

Property Inspector Help

Utility for filtering, searching, sorting and editing VI properties and metrics from the project

Version 4.0.1, September 2021, © 2013 - 2021 ABCDEF. All rights reserved.

Contents

Property Inspector Overview.....	3
Installation Requirements	4
Toolbar.....	5
Open Project button.....	5
Settings button	5
Filter on button	5
Open VI.....	5
Refresh (F5)	5
Back	5
Next	5
Search Parameters.....	6
Search drop-down (F2).....	6
Compare Modes (F3).....	8
Value Field	9
Edit Text (F4)	9
Edit Mode (F6).....	9
Search Button (F8).....	9
Search In Button (F9).....	9
Edit Selected Button (F10).....	9
STOP Button	9
Results Table.....	10
Status Bar.....	10
Menu Bar	11
Settings	14
Object Types.....	14
Columns to Display.....	14
Search Options	15
Edit Options.....	15
Sort Columns	15
Background Refresh	15
Select Group Size.....	15
VI Scripting.....	16
VI Script	16
QD Script.....	16
Error Details.....	17
Export... ..	17
Other Errors.....	17

Unique Features	18
Global Exclusion™	18
Persistent Selection™	18
Static Selection™	18
Sort by Unsorted	18
Compound Sorting	18
Window Cascade, Tile and Stack	18
Show only Selected Items	18
Retain Wire Values	18
Show BD of First Clone	18
Impossible Features.....	19
Callers	19
Clones	19
Close Opened Windows	19
Date Modified.....	19
Delete Selected VIs, even from LLBs	19
Execution State.....	19
Export Search Results	19
Find all Dialog Windows	20
Find and Prepend, Append, Remove or Replace All.....	20
Find Anything but That.....	20
Find Anything or Nothing	20
Find, Find and Find Again	20
Find More or Less	20
Find True or False	20
Find Recent Changes	20
In Memory.....	20
Inline Allowed.....	20
Open Block Diagrams	21
QuickDrop on Steroids	21
Scan Time	21
Sort by Number	21
Suspend when Called	21
View in Project	21
View Specific Item Types.....	21
Search Examples	22
Advanced Search Examples	22
Support Resources.....	22

Property Inspector Overview

Property Inspector is a powerful utility for viewing, searching, sorting and editing VI properties of all VIs in a project, class or library. All project items can be filtered by 20 object types allowing only relevant items to be visible. It allows searching all the items based on 44 VI properties including **VI Description**, **Data Size** and **Allow Debugging**. Some properties cannot be displayed in LabVIEW including **Clones**, **Date Modified**, **Execution Status**, **In Memory**, **Inline is Allowed** and **Modified**. After selecting all or some of the results, 23 of the 44 specified VI properties can be edited for all VIs in a single click. Powerful string manipulation functions allow for the entire field to be searched, edited, replaced or removed. For example, find all VIs based on a VI template with the default description like "Use this template to ", selected all, and then remove the entire description. It is also possible to modify a different VI property than the one searched.

Property Inspector supports many search criteria that are not allowed by LabVIEW, like finding any value, a missing value, comparing a value against a threshold, comparing string length or finding a value of True or False. Search **Description** for blank identifies VIs that need documentation. Search **LV Version** for Less Than the current version will show all older code. Search **Execution Status** Equals 'bad' shows all broken code. Search **Reentrant** is True followed by Search In with **Reentrancy Type** Contains 'Pre' shows all VIs with Preallocated Reentrancy. Searching a single property by True can quickly isolate all VIs with **Allow Debugging**, **Auto Error Handling** or **Modified** status. Easily disable **Allow Debugging** on hundreds of VIs, compare the performance of your program, and then turn it back on.

Flexible edit modes allow replacing a string with an empty value, replacing the whole string and not just the match, or removing a path value by replacing it with Not-A-Path. Sorting all VIs by value would identify VIs with the highest **Data Size**, **Code Size** or **Revision Number**. Unlike VI Profiler, the **Data Size** shows the memory used before the VI is executed and would identify static data saved in the front panel or block diagram.

Selected search results can all be edited, deleted, opened or run in a single operation. One click can show the block diagrams of all VIs or the first clone of reentrant VIs. A single checkbox option will enable **Wires Retain Values**. All opened windows can be closed with a single operation. Multiple objects can be highlighted in the Project Explorer. The visible details in the results window can be exported to a spreadsheet file for additional analysis or documentation. The settings can enable all elements by 20 object types and which of the 44 VI properties are displayed, allowing just the items of interest to be shown.

Exclusive features in Version 4 are **Global Exclusion™**, **Persistent Selection™** and **Compound Sorting™**. Searching the category ALL with a mode of Exclude allows a quick way to remove a specific set of items from the search. **Persistent Selection™** maintains the selection of all items in the window after changing Object Filters, Search or Sort settings. Every sort operation can be based on the previous sort results, allowing for **Compound Sorting™**. For example, sort by data ascending to see the oldest object. Then click to sort by object type and quickly see the oldest of each type.

Powerful **VI Scripting** is included using the standard interfaces for VI Analyzer and Quick-Drop Plugins. This allows complex operations to be performed on some or all VIs with just a small amount of your own code. Leveraging the existing library of [Community Quick-Drop Keyboard Plugins](#) speeds the execution of standard operations like [Align front panel controls to connector pane pattern](#) or [Move and Size](#). VI Scripts can also be used to select project items using custom criteria.

The utility keeps a history of the last 9 individual Search operations and each can be repeated in a single click. Separately there is a list of 12 favorite searches that include compound include and exclude searches. All 21 history and favorites can be repeated in a single keystroke. Disabling **Search on Select** allows editing the criteria before searching or compounded with Search In. This history can also be cleared. An audible beep will sound if a process takes over one minute.

Powerful Window management feature include **Tile**, **Cascade**, **Stack** and **Close All**. If 20 block diagrams are opened, tile will only arrange the diagrams. Stack positions all windows together for consistency.

It optionally includes all targets found in the project including cRIO and FPGA targets. Another option will include all dependencies shown in the project as well as Packed Libraries.

The utility can be found under the Tools menu -> ABCDEF -> Property Inspector...

Installation Requirements

Compatible LabVIEW Versions: \geq 2015.

Tested LabVIEW Versions: 2015, 2016, 2017, 2018, 2019 and 2020

Compatible OS Versions: Windows, Mac.

Hardware Requirements: None

Installation Requirements:

Both LabVIEW and VIPM must be running as Administrator to install

Requires VIPM 2018 or later

Dependencies:

jki_tool_right_click_framework \geq 1.0.2.208

JKI State Machine \geq 2018.0.7.45

LabVIEW must be restarted after installation.

Toolbar



Buttons on the toolbar speed operation of common functions. Some buttons are disabled if they don't apply to the selection or can't be used during the current operation.

Open Project button

Press the Open Project button to browse for the project file to open. The project file cannot be newer than the current LabVIEW version. If the project is older than the current version, any VIs edited would be updated to the current version. Use the File menu to open a Class or Library.

Settings button

Opens the settings dialog. Options include the project object types, columns to display, search options, edit options and column sorting.

Filter on button

When a filter is applied, the Filter on button turns on (depressed). Click the filter on button to disable the filter and see the unfiltered results. Hovering over the filter button will show the last filter used. Clicking the button when it is off will repeat the last filter and turn the button back on.

Open VI

When one or more VIs are selected, this button will open all VI front panels. Holding the control key will show the block diagrams instead by minimizing the front panels. Use the Window->Close opened windows to close them automatically. Double-clicking a single item in the results window opens the front panel and holding control opens the block diagram.

Refresh (F5)

When one or more VIs are selected, this button redraws the results for the selected row(s).

Back

Moves the selection one row higher in the results table. If a sequential group is selected the previous group is selected. This button is only enabled if a result item is selected except the first one.

Next

Moves the selection one row lower in the results table. If a sequential group is selected the next group is selected. This button is only enabled if a result item is selected except the last one.

Search Parameters

Search Excludes

Edit Mode Edit Text

Search Contains

Edit Mode

Use the search parameters section to specify the properties or metrics to search by, the type of comparison and the value to display. Use the Edit Mode and Edit Text to change the value of the selected results.

Search drop-down (F2)

Use this drop-down to select the VI property to search. Choices include:

Name	Description	Editable	Notes
ALL	Combines all string properties to simplify searching and excluding	No	Compare options are Contains and Excludes
Allow Debugging	VI Property found under Execution	Yes	
Auto Error Handling	VI Property found under Execution	Yes	
Block Diagram Size	VI Metric found under Memory Usage	No	
Callers	Hidden property visible by selecting View->Browse Relationships->This VI's Callers and then counting the VIs shown	No	
Class	Shown in the hierarchy of the project	No	
Clones	Hidden property	No	Found by searching reentrant VIs in memory
Code Complexity	Shows the complexity factor used when compiled	No	
Code Size	VI Metric found under Memory Usage	No	May show as 0 if Separate Code from Source is ON and the Vi is not in memory
Data Size	VI Metric found under Memory Usage	No	
Date Modified	File system property found on the disk	Yes (by saving)	Enter a date in mm/dd/yy format and use Greater or Less Than. Just enter a year (2000 and later) and Less Than will use 1/1 and Greater Than 12/31. Contains and excludes will only use the date (mm/dd/yyyy)
Description	VI Property found under Documentation	Yes	
Execution Priority	VI Property found under Execution	Yes	Includes background (0), normal, above normal, high priority, time critical and subroutine (5). Enter the first letter or 0 to 5 to edit the value

Execution State	VI State indicated by a broken run arrow	No	Includes invalid, idle, running, bad, run top level
File Size	VI Metric shown under Memory Usage	No	Size of the VI file, LLB or library file
Front Panel Objects	Count of all the objects on the front panel	No	
Front Panel Size	VI Metric found under Memory Usage	No	
History Text	VI Property found under General, Revision History...	Yes	Can only be cleared (remove)
In Memory	VI Metric found by searching memory for each VI	No	
Item Name	VI Property as shown in the project window	No	
Inline is Allowed	Hidden property	No	Shows if Inline SubVI can be set to true
Inline SubVI	VI Property found under Execution	Yes	Allowed only if debugging is false and reentrant is set to pre-allocate
Item Path	VI Property as shown in the Files display under the project. This path is relative to the project folder unless the item is outside the project folder	No	
Item Type	VI Property indicating the item type (Control VI, Standard VI, Global VI, Folder, etc.)	No	
Library	Shown in the hierarchy of the project	No	
Logging File Path	Used with Operate -> Data Logging	Yes	
LV Version	VI Property found under General (Source version)	No	
Modified	Hidden property shown when closing the project	Yes	Modified only by saving
Preferred Execution System	VI Property found Execution, Preferred Execution System	Yes	Includes user interface, standard, instrument I/O, data acquisition, other 1, other 2, same as caller. Use the first 4 letters to edit the value or 1, 2 for other
Protection	VI Property found under Protection	Yes	
Reentrant	VI Property found under Execution	Yes	
Reentrancy Type	VI Property found under Execution (Shared or Prealloc)	Yes	
Revision Number	VI Property found under General	Yes	Remove to set to zero. When writing this property, only replace with a value greater than the current revision number.
Run Time Menu	Setting found under Edit->Run-Time Menu...	Yes	
Same as VI Name	VI Property found under Window Appearance	Yes	
Scan Time	Time to scan the object's properties	No	In milliseconds

Separate Comp. Code	VI Property found under General (Separate compiled code from source file)	Yes	
Show Front Panel when Called	VI Property found under Window Appearance, Customize	Yes	
Show Front Panel when Loaded	VI Property found under Window Appearance, Customize	Yes	
Show Menu Bar	VI Property found under Window Appearance, Customize	YES	
Suspend when Called	VI Property found under the Operate menu	Yes	
Target	VI Property displayed in the bottom left corner of each VI after the project name	No	Typical values are My Computer, cRIO-xxxx or FPGA
Total Data Size	VI Metric shown under Memory Usage	No	This value is the sum of the other memory values
Transparency	VI Property found under Window Appearance, Customize	Yes	0 to disable, 100 is completely transparent
Window Title	VI Property found under Window Appearance	Yes	Set Same as VI Name to False before setting

Compare Modes (F3)

Use this drop-down to specify the comparison method. Text match case is an option specified in the settings.

String	Equals	Includes results where the property or metric equals the specified string value
	Contains	Includes results where the property or metric contains the specified string value. This mode supports the special characters: .*?\$^\
	Excludes	Includes results where the property or metric excludes the specified string value. This mode supports the special characters: .*?\$^\
	Starts With	Includes results where the property or metric starts with the specified string value
	Ends With	Includes results where the property or metric ends with the specified string value
	Shorter than	Includes results where the length of the string is less than the specified value
	Longer than	Includes results where the length of the string is more than the specified value
Binary/String	Is Blank/0/False	Includes results where the property or metric is a blank string or a 0 value. This includes settings that are False.
	Not Blank/True	Includes results where the property or metric is not blank string or a not a 0 value. This includes settings that are True.
Numeric	Less Than	Includes results where the Metric is less than the specified value. Strings are compared character by character.
	Greater Than	Includes results where the Metric is greater than the specified value. Strings are compared character by character.

Search Value Field

Enter the value to search for or to filter out. Enter the text to include or exclude from the results. Value cannot be blank if the Edit Mode is set to Match.

String	Enter text	Enter the text to include or exclude from the results.
Binary/String	1/T/Y/0/F/N	Enter a 1, T or Y for true and a 0, F or N for false.
Numeric	Enter number	Enter the number to compare. Metric syntax is allowed including K for 1,024 and M for 1,048,576. All values are integer except LabVIEW version, which returns a single decimal digit (i.e. 8.6).
Special	Reentrancy Type	Enter 0 or S for Shared or 1 or P for Pre-allocate

Edit Text (F4)

Enter the text to edit the property. If the Edit Text box is disabled, selected search property is not editable. A multi-line string can be entered by pressing Control-Enter, or changing the settings to show slash codes (i.e. /r/n)

Edit Mode (F6)

Enter the how the edit is to be applied to the selected Search item. Note that it is possible to change the Search item and edit a different property than was used to filter the results window.

Match	Replaces the part of the property that matches the filter by value with the edit text, same as replace for binary settings
Prepend	Inserts the edit text to the start of the filter by property, same as replace for binary settings
Append	Appends the edit text to the end of the filter by property, same as replace for binary settings
Replace	Replaces the entire property with the contents of the edit text, or True or False, or 0 or 1
Remove	Removes the entire property contents and leaves the field empty, false or Not-A-Path value

Search Button (F8)

Applies the search criteria to the entire project and displays the filtered results. This also sets the Filter On button.

Search In Button (F9)

Applies the search criteria to the displayed results window and displays the combined search results. This also sets the Filter On button. There is no limit to the number of combined searches.

Edit Selected Button (F10)

Applies the Edit Mode to the Search property, optionally using the Edit Text and the Value Field

STOP Button

The STOP button will appear during long processing time including, project loading, editing, opening, closing and running scripts. Pressing this button will immediately abort the current process.

Results Table

Item Name	Item Type	Exec State	Description	In Mem	Debug	Auto Err	Date Modified	Code Size	Data Size	FP Size
Test A.vi	Standard VI	Idle	Test a basic DAQ board image.	0	1	1	12/19/2012 9:00 PM	10.36k	141.82k	4.06k
Test B.vi	Standard VI	Idle	Test an elite DAQ board image.	0	1	1	12/19/2012 9:00 PM	4.62k	138.52k	4.06k
Test C.vi	Standard VI	Idle	Test a GPIB board image.	0	1	1	12/19/2012 9:00 PM	13.22k	143.48k	4.06k
Test_Common To All Boards.vi	Standard VI	Idle	Image check for blocks that are common t	0	1	1	12/19/2012 9:00 PM	7.58k	4.74k	4.06k
Get Test Name.vi	Standard VI	Idle	Build a test name from the board type and	0	1	1	12/19/2012 9:00 PM	2.14k	1.25k	4.75k
Test Boards_TASK.vi	Standard VI	Idle	See Read_Me.html in the Boi	0	1	1	12/19/2012 9:00 PM	9.34k	2.87k	6.88k
Board Design.lvclass	LVClass		Base class for all board types.	1	0	0	12/19/2012 9:00 PM	0	0	0
Component Orientation and Positi	Control VI			1	0	0	12/19/2012 9:00 PM	0	0	5.76k

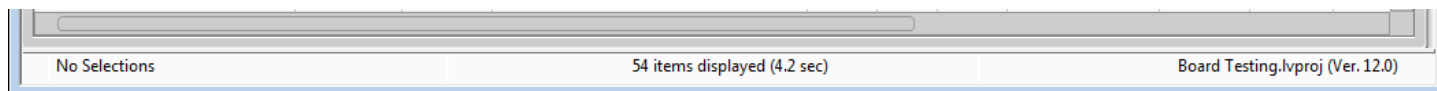
This table shows the project items flattened into a row format with the details separated by columns. Use the settings dialog to select which columns to display. Clicking the columns will sort the results by the property name ascending or descending. The column widths can be adjusted by dragging the dividers. Double-click a VI in the results table to open it. Right-click any cell to copy it to the Edit Text box and the clipboard.

When the Filter On button is selected, the results table shows the filtered results. To edit VI properties, select the property in the Search drop-down, select the Edit Mode, enter the change into the Edit Text box, select the rows in the table to change and press the Edit Selected button. Note that the editing the results table does not edit the VI properties in the VI.

Click on a column header to sort the results by the selected parameter. Click the same column to reverse the sort order. These sorting options can also be selected by the View menu.

New feature: Right-clicking on any field in the table with copy it to the Edit Text field and the Clipboard.

Status Bar



Right Status: The status bar at the bottom of the window shows the current details of the project and the utility.

The left one-third shows the status of the toolbar and the Search Parameters. Messages include 'Initializing', 'Opening', 'Ready', 'Searching', 'Replacing', 'No Items', 'No Results', 'No Selections', 'Enter O, F, 1, or T', 'Enter Edit Text', "Enter year or date (mm/dd/yy)", 'Enter Current password' and 'Enter New Password'. If the status shows 'Cannot edit ???', then the Search property is not editable. This is common for all VI metrics. Select a different property to search by or perform an edit operation.

After performing a search, this box will show the exact criteria of the search. If a compound search is performed, the entire search criteria will be displayed. Recalling a search from History or Favorites will also show the search here. If the search exceeds the space, hover over the status bar to see the full search as a tip string.

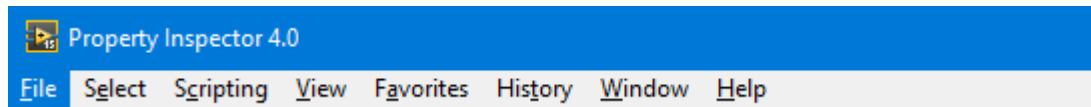
Left Status: When opening a project, this status will show the name of the top-most folder being scanned. Stopping the scan when loading reaches the desired point in the project. This will allow just the items scanned to be visible.

If the status is 'No Selections', one or more results must be selected for the edit operation.

Middle Status: The middle one-third shows the details of the selections in the results window. It shows total number of items in the project, number of items in the filtered results (with percentage), number of items selected and the name of the item if only one is selected. Some operations include the total time to complete the operation.

The right one-third shows the name of the selected project and its saved LabVIEW version.

Menu Bar



The menu bar includes several operations that are also on the toolbar, and several that only on the menu. Shortcut keys are shown by underlines, just like in the menu. Hold the Alt key to use the shortcuts. Control shortcuts are shown with the (^) symbols and shift with (⇧) symbols. Items in parenthesis are replaced by the actual content described.

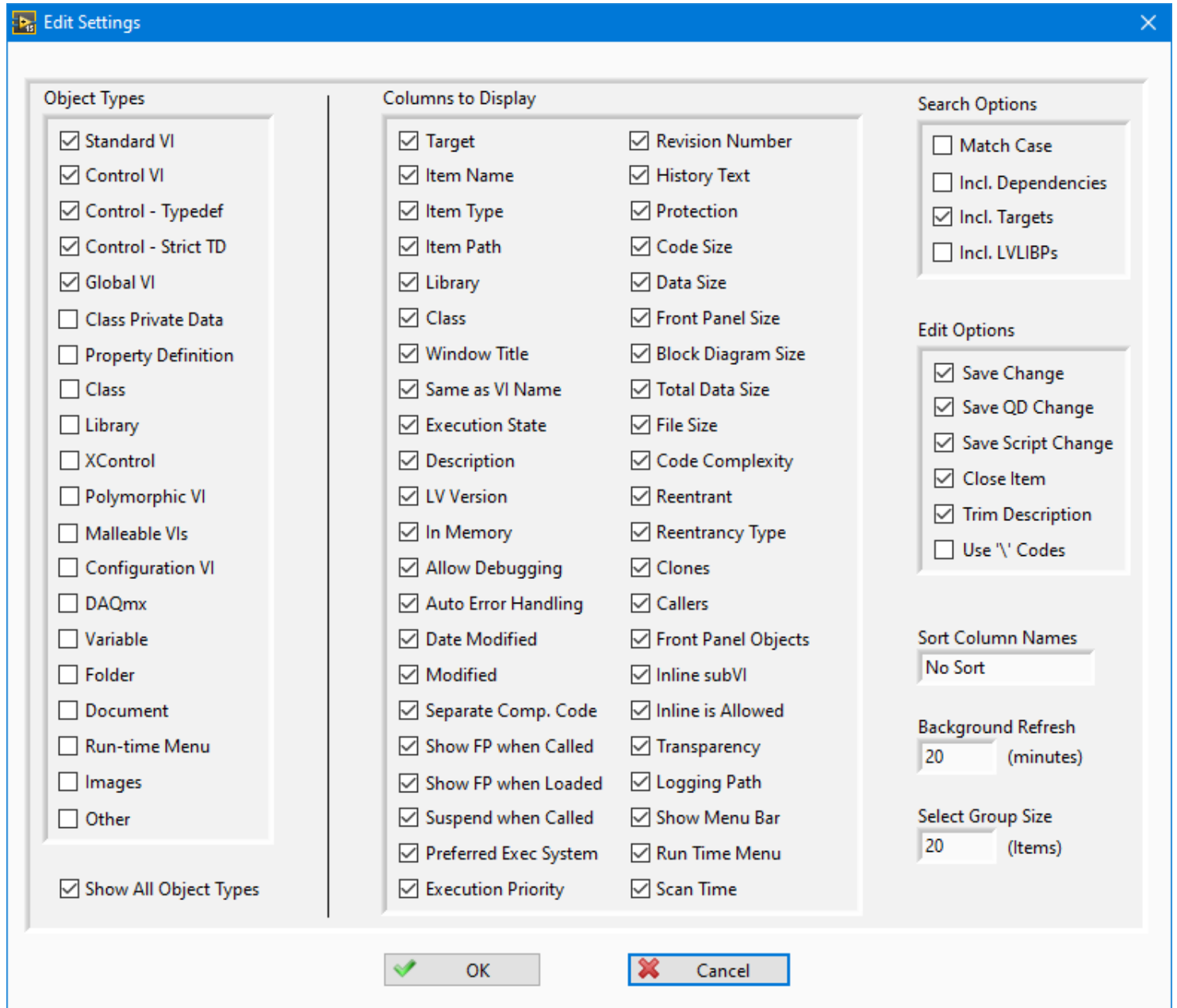
File Menu	<u>O</u> pen Project ... (^O)	Browses for a LabVIEW project file to load. An error shows if the Project file is newer than the current LabVIEW version.
	O <u>p</u> en <u>L</u> VLib ...	Browses for a LabVIEW library file to load (.LVLlib).
	O <u>p</u> en <u>L</u> V <u>C</u> lass	Browses for a LabVIEW class file to load (.LVClass).
	<u>R</u> eload Project	Reloads the currently opened project to synchronize all items
	<u>C</u> lose All	Closes the current project, library or class
	<u>D</u> elete Selected <u>V</u> is (^X)	Deletes the selected Vis from the project. A confirmation dialog asks if Vis in auto-populating folders should be deleted from disk.
	<u>S</u> ave Selected (^S)	Saves the selected items
	Save <u>A</u> ll	Saves all the items in the results window
	<u>E</u> xport Results	Exports the filtered results window to a spreadsheet file
	<u>E</u> xit (^Q)	Exits the utility and closes the open project
Select Menu	<u>S</u> elect All (^A)	Selects all rows of the results window
	<u>D</u> eselect All (^D)	Deselects all results
	<u>I</u> nvert Selection (^I)	Inverts the selection. Selects all results that are not selected and deselects all results that are
	Select from Project	Select only the items already selected in the project explorer
	Select to Project (^⇧E)	Highlight the current selections in the project explorer
	Select <u>P</u> revious (^Z)	Restore the previous selections
	Select <u>G</u> roup (^G)	Selects a continuous group based on the Selected Group Size setting
	Select <u>C</u> allers (^⇧C)	Select the callers of the selected items
	Select Background Updates	Select any items updated in the background
	Select Opened Windows	Select all items that were Previously opened by Property Inspector
	Select Open Windows	Select all items that are currently open and not minimized
	Show only Selected Items	Displays only items that are selected. Selecting and deselecting items will not change items displayed

Scripting Menu	Open Script Template	Opens a sample Script Template VI for creating custom scripts using the VI Reference. The connector pane must not be modified.
	Select Script File	Selects a custom Script File to run
	Run <u>V</u> I Script (^↑V)	Run the custom Script File on the selected items
	Update <u>S</u> elections on Failure	When checked, only the VI Scripts that return a failure will remain selected
	<u>R</u> ecent VI Scripts	Submenu showing a history of the last 9 VI scripts. Selecting a script will execute it immediately and save it as the current script to execute with ^R
	<u>O</u> pen QD Script Template	Opens a sample QD Script Template VI for creating custom scripts using the VI Reference. The connector pane must not be modified.
	Select QD Script <u>F</u> ile	Selects a custom QD Script File to run
	Run <u>Q</u> D Script (^↑R)	Run the custom QD Script File on the selected items
	<u>R</u> ecent QD Scripts	Submenu showing a history of the last 9 QD scripts. Selecting a script will execute it immediately with the QD options dialog and save it as the current script to execute with ^R
View Menu	<u>O</u> pen VIs	Opens the selected VIs and controls
	Open <u>B</u> lock Diagrams ^E	Opens the block diagrams of the selected VIs by minimizing the front panels
	Open <u>F</u> irst Clone BD (^M)	If the VI is reentrant and in memory, opens the first clone diagram of all selected VIs. Clone VI windows can be closed, but not arranged
	Retain <u>W</u> ire Values	When checked, all opened block diagrams have Retain Wire Values enabled
	Run VIs (^R)	Opens and runs the selected VIs
	<u>S</u> ort By	Select the property column to sort by. Clicking the column header will also change the sort by setting. Select Unsorted to revert back to project order.
	<u>A</u> scending	When checked, the sort is ascending and unchecked is descending
	<u>C</u> ompound Sorting	When checked, each sort starts with the previous sort results. If unchecked, each sort starts with the original project order
Favorites Menu	(Favorites list)	This menu provides a quick and easy way to perform repetitive operation on several projects or VIs. This list is a personalized history of 12 compound search operations. This list is stored on disk and kept for every use. Duplicates are filtered out. Shortcuts ^F1 to F12 select the items from top to bottom. If only No Favorites is shown, then there are no favorites.
	<u>A</u> dd to Favorites	Appends the current search to the favorites list. New items are added to the bottom. Adding to a full list will overwrite the last item.
	<u>C</u> lear Favorites	This clears the list of Favorite items
History Menu	(History list)	This menu provides a quick and easy way to perform repetitive operation on several projects or VIs. This list keeps a history of the last 9 operations. This list is stored on disk and kept for every use. Duplicates are filtered out. New items are added to the top. Shortcuts Control-shift-1 to 9 select the items from top to bottom with the most current item as shortcut 1. If only No History is shown, then there is no history.
	<u>C</u> lear History	This clears the list of history items
	<u>S</u> earch on Select (^0)	To speed repetitive operations, checking this option will perform the search on the history selection immediately. Disable to perform Search In results.

Window Menu	(Opened files)	This menu shows all the opened VIs.
	(Opened projects)	This section of the menu shows all the opened projects
	<u>T</u> ile Selected Windows (^⇧T)	Tiles all open front panels or block diagrams on the primary monitor without changing the panel size.
	<u>C</u> ascade Selected Windows	Cascades all open front panels or block diagrams on the primary monitor
	<u>S</u> tack Selected Windows	Stacks all selected front panels or block diagrams in the upper left corner of the primary monitor
	<u>C</u> lose Opened Windows (^⇧E)	Closes only the open windows that were opened by Property Inspector. This option is not available if no windows were opened
	<u>C</u> lose <u>S</u> electd Windows	Closes only the open windows of the selected items. This option is not available if nothing is selected
	<u>S</u> how Edit Error Dialog	Displays the error dialog containing previous edit errors
Help Menu	<u>S</u> how <u>H</u> elp (F1)	Opens the help document (this one)
	<u>O</u> nline Support and Resources	Opens the product support webpage
	<u>Y</u> ouTube Videos	Opens a page with demonstration videos of the utility
	<u>S</u> earch Examples	Loads examples shown below into the search and edit criteria
	<u>A</u> lliance Member Page	Opens the NI Alliance Member directory page for ABCDEF
	<u>P</u> roduct Homepage	Opens the product page at abcdef.biz
	<u>N</u> I <u>T</u> ools Network	Opens the product page on the LabVIEW Tools Network
	<u>C</u> ommunity Quick Drop Keyboard Shortcuts	Opens the NI forum page listing the latest Quick Drop Plugins
	<u>N</u> I VI Analyzer Page	Opens the NI.com VI Analyzer community page
	<u>A</u> bout Property <u>I</u> nspector	Displays the version number of the utility

Settings

The settings window allows control of project object types to include, columns to display, filter options, edit options and column sorting.



Object Types

Select the different types of project items to show in the results window. Types include Standard VI, Control VI, Control – Typedef, Control – Strict TD, Global VI, Class Private Data, Property Definition, Class, Library, XControl, Polymorphic VI, Malleable VIs, Configuration VI, DAQmx, Variable, Folder, Document, Run-time Menu, Images and other items. Changes to Object Types will be applied immediately. A single checkbox, **Show All Object Types**, allows an easy method to toggle between all items and a select few.

Columns to Display

Select the VI property and VI metrics to show in the results window. Items include Target, Item Name, Item Type, Item Path, Library, Class, Window Title, Same as VI Name, Execution State, Description, LV Version, In Memory, Allow Debugging, Auto Error Handling, Date Modified, Modified, Separate Compiled Code, Show FP when Called, Show FP when Loaded, Suspend when Called, Preferred Exec System, Execution Priority, Revision Number, History Text, Protection, Code Size, Data Size, Front Panel Size, Block Diagram Size, Total Data Size, File Size, Code Complexity, Reentrant, Reentrancy

Type, Clones, Callers, Front Panel Objects, Inline SubVI, Inline is Allowed, Transparency, Logging Path, Show Menu Bar, Run Time Menu and Scan Time. Turning off the display of a column does not prevent it from being searched or edited.

Search Options

Match Case selects if text filters are case-independent or case-sensitive. Editing in Match mode will use the same case setting. Select **Include Dependencies** to include them in the results. Enable **Include Targets** to show cRIO and FPGA targets. Enable **Include LVLIBPs** will include packed libraries from the project. Packed libraries take significantly longer to load and cannot be edited. Unchecking this option can load the project faster.

When changing dependencies, targets and LVLIBPs, use the File -> Reload Project to see the changes.

Edit Options

Save Change will automatically save the VI after each edit operation. **Close Item** will close each VI. If Save Change is off and Close Item is on, LabVIEW will prompt for every VI to be saved or not. **Trim Description** will trim the whitespace from both ends of the Description before editing the string. **Use '\ Codes** will change the display of the Search Text and Edit Text to display \ codes.

Sort Columns

The result columns are pre-arranged in the same order as the Columns to Display settings. For convenience, all the columns may be sorted alphabetically by the displayed name of the column. Options include no sort, ascending order and descending order.

Background Refresh

When the application is idle, all the items visible in the item list will be intelligently refreshed in the background at the specified interval. Updates will be displayed immediately, but searching and sorting will not be affected until the next time they are performed. The select menu contains a function to automatically select all items updated in the background. Setting the delay to 0 minutes will stop the background refreshing.

Select Group Size

When a large number of items are displayed, sometimes it is more efficient to open or edit them in small groups. This setting defines the number of items selected by **Select Group** (^G). If nothing is selected, the first items will be selected. If more items than the group are selected then only the first items selected will be selected. Clicking Back or Next on the toolbar will automatically select the next group.

VI Scripting

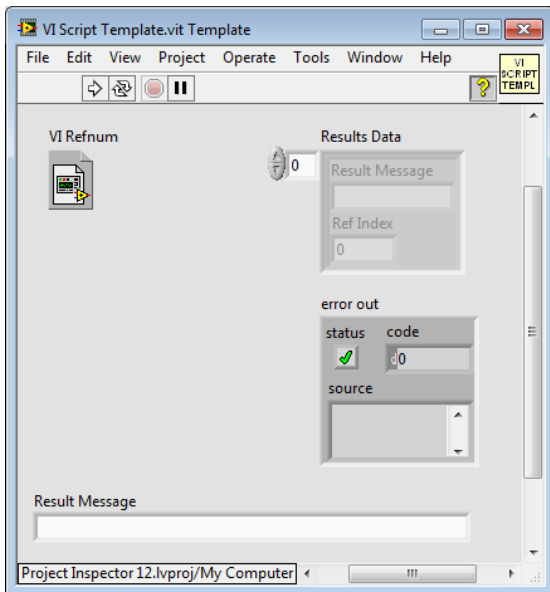
Scripting allows custom extension of the operation of the Property Inspector beyond the properties that can currently be edited. In only a few minutes, a scripting VI could be written to search or edit other properties or perform more complex edits. Leveraging the existing library of Quick Drop shortcuts provides a quick and easy capability of performing common operations without writing any code.

Standard templates are provided to assist the creation of these scripts. The front panel of the template must not be altered, since the script is called dynamically. Save the modified script in any location and select it using the menu option. Then select all the VIs to be edited and run the script. Any errors generated will be displayed by the Error Details dialog. Both selected QuickDrop and VI scripts files are stored separately, allowing both to be used alternately. It is possible to disable save and close for the first scripts and enable both for the last script run to minimize revisions.

Property Inspector 4 now saves a history of the last 9 VI and QuickDrop scripts, allow repetitive operations in fewer clicks.

VI Script

Use the included template VI based on the standard VI Analyzer template to write custom operations. PI will open the VI before the script and optionally save and close it (based on settings dialog). If Update Selections on Failures is checked, then only the VIs that return an array of Results Data will remain selected. This can be used to perform custom or complex search criteria. The status bar in the center will display the number of failures and the selected items.



QD Script

Use the included template VI based on the standard Quick Drop template to write custom operations. If there are Quick Drop shortcuts installed with LabVIEW, they can also be used. Alternatively, use the included link in the Help menu to download pre-written QD scripts from the Quick Drop Community. Use the Quick Drop Configuration panel to set the required options once for all VIs.

PI will open the VI before the script and optionally save and close it (based on settings dialog).

Unique Features

Some of the unique features in Property Inspector are **Global Exclusion™**, **Persistent Selection™**, **Compound Sorting™** and Sort by Unsorted.

Global Exclusion™

This feature allows simple, one step elimination of object based on what parameters you need to exclude. The global search includes all of the string parameters including Item Name, Item Path, Item Type, Description, History Text, and Protection. Many times, a search is narrowed by just eliminating an entire category by using the exclusion.

Persistent Selection™

This revolutionary new feature maintains the selected items even after changing the object filter, search criteria or Sort order. Start by performing a search and selecting a range of objects. After changing the settings to hide the controls, the remaining selection remains. Then you prefer to sort the list by a different parameter before edit the objects. The items originally selected are still selected, even if they are no longer sequential

Static Selection™

When changing the filter, sort or performing a new selection function, this feature ensures that the first item selected is still visible, usually in the same row.

Sort by Unsorted

When the project is first opened, the entire list is in the same order as the items in the project. This is a great convenience if both the Project Explorer and Property Inspector are both open. After several sort operations have been performed, sometimes it is nice to go back and see the list in the original order. That is what Sort by Unsorted does. It returns the results list back to the original, unsorted, ascending order.

Compound Sorting

When the items are sorted, the list is automatically sorted by the selected parameter, followed by the original order. When enabled, this feature will sort by the selected parameter, followed by the previous order, and then by the original order. For example, sort the entire list by date to see the oldest first, and then sort by item type. The list shows all the objects grouped by item type, with the oldest item first in each group.

Window Cascade, Tile and Stack

When multiple VIs are opened, Stack will position all the windows together and Tile will spread the windows evenly on the screen. If only diagrams are opened, only the diagrams will be arranged. Clone VIs cannot be arranged by these methods.

Show only Selected Items

After running a search to select items like Caller, Open Windows or a custom VI Script, there may be several objects selected throughout the project. To further identify or select these items, Show only Selected will show only the selected items all together. These items can then be selected individually or in a group for additional modification.

Retain Wire Values

LabVIEW has a feature to retain wire values when probing the diagram of a VI. Unfortunately, this feature is off by default and needs to be on before probing the wires. Checking Retain Wire Values before opening all diagrams will open all with the feature enabled.

Show BD of First Clone

Before probing a reentrant VI you need to switch to the clone VI. If you open the project, select Browse Relationships, Reentrant Items and select the clone, you have both versions open. Opening the clone directly is only possible by double-clicking the VI on the diagram. Property Inspector will open the block diagram of the first clone directly. Clone VI windows can be closed, but not arranged (tile, cascade or stack).

Impossible Features

Most of the features of Property Inspector are unusual because they are difficult to do in LabVIEW. Many of the features are remarkable, because they are **impossible in LabVIEW!**

Callers

Many Vis are used repeatedly throughout the project. You can see the number of callers by selecting the VI in the project, right-clicking and selecting Find -> Callers. To get the callers for all Vis, this process would have to be repeated for each VI. Property Inspector displays the number of *Callers* and can sort this value to highlight the largest numbers.

Clones

When using pre-allocated, re-entrant clones each clone occupies its own memory space and since 2013 is assigned a unique numeric code. Common questions might include how many clones are there? Which Vis have the most clones? Although it is possible to count the clones by using the View -> Browse Relationships -> Reentrant Items menu, this is not practical. LabVIEW cannot identify which VIs are reentrant or pre-allocated. You would have to open each VI and browse 3 levels of the menu to see how many clones there are. This may also require scrolling down over 10 pages to get the final count **of one VI**. Property Inspector displays the number *Clones* and can sort this value for all VIs to highlight the largest numbers.

Close Opened Windows

The options to close VIs include closing each VI directly, Close All (this project) and using the All Windows dialog. What if you just opened 25 block diagrams and want all 25 Vis closed? This process takes several steps if not 50 or more. When a group of front panels or block diagrams is opened by Property Inspector, a single click can close all of those VIs.

Date Modified

With the exception of the LLB Manager, LabVIEW cannot display the date last modified of any object in the project. Property Inspector shows the date for all VIs, controls, libraries, classes, documents, LLBs, folders and most virtual folders. Yes, PI3 can display the date of the LLBs, folders and many of the virtual folders. If the virtual folder matches a name of an existing folder at the same hierarchy location, that date is displayed.

Delete Selected VIs, even from LLBs

I don't remember when LabVIEW had the option to delete a VI from the project while removing it from the disk, but it was a very long time ago. LabVIEW can show the VI in Windows Explorer and separately remove the VI from the project. The file still needs to be manually deleted using the Windows Explorer. What if the VI is stored in an LLB? Explore only shows the LLB, where you have to open it, locate the file, delete the file and close the LLB manager. Property Inspector can delete an unlimited number of VIs from multiple locations including multiple LLBs in a single operation.

Execution State

LabVIEW will only report a VI is broken if it is in memory and the Error List is displayed. The Error List window displays the list of VIs in the top third of the window, does not count the VIs and is not multi-select. LabVIEW cannot report which Vis are running or idle, unless each one is open and the run arrow is visible. Property Inspector displays and searches the *Execution States* of Idle, running, broken, invalid or run top-level for all VIs in the project.

Export Search Results

Found a set of problems, work items, changes or features that you need to document or share with another programmer? Other than printing individual front panels or block diagrams, LabVIEW has never had a feature to print any part of the project explorer, much less a specific search results or selected VI properties. Property Inspector will export the results window exactly how it appears on screen with customizable columns, a specific search criterion and sorted by any property to a tab-delimited spreadsheet file.

Find all Dialog Windows

What if you need to modify all the popup dialog windows? How can you find all the user facing windows? Property Inspector can find all VIs set to *Show Window when Called* is true. This search will show all dialog VIs.

Find and Prepend, Append, Remove or Replace All

LabVIEW has a very powerful search that will find text anywhere in the front panel, block diagram, tip strip, description and history text. You may not believe this, but LabVIEW cannot remove the text that it found, much less the surrounding text. In the Replace window, you must enter some text, or LabVIEW will not perform the replace or replace all operation. When searching the description contains specific text, Property Inspector can (1) replace the text with new text, (2) remove the specific text, (3) prepend new text on the original description, (4) append new text to the original description or (5) replace the entire description with new text, not just the specific text that was found. (i.e. Description excludes written by, append Written by ABCDEF or Description contains Use this template to, remove all).

Find Anything but That

LabVIEW can only search for something to find. Because Property Inspector can search the entire project by 11 different criteria, it can find all the VIs that exclude specific criteria (i.e. *Description* excludes written by).

Find Anything or Nothing

LabVIEW can only search for text in string or numeric values. Because Property Inspector can search each VI for a single VI property, it can find all the VIs that have any value or no value (i.e. (*Class* is not blank or *Description* is blank).

Find, Find and Find Again

LabVIEW can only perform a single search and display results. Property Inspector can search within the previous results an unlimited number of times. (i.e. *LabVIEW Version* < 19 and *Execution State* equals Bad and *In Memory* is False).

Find More or Less

LabVIEW can only search for text in string or numeric values. Because Property Inspector can search each VI for a single VI property, it can find all the VIs that have a value more or less than a specified threshold (i.e. *LabVIEW Version* < 14 , *Data Size* greater than 100K).

Find True or False

LabVIEW can only search for text in string or numeric values. Because Property Inspector can search each VI for a single VI property, it can find all the VIs that have a value of True or a value of False (i.e. *Separate Compiled* code is False or *Allow Debugging* is True).

Find Recent Changes

Every need to see a timeline of your changes or of everyone in your team? Want to find all those files you worked on last week? Just synchronized with SCC and want to see the diagram of all changed VIs? Property Inspector can sort the entire project by *Date Modified* to show the most recent changes first.

In Memory

LabVIEW keeps a record of which VIs are loaded into memory, but cannot show if a particular VI in the project is in memory or not. Property Inspector cross-references each VI and control against the list of VIs in memory.

Inline Allowed

Setting a pre-allocated, re-entrant VI to inline can improve performance by eliminating the overhead of calling the subVI. LabVIEW internally knows which VIs can and cannot be changed to inline, but will not tell you. Only by selecting the inline check box will you know if there is an error preventing inlining. Property Inspector displays the state of this eligibility so you can quickly search for all VIs that qualify and even set them all at once.

Open Block Diagrams

LabVIEW has had the capability to show you the block diagram directly by holding control and double clicking on the SubVI in the top diagram since the early days. But in the process the front panel is also displayed. What if you wanted to compare 20 or more block diagrams? Property Inspector will open an unlimited number of block diagrams while simultaneously hiding the front panels.

QuickDrop on Steroids

Since 2009, LabVIEW has built upon QuickDrop to create an efficient and extensible way to speed development. Many developers have written plugins to add powerful editing operations in just two steps. My favorite is **Move and Size**. This QD plugin performs 4 diagram cleanup steps at once. There is no reason not to run this script on every VI in your project. But how? Property Inspector allows any QuickDrop script to run on all selected VIs in the project.

Scan Time

Many LabVIEW users want to know why it takes so long to load their project. What is causing the delay? Are there specific items in the project that slow or freeze the process? Property Inspector now records the time required to scan all the properties of each object. Results clearly show that it takes considerably longer if the VI is in an older *LabVIEW Version*, broken or missing SubVIs.

Sort by Number

LabVIEW recently added the capability to sort the project find results by one of the 3 visible columns, but only sorts by text (i.e. vi12 before vi2). Property Inspector can sort VIs by numeric data value. This includes fields like *Date Modified*, *Code Size*, *Data Size*, *File Size*, *Code Complexity*, *Revision Number*, *LabVIEW Version*, *Callers* or *Clones*. Sorting all VIs by *Data Size* quickly identifies all VIs that have large arrays stored in the diagram or front panel.

Suspend when Called

How many times have you compiled an executable and then realized that one of the VIs had suspend when called still enabled? Property inspector can find all these VI by searching for *Suspend When Called* is True, and clear them all at once.

View in Project

LabVIEW has had the capability to find the open VI in the project for many years. But what if you need to highlight several or even a hundred Vis in the project? After performing a search and/or a sort operation, just select as many Vis as desired and View in Project will highlight them all.

View Specific Item Types

The LabVIEW project window is missing a basic function to hide project items by type. Imagine you need to see just controls, standard VIs, typedefs, strict-typedefs, classes, libraries or variables. What if you only need to hide images, documents, property nodes or global variables? Property Inspector scans every item in the Project Explorer window and will display only the items selected in 20 different categories!

Search Examples

Here are several useful examples of Property Inspector operations. The Property Inspector settings are expressed exactly as it appears on the screen. Some examples show the edit operation with sample edit text and notes in parenthesis.

All of these examples can be found in the Help menu under Search Examples.

1. Description Starts With Use this template to, Remove
2. Description Is Blank/0/False, Replace (Description of each VI) (then press Next to advance to the next VI)
3. Allow Debugging Not Blank/True, Remove (remove will disable Allow Debugging)
4. Execution State Equals Bad
5. Data Size Greater Than 100K
6. VI Version Less Than 12
7. In Memory Is Blank/0/False (locates all VIs not in memory)
8. Auto Error Handling Not Blank/True, Remove
9. Is Reentrant Not Blank/True
10. Reentrancy Type Contains Pre, Replace S (Search In the results of example 9 to narrow the results)
11. Date Modified Less Than 2010
12. Separate Comp. Code Is Blank/0/False, Replace T

Advanced Search Examples

Here are more advanced examples of compound searches:

1. Description Not Blank/True AND Description Shorter Than 10
2. Description Is Blank/0/False AND Date Modified Greater Than 2020
3. Reentrant Not Blank/True AND Reentrancy Type Contains Pre AND Inline Is Blank/0/False
4. Execution State Equals Running AND Item Name Contains Dialog
5. Execution State Equals Bad AND LV Version Less Than 15 AND In Memory Is Blank/0/False
6. Item Name Greater than B AND Item Name Less Than E
7. Class Not Blank/True AND Class Excludes Driver AND Class Excludes DAQ
8. In Memory Not Blank/True AND Data Size Greater Than 200K
9. Class Not Blank/True AND Front Panel Object Equals 5
10. Item Name Contains Actor Core AND Library Contains MyLib
11. Class Not Blank/True AND Item Name Starts With Read
12. Item Path Excludes llb AND Item Path Longer Than 200

Support Resources

For support with Property Inspector, use the following resources:

Website: <http://support.abcdef.biz>

Phone: 1-800-422-1523

Email: support@abcdef.biz

View quick demonstration Videos on the [ABCDEF YouTube Channel](#)