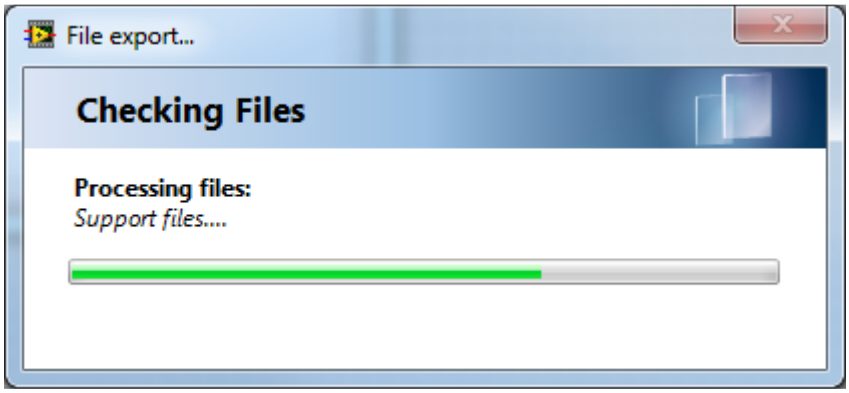
| Doc no | Issued by | Rev |
|---|---|---|
| AC0007-002 | Jonas Mellroth | A |
| Project/Product name | Approved by | Date |
| WireFlow ProgressBar | Jonas Mellroth | 2012-04-27 |

# ProgressBar

**Abstract**

The WireFlow progressbar module is an easy way to add progress bars to an application.
It is easy to customize the look of the displayed progress window, since all update handling is performed in a common progress bar engine. It also supports common string formatting for the message text (bold, underline, italic) to put focus on the specific details in the progress.
There is an example showing most of the features available in the ProgressBar module, like;

- Delayed opening of the Front panel
- Support of cancel request from the ProgressBar
- …



**Contents**

1 License ... 3
2 OverView ... 4
3 API Methods ... 5
3.1 ProgressBar_WireFlow.lvclass:CheckCancelNotification.vi ... 5
3.2 ProgressBar_WireFlow.lvclass:GenerateCancelNotification.vi ... 5
3.3 ProgressBar_WireFlow.lvclass:GetEventRefs.vi ... 6
3.4 ProgressBar_WireFlow.lvclass:PB_Clear.vi ... 7
3.5 ProgressBar_WireFlow.lvclass:PB_Init.vi ... 7
3.6 ProgressBar_WireFlow.lvclass:ProgressBarEngine.vi ... 8
3.7 ProgressBar_WireFlow.lvclass:SetMessageString.vi ... 9
3.8 ProgressBar_WireFlow.lvclass:SetPanelState.vi ... 10
3.9 ProgressBar_WireFlow.lvclass:SetProgressMessage.vi ... 10
3.10 ProgressBar_WireFlow.lvclass:SetProgressValue.vi ... 11
4 Engine Methods ... 13
4.1 ProgressBar_Win_WireFlow.lvclass:CancelButtonVisible.vi ... 13
4.2 ProgressBar_Win_WireFlow.lvclass:Clear.vi ... 13
4.3 ProgressBar_Win_WireFlow.lvclass:DeferPanelUpdates.vi ... 14
4.4 ProgressBar_Win_WireFlow.lvclass:DisableFPitems.vi ... 14
4.5 ProgressBar_Win_WireFlow.lvclass:Init.vi ... 15
4.6 ProgressBar_Win_WireFlow.lvclass:PanelCheckDelay.vi ... 16
4.7 ProgressBar_Win_WireFlow.lvclass:PanelClose.vi ... 16
4.8 ProgressBar_Win_WireFlow.lvclass:PanelOpen.vi ... 17
4.9 ProgressBar_Win_WireFlow.lvclass:PanelUpdate.vi ... 17
4.10 ProgressBar_Win_WireFlow.lvclass:SetMessage.vi ... 18
4.11 ProgressBar_Win_WireFlow.lvclass:SetRange.vi ... 18
4.12 ProgressBar_Win_WireFlow.lvclass:SetValue.vi ... 19
5 Examples ... 20
6 HowTo´s ... 21
6.1 Create a customized Progressbar window ... 21

# 1 License

BSD License

Copyright (c) 2012, WireFlow AB
All rights reserved.

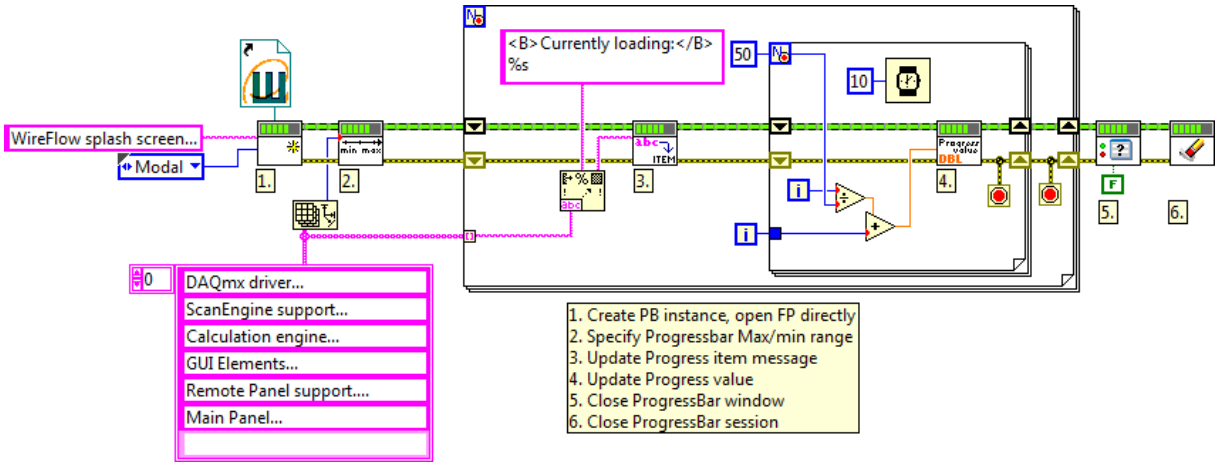
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of WireFlow nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

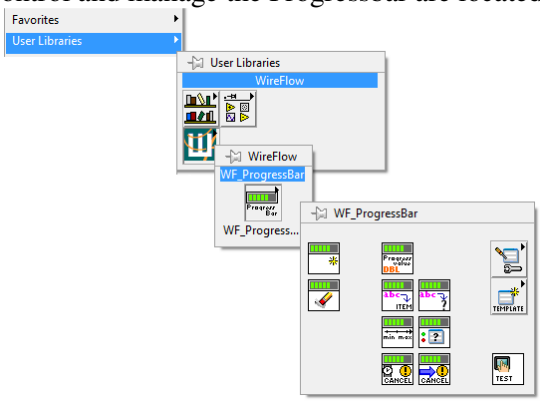
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE PO

# 2 OverView

The ProgressBar module is intended to be a simple way of displaying progress for different tasks, e.g. Listing directories, copying files etc.



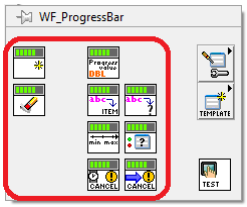

All methods needed to control and manage the Progressbar are located in the User.lib palette




The ProgressBar API uses a number of different events to update the ProgressBar window. These events are handled together with the control/indicator references of the ProgressBar window in the engine VI.

# 3 API Methods

The API methods are used to launch and control the ProgressBar window. It has methods to control the content (strings, value and range) and the visibility of the ProgressBar window. The API palette is located in <user.lib>\WireFlow\ProgressBar




## 3.1 ProgressBar_WireFlow.lvclass:CheckCancelNotification.vi

Checks if the ProgressBar window generated a Cancel notification.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

**cancelled?**

**message**

## 3.2 ProgressBar_WireFlow.lvclass:GenerateCancelNotification.vi

Generates a Cancel notification.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**message**

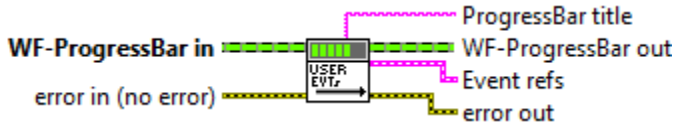
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

## 3.3 ProgressBar_WireFlow.lvclass:GetEventRefs.vi

Returns the User Event refnums initialized by the ProgressBar API Init method. Mainly used in the PB engine, but can also be used in the actual ProgressBar window to perform special actions for a specific event.




**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

**Event refs** This typedef contains the default progress bar event types that can be registered in the ProgressBar engine.

**Exit** This event is generated from the API when the PB engine should stop.

**MinMax** This event is sent frm the API to define the Min/Max range of the PB slide control.

**DisplayState** This event is sent from the API to show/hide the PB window.

**Message** This event is sent from the API to update one of the message fields in the PB.

**Value** This event is sent from the API to update the progress bar value.

**ProgressBar title** This is the title set for the Progressbar window at init. Only used when the ProgressBar window is initialized, and can be changed later.

## 3.4 ProgressBar_WireFlow.lvclass:PB_Clear.vi

Clears the Progressbar session and closes any open panel or references.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

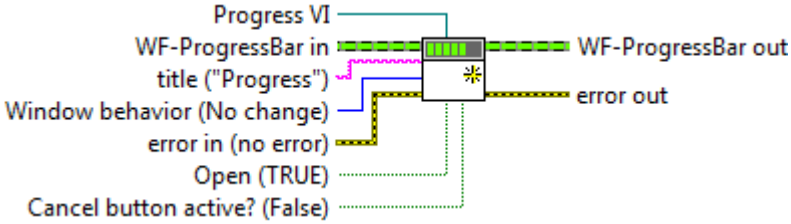
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

## 3.5 ProgressBar_WireFlow.lvclass:PB_Init.vi

Creates a new Progressbar session.
* "title" specifies the title of the progressbar window.
* Progress VI specifies the Progressbar window that should be launched.
* If Open = True, the progress bar window is directly opened.
* Windows behavior = specifies if this instance should be run as "modal", "floating" or as set in the VI properties.




**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**Progress VI** VI reference of the ProgressBar VI. If invalid reference is specified, the default ProgressBar window will be used.

**title ("Progress")** The starting title of the ProgressBar window. This can be changed later using another API method.

**Open (TRUE)** If TRUE the PB window opens directly when init is done, otherwise the PB has to be opened with the specific Window state method.

**Window behavior (No change)** Defines the PB windows behavior;
No Change = behavior is as set in the PB VI properties.
Modal = Acts as a dialog, and stays on top
Floating = PB window stays on top, but only floats (i.e. is not active)

**Cancel button active? (False)** If TRUE the Cancel button is visible and the value change event is monitored.

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

## 3.6 ProgressBar_WireFlow.lvclass:ProgressBarEngine.vi

This is the engine that runs in the progress bar window. This means that a new progress bar window only needs to call this Engine VI to get the functionallity of the Progress bar.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar class**

**Progress control refs.** This cluster contains the references to the default supported window elements. These control references are then used to update the PB when events are received.
If a reference is not supplied (invalid), the actions for this control/indicator are not handled, and no error is generated.

**General message** The General message field is a short Title text, and is probably only updated a few times.

**Current message** Current message describes the action that is currently taking place, e.g. "Copying file pathA to pathB"

**Progress value** This is the reference to the progressbar slider.

**Cancel Button**

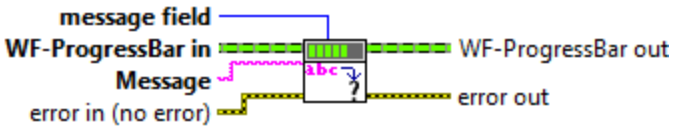
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar class out**

## 3.7 ProgressBar_WireFlow.lvclass:SetMessageString.vi

Updates the specified message field in the progress bar window.
The General message and Current item message can be formated as italic, underline and bold.




**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**Message** String that should be written.
The General and the Current Item strings supports formatting instructions.
Formatting is specified in html style like <X>some text</X>,
where X can be set as
* b = for bold text
* i = for italic text
* u = for underline text

**message field** Defines the message that should be updated;
* The Title is the name of the BP panel.
* The General message is a short Title text.
* The Current message describes the action that is currently taking place, e.g. "Copying file pathA to pathB"

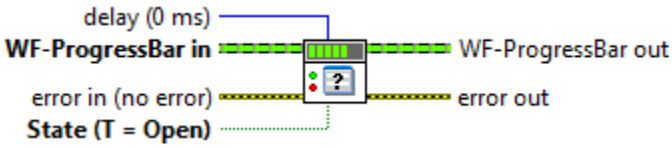
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

## 3.8 ProgressBar_WireFlow.lvclass:SetPanelState.vi

This VI hides, or unhides the progress bar window.
The user can optionally specify a delay in ms before the progressbar is shown.




**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**State (T = Open)** Specifies if the PB should be opened (T) or closed (F).

**delay (0 ms)** Specifies the delay in milliseconds from PB activation until it is actually displayed.

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

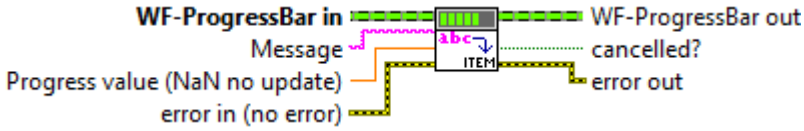
The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

## 3.9 ProgressBar_WireFlow.lvclass:SetProgressMessage.vi

Updates the progress bar "Current item" message, and optionally the progress value.
Leave value at NaN to skip update.

If user pressed Cancel in the Progressbar panel (if this button is activated), this boolean output is true.
It is up to the calling code to handle the cancel action.




**error in (no error)** The **error in** cluster can accept error information wired from VIs

previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**Progress value (NaN no update)** Specifies the progress bar value. If this value is NaN only the progress message will be updated.

**Message** Message describes the action that is currently taking place, e.g. "Copying file pathA to pathB"

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

**cancelled?**

## 3.10 ProgressBar_WireFlow.lvclass:SetProgressValue.vi

Updates the value of the progress bar in the range that is defined in the SetRangeMinMax method. Default is a range between 0 and 1, but a vlue outside the range will still be written.

If user pressed Cancel in the Progressbar panel (if this button is activated), this boolean output is true. It is up to the calling code to handle the cancel action.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**Value** Value is the current value of the progress bar.
This value should be set in the range defined in the Max/Min method.
Default is a progress value between 0 and 1.

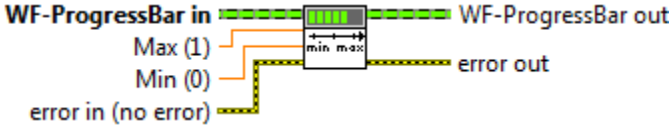
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

**cancelled?**

**ProgressBar_WireFlow.lvclass:SetRangeMinMax.vi**
This VI sets the min and max range values for the progress bar.
Wire NaN to leave a value unchanged.




**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar in**

**Max (1)** Sets the Max scale value of the progressbar slider.
Default is 1.

**Min (0)** Sets the Min scale value of the progressbar slider.
Default is 0.

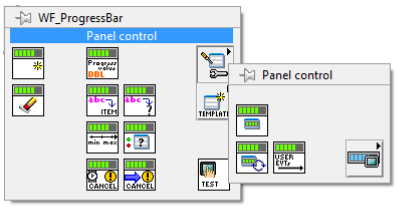
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

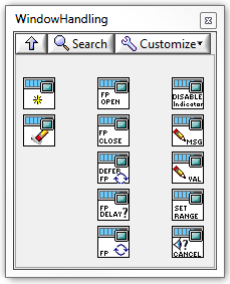
The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**WF-ProgressBar out**

# 4 Engine Methods

The Engine methods are used internally in the ProgressBar engine to handle UserEvents and updates of the ProgressPanel. These are only listed for reference, since they should normally not be changed.




## 4.1 ProgressBar_Win_WireFlow.lvclass:CancelButtonVisible.vi

Set Cancel button visible or hidden.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**cancel button visible?**

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.2 ProgressBar_Win_WireFlow.lvclass:Clear.vi

Clear ProgressBar engine handling

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the

error displayed.

**ProgressBar_Win in**

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.3 ProgressBar_Win_WireFlow.lvclass:DeferPanelUpdates.vi

Sets or unsets the Defer Panel Update property

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**Defer Panel Updates** Set to the defer panel status of the ProgrtessBar VI.
T = Defer updates

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.4 ProgressBar_Win_WireFlow.lvclass:DisableFPitems.vi

Disable the string elements on the progress panel. Just to make sure they are not clicked during updates.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

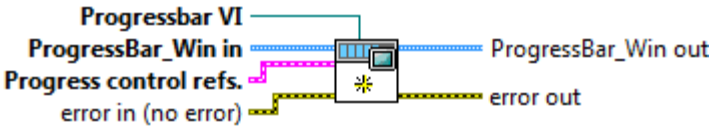
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.5 ProgressBar_Win_WireFlow.lvclass:Init.vi

Initialize the PB window handling




**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**Progress control refs.** This cluster contains the references to the default supported window elements. These control references are then used to update the PB when events are received.
If a reference is not supplied (invalid), the actions for this control/indicator are not handled, and no error is generated.

**General message** The General message field is a short Title text, and is probably only updated a few times.

**Current message** Current message describes the action that is currently taking place, e.g. "Copying file pathA to pathB"

**Progress value** This is the reference to the progressbar slider.

**Cancel Button**

**Progressbar VI** This is the VI reference to the ProgressBar window, and is used to Show/hide window etc.

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the

error displayed.

**ProgressBar_Win out**

## 4.6 ProgressBar_Win_WireFlow.lvclass:PanelCheckDelay.vi

Handles opening of the progressbar window if a delay is specified.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.7 ProgressBar_Win_WireFlow.lvclass:PanelClose.vi

Closes the Progressbar window.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

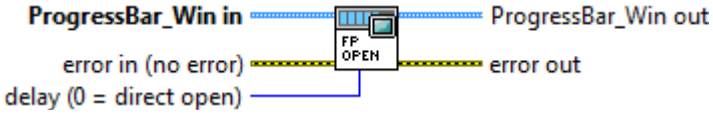
**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.8 ProgressBar_Win_WireFlow.lvclass:PanelOpen.vi

Sets the Progress Bar panel state = TRUE and optionally sets the delay.
If delay>0 the progress bar window is not opened until delay ms has passed.




**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**delay (0 = direct open)** Specifies the delay in milliseconds from PB activation until it is actually displayed.

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.9 ProgressBar_Win_WireFlow.lvclass:PanelUpdate.vi

Updates the ProgressBar window elements ("Value" and "Current Item message").
This is called in the timeout case to handle the case that a number of events come in a burst.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.10 ProgressBar_Win_WireFlow.lvclass:SetMessage.vi

Sets the message string for the specified message type. If type is "Current item" update is not direct, but when there is no more event to serve (i.e. in the timeout case, with timeout = 0).

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**Message** String that should be written.
The General and the Current Item strings supports formatting instructions.
Formatting is specified in html style like <X>some text</X>,
where X can be set as
* b = for bold text
* i = for italic text
* u = for underline text

**message field** Defines the message that should be updated;
* The Title is the name of the BP panel.
* The General message is a short Title text.
* The Current message describes the actioin that is currently taking place, e.g. "Copying file pathA to pathB"

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.11 ProgressBar_Win_WireFlow.lvclass:SetRange.vi

Sets the ProgressBar scale range min/max values.

**error in (no error)** The **error in** cluster can accept error information wired from VIs

previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**max (NaN)** This is the range max value for the progressbar.
NaN means that it is not set.

**min (NaN)** This is the range min value for the progressbar.
NaN means that it is not set.

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win out**

## 4.12 ProgressBar_Win_WireFlow.lvclass:SetValue.vi

Sets the progress value, update is not direct, but takes place when there is no more event to serve.

**error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**ProgressBar_Win in**

**Value** Value is the current value of the progress bar.
This value should be set in the range defined in the Max/Min method.
Default is a progress value between 0 and 1.

**error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

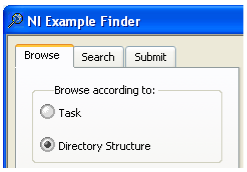
The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

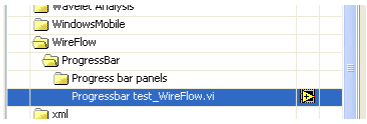
**ProgressBar_Win out**

# 5 Examples

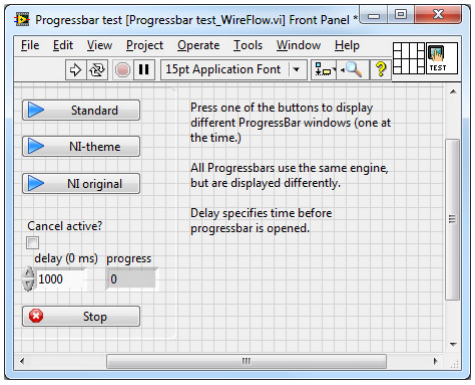
The VIPM package installs example code to the examples directory
<LabVIEW>\examples\WireFlow\ProgressBar\

To open this example either browse on disk, or use the Example finder.




In the example finder Browse the directory structure, and browse to WireFlow\ProgressBar. Then open the VI ProgressBar test_Wireflow.vi.




When this example runs it first displays a simple splash screen and then a simple user interface where the user can select to show three different ProgressBar panels, using the same code to manage updates. The example uses methods to demonstrate;

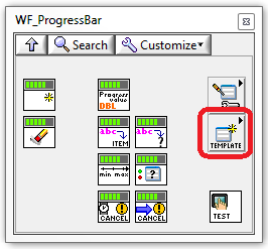
- general usage of the ProgressBar API
- setting max/min values for the progressbar slider
- setting different strings with different formatting
- displaying the ProgressBar panel with or without delay
- setting front panel mode (modal, floating or normal) of the ProgressBar window.
- Enabling/Disabling Cancel functionality in the ProgressBar

# 6 HowTo´s

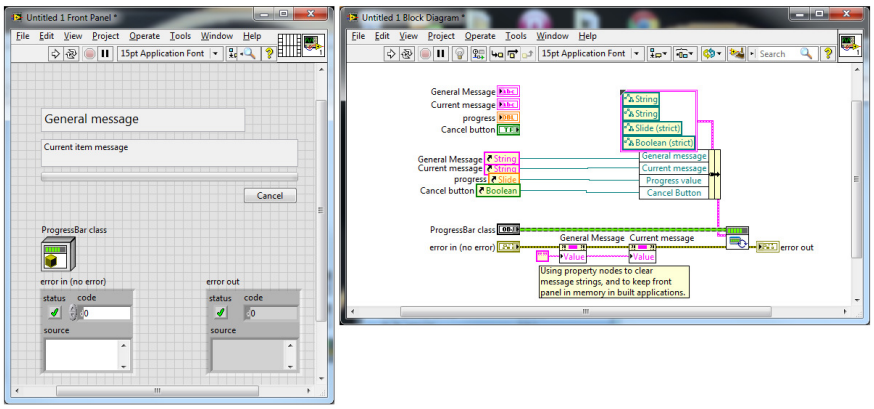
This section briefly describes some typical customizations and how they could be done using the ProgressBar API methods.

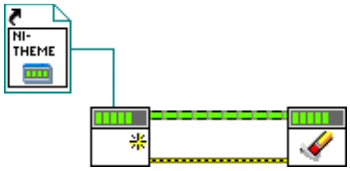
## 6.1 Create a customized Progressbar window

If we just want to create a new look for the ProgressBar we can modify the template.
To do this we create a new VI and then we drop one of the templates, found in the template palette, on the block diagram.




This will add a number of FrontPanel and BlockDiagram objects to the new VI.




Modify the FP to have the look and feel that you prefer, but don't rename the "ProgressBar class" control since this is used to pass information to the ProgressBar at launch.
The only thing that should be visible at runtime is the message fields and the progressbar.

To use this modified ProgressBar it is only necessary to wire a VI reference to the VI to the Init method, e.g. using StaticVI reference