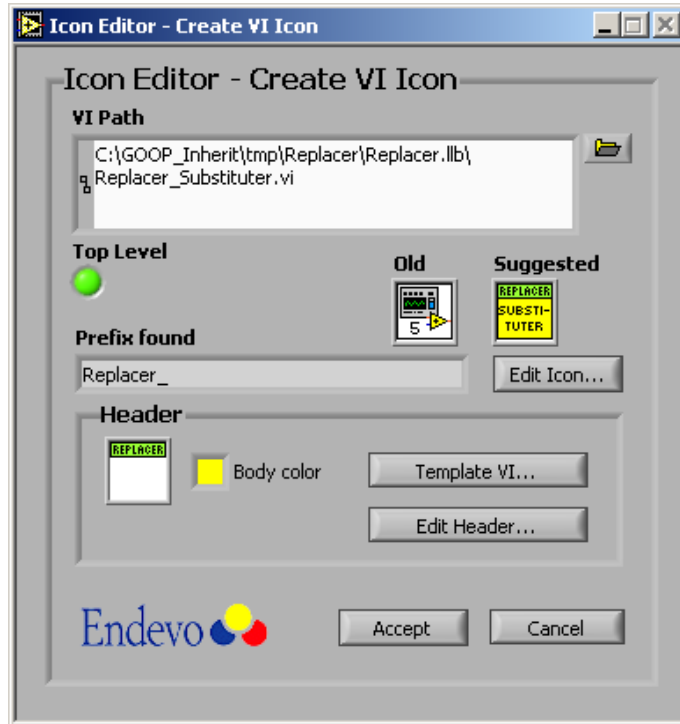


Figure 9. The front panel of the Create VI Icon tool.

Figure 9 shows the front panel of the Create VI Icon tool. It is started from the LabVIEW™ Tools menu -> Icon Editor – Create VI Icon. **Notice that VI to be updated is the launching VI** (the VI that was active and whose “Tool” menu was used). When the tool is launched, a search is performed in the folder or llb where the launching VI is located⁷ to find a common file name prefix and icon style. The tool then automatically shows a suggested icon next to the present one. If the icon is acceptable, just press “Accept” and the new icon is applied. Normally this is all that has to be performed! Just launch the tool and press “Accept”. Could it be simpler?

However, if the VI is alone in the folder or LLB, there is of course no way of getting the prefix, header and colour. No icon will be suggested and the header must be defined before an icon may be suggested. The header is defined either by selecting a template VI or create a header from scratch (select “Edit Header”). As soon as the header is defined, a new icon will be suggested.

If the suggested icon is not acceptable, there are two ways of dealing with this:

⁷ If the VI is not yet saved, select a filename and the VI will be automatically saved.

1. The header and colors found are not the wanted. To fix this, either select a template VI with the correct header and color or simply just edit⁸ the header.
2. The suggested icon is not good enough, maybe there were problems identifying a prefix or the suggested body text (that is based on the VI name) does not fit into the body. Simply just press “Edit icon” to edit⁹ the suggested icon.

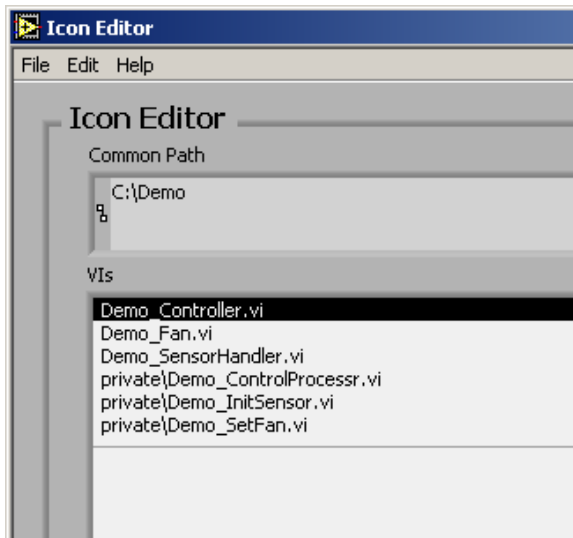
6 System requirements

- LabVIEW™ 6.0.2 or later (Professional Development System or Full Development System)

7 Example

This simple example shows how to generate the entire icon for a group of VIs with the *Icon Manager*. The VIs has a common prefix in the filename. The VIs are in a folder and all sub-VIs in a subfolder called private.

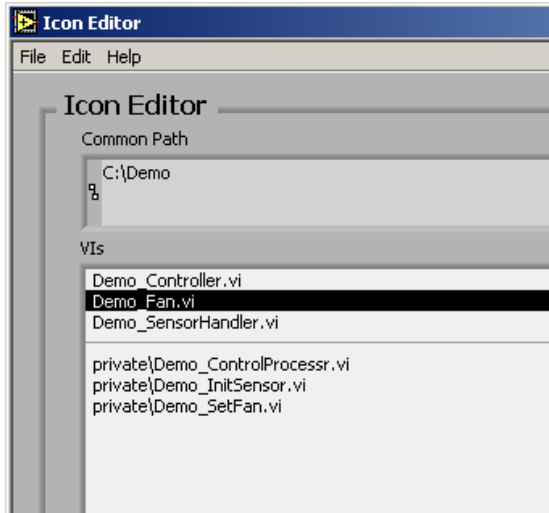
1. Start the Icon Editor from the LabVIEW™ Tools menu -> Icon Editor.
2. Check the *Recursive?* checkbox and select *Add folder contents*.



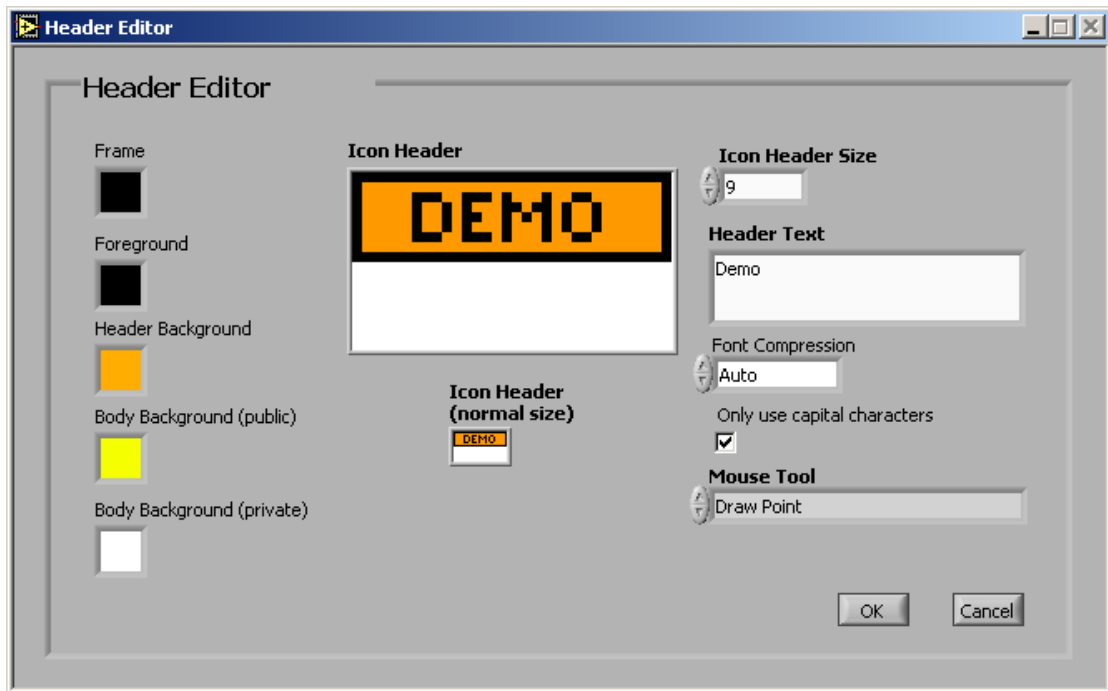
3. Now, the sub-VIs must be set as private. Select the last three VIs and change the *Private* checkbox.

⁸ See chapter 4.2.1.

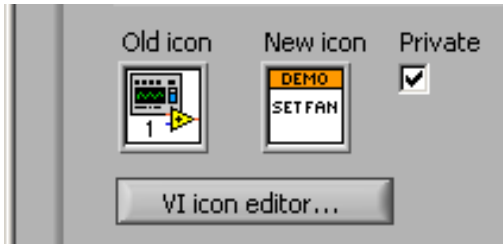
⁹ See chapter 4.4.1.



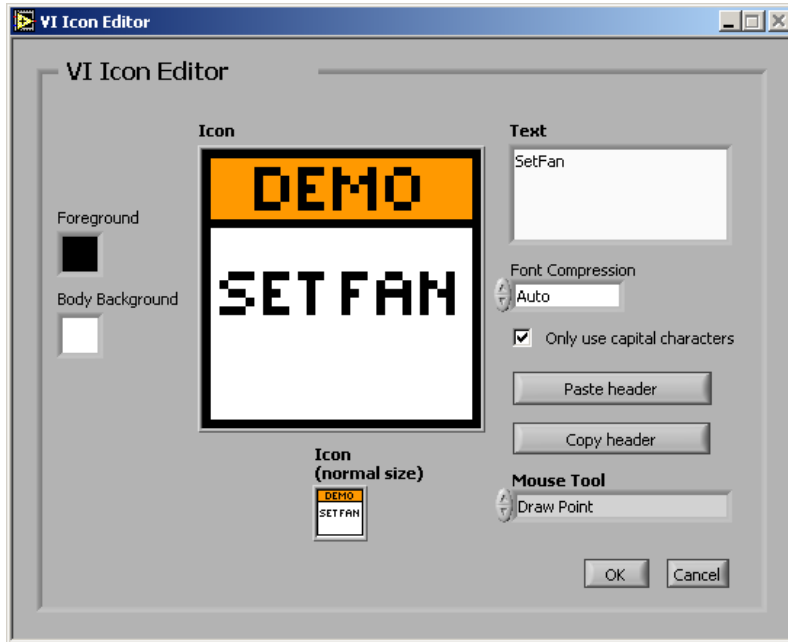
4. No previous header or colors exists for these VIs. Press *Edit* in the header section.



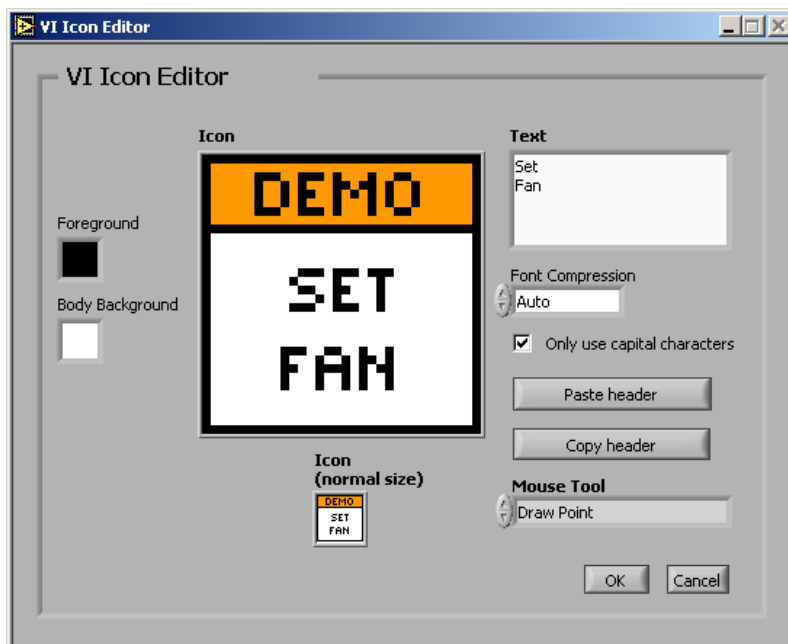
5. Notice that “Demo” automatically is shown as Header Text and also in the header (no header previously existed, else the old header would be unaffected). Change the colors and text if needed. Press ok.
6. Now the header and colors are defined. It is time to generate a proposal for the icons. Select the *Entire Icon* generation mode and press *Generate icons*.



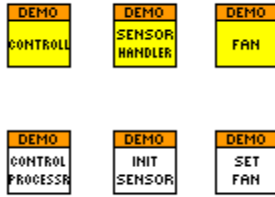
7. Select the VIs in the list one by one and inspect the icons. In this example, one icon is not perfect. Select this VI and press *Edit icon*.



8. The *VI Icon Editor* pops up. Now lets put the words Set and Fan on two different lines. Press Ok when done.



9. The icon generation is completed. Press *Apply icons to VIs*.



10. Finished!