VI Analyzer - Exclude Tests from VIs Dialog Box

Installed With: Base Package



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

This dialog box appears when you click the **Next** button from the <u>VI</u> <u>Analyzer - Select Tests</u> dialog box in the VI Analyzer. The dialog box does not appear if you analyze a single VI or if you run a single test.

Use this dialog box to exclude specific tests from running on specific VIs in a VI Analyzer task.

This dialog box includes the following components:

- Items to Analyze—Lists the VIs you selected for the analysis in the Items to Analyze hierarchy in the VI Analyzer Select VIs dialog box.
- Path to Item—Displays the path to an item you select in the Items to Analyze tree.
- Tests to Include—Lists the tests you selected for the analysis in the VI Analyzer - Select Tests dialog box. If you want to exclude a specific VI from a specific test or category of tests, select the VI in the Items to Analyze hierarchy and remove the checkmark from the checkbox next to the test. The VI Analyzer Results Window lists the test you exclude in the Skipped Tests category.

VI Analyzer - Select Tests Dialog Box

Installed With: Base Package



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

This dialog box appears when you click the **Next** button from the **VI** Analyzer - Select VIs dialog box in the VI Analyzer.

Use this dialog box to specify the tests you want to run for a VI Analyzer task.

This dialog box includes the following components:

- **Tests**—Lists all tests available to include in the VI Analyzer task. Remove the checkmark from the checkbox next to a test if you do not want to include a test in the VI Analyzer task.
- **Description**—Displays the description of a test that you select in the **Tests** hierarchy.
- Selected Test's Configuration—Displays the configurable settings for a test you select in the Tests hierarchy. If a test has no unique configurable settings, the message No Configuration Necessary appears. Every test includes the Maximum Reported Failures and the Test Ranking configuration settings.
 - Maximum Reported Failures—Specifies the maximum number of failures you want the VI Analyzer to report for each VI this test analyzes. If a test fails more than the specified number of times, the VI Analyzer Results Window includes the notation Maximum Failures Reported for that test.
 - Test Ranking—Specifies whether the results of the current test appear in the top, middle, or bottom of the results list.
- Reset This Test Configuration—Resets the current test settings in the Selected Test's Configuration frame with the default settings.
- Apply Current Max Fails to All—Applies the number you specify in Maximum Reported Failures to every test in the

analysis.

• **Reset All Test Configurations**—Resets the configuration settings to the default settings for every test in the VI Analyzer task.

VI Analyzer - Select VIs Dialog Box

Installed With: Base Package



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

Select **Start a new VI Analyzer task** from the <u>VI Analyzer - Select Task</u> dialog box to display this dialog box.

Use this dialog box to specify the VIs you want to analyze.

This dialog box includes the following components:

- Items to Analyze—Lists all VIs, LLBs, and folders to include in the VI Analyzer task.
- Path to Item—Displays the path to an item you select in the Items to Analyze tree.
- Add Top-level Item—Adds a VI, LLB, or folder to the Items to Analyze hierarchy.
- Remove Top-level Item—Removes a top-level VI, LLB, or folder from the Items to Analyze hierarchy. The button is active only if you select a top-level item in the Items to Analyze hierarchy.
- Include/Exclude Item(s)—Specifies whether to include a VI, LLB, or folder in the Items to Analyze hierarchy. The button is active only if you select an item in the Items to Analyze hierarchy.
- Passwords—Launches the Add Password(s) dialog box.

VI Analyzer Results Window

Installed With: Base Package



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

Click the **Analyze** button from the <u>VI Analyzer - Exclude Tests from VIs</u> dialog box or select **Tools»VI Analyzer»Show Results Window** to display this window.

Use this window to view the results of a VI Analyzer analysis. You can load a VI Analyzer results file (.rsl) to view the results you saved from a previous VI Analyzer task.

This window includes the following tabbed pages:

- **Test Results**—Displays the results of the analysis. The page includes the following components:
 - Results List—Lists the results for each test, as well as each occurrence where the test failed. Double-click an occurrence to open the VI in which the test failed and display the object that caused the failure.

Subitems—Lists the number of items that failed in the indicated category.



Note When you view a previously saved results file in the **VI Analyzer Results Window**, the VI Analyzer might not be able to locate failure occurrences when you double-click errors if you modified the VI since you generated the results file.

- Show Failures Only—Indicates whether the Results
 List hierarchy displays only occurrences where the test
 failed or all occurrences where the test failed, passed, or
 was skipped for each VI.
- Sort by VI—Sorts the tests and VIs tested in the Results List hierarchy by the VIs included in the VI Analyzer task.
- Sort by Test—Sorts the tests and VIs tested in the

Results List hierarchy by test.

- Ranking Legend—Displays the icons that denote high-ranking and low-ranking tests. High-ranking tests appear at the top of the list and low-ranking tests appear at the bottom. High-ranking tests involve issues that have a significant impact on VI performance. Low-ranking tests involve minor style or cosmetic issues that do not significantly affect VI performance. Tests that do not have a ranking icon next to them fall into a middle-ranking area.
- Description—Displays details about the item you select in the Results List hierarchy, depending on the item you select. For VIs, the text box contains the path to the VI.
 For tests, the text box contains the test description. For occurrences of a failure, the text box contains a description of the error.
- **Testing Errors**—Displays errors related to the analysis itself, such as VIs or tests that did not load. The page includes the following components:
 - Errors List—Lists the errors that occurred during the analysis. Errors include VIs and tests that did not load, tests that did not run, and tests in which errors occurred.
 - Error Description—Displays details about the item you select in the Errors List listbox, depending on the item you select. For VIs, the text box contains the path to the VI. For tests, the text box contains the test description. For occurrences of an error, the text box contains a description of the error.
 - Export—Launches the Export VI Analyzer Results dialog box.
 - Load—Loads a VI Analyzer results file into the VI Analyzer Results Window.
 - Save—Saves the results to a VI Analyzer results file that you can reload into the VI Analyzer Results Window at a later time.
 - Done—Closes the VI Analyzer Results Window. The window prompts you to save the results to a VI Analyzer

results file and gives the option of returning to the VI Analyzer.

- **Summary**—Displays summary results for the analysis, including the following information:
 - **Tests that Ran**—Provides the total number of tests that ran, including the number that passed and failed.
 - **Errors**—Provides the number of VIs that did not load, tests that did not load, tests that did not run, and tests that returned errors when run.
 - **Timing Statistics**—Provides the total time for the VI Analyzer analysis to run and complete.
 - **Numbers**—Provides the total number of VIs tested.

VI Analyzer Test Descriptions (VI Analyzer Toolkit)



Note The VI Analyzer tests are available only if you install and activate the LabVIEW VI Analyzer Toolkit. Refer to the <u>National</u> <u>Instruments Web site</u> for more information about the VI Analyzer Toolkit.

The <u>VI Analyzer - Select Tests</u> dialog box contains a list of tests you can run. When you click a test name in the **Test** hierarchy, a description of the test appears in the text box below the list. The VI Analyzer organizes the tests into four categories.

Block Diagram tests
Documentation tests
Front Panel tests
General tests

VI Analyzer - Select Task Dialog Box

Installed With: Base Package



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

Select Tools»VI Analyzer»Analyze VIs to display this dialog box.

Use this dialog box to select the type of task you want the VI Analyzer to perform.

This dialog box includes the following options:

- Start a new VI Analyzer task—Opens dialog boxes that allow you to select the VIs you want to analyze and the tests you want to run.
- Load a previously saved analysis configuration file—Opens a file dialog that allows you to select a configuration file (.cfg) to use.
- Analyze the VI you are currently editing—Opens dialog boxes
 to create a VI Analyzer task for the VI from which you launched
 the VI Analyzer. This option is available only if you run the VI
 Analyzer from an open VI that you have saved.
 - VI Name—Displays the name of the VI to be analyzed.

Add Password(s) Dialog Box

Installed With: Base Package

Click the **Passwords** button on the <u>VI Analyzer - Select VIs</u> dialog box to display this dialog box.

Use this dialog box to add passwords to the VI Analyzer task so you can analyze password-protected VIs if the LabVIEW password cache does not include the passwords.

This dialog box includes the following components:

- **Number stored**—Displays the number of passwords that the VI Analyzer password list contains.
- Empty List—Clears the VI Analyzer password list.
- Password to Add—Contains the password you want to add to the VI Analyzer password list.
- Add—Adds the password you specify in the Password to Add text box to the VI Analyzer password list.
- Done—Closes the Add Password(s) dialog box.

Saving and Loading VI Analyzer Results



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

You can save VI Analyzer results into a results file (.rsl) that you can load into the VI Analyzer Results Window.

Saving a Results File

Complete the following steps to save a results file.

- 1. From the **VI Analyzer Results Window**, click the **Save** button. A file dialog box appears.
- 2. Navigate to the location where you want to save the results file. The default is the LabVIEW directory.
- 3. Enter the filename you want for the results file and click the **OK** button.

Loading a Results File

Complete the following steps to load a results file into the VI Analyzer **Results Window.**

- 1. Select Tools»VI Analyzer»Show Results Window. If the VI Analyzer Results Window is already open, click the Load button. A file dialog box appears.
- 2. Navigate to the results file you want to load and select it. Click the **OK** button. The **VI Analyzer Results Window** displays the results you saved previously.



Note When you view a previously saved results file in the VI **Analyzer Results Window**, you might not be able to double-click errors to highlight them accurately in the VI if you modified the VI.

You also can export VI Analyzer results to a file in ASCII or HTML format.

Export VI Analyzer Results Dialog Box

Installed With: Base Package

Click the **Export** button from the <u>VI Analyzer Results Window</u> to display this dialog box.

Use this dialog box to export VI Analyzer results to an HTML file or to a tab-delimited ASCII file.

This dialog box includes the following components:

- Export File Path—Contains the path to the ASCII or HTML file to which you want to save the results. The file can be new or existing. If the file already exists, the dialog box confirms that you want to overwrite the existing file.
- **Export Type**—Specifies whether you want to save the results as an HTML file or as a tab-delimited ASCII file.
- **Sort by**—Specifies whether you want to sort the results by VI or by test.
- **Export Options**—Lists the sections you can include in the report. Place a checkmark in the following checkboxes to select the sections you want to include in the report:
 - Summary Data—Includes numeric summary data from the VI Analyzer task.
 - Failed Tests—Includes details on tests that failed in the VI Analyzer task.
 - Passed Tests—Includes details on tests that passed in the VI Analyzer task.
 - Skipped Tests—Includes details on tests that the VI Analyzer skipped.
 - Errors—Includes details on errors that occurred during the VI Analyzer task.
- **Export**—Exports VI Analyzer results using the settings you selected in the dialog box and closes the dialog box.

Block Diagram Tests (VI Analyzer Toolkit)



Note The VI Analyzer tests are available only if you install and activate the LabVIEW VI Analyzer Toolkit. Refer to the <u>National Instruments Web site</u> for more information about the VI Analyzer Toolkit.

Tests in the Block Diagram category of <u>VI Analyzer tests</u> analyze VI performance and style related to the block diagram. The VI Analyzer groups the Block Diagram tests into three subcategories: Performance, Style, and Warnings.

Performance

The tests in the following list analyze coding conventions that affect VI performance.

- Arrays and Strings in Loops—Checks loops to see if they
 contain <u>Build Array</u> or <u>Concatenate Strings</u> functions. Avoid using
 these functions in loops because each call to them requires a
 dynamic resizing of the array or string, which can affect memory
 and processor time. This test does not check timed loops.
- **Coercion Dots**—Checks the total number of coercion dots on the block diagram and the number of coercion dots on individual wires and compares them to user-specified limits.
- **Enabled Debugging**—Checks whether debugging is enabled or disabled. Disabling debugging improves VI performance.
- Wait in While Loop—Checks While Loops with front panel control terminals for structures or functions other than I/O functions that regulate the speed of the While Loop.
- Wired Terminals in Subdiagrams—Checks to see if any control or indicator that is wired on the connector pane does not reside within the top-level diagram. In order to avoid unnecessary memory copies, place control and indicator terminals that are wired on the connector pane should be placed on the top-level diagram.

Style

The tests in the following list analyze block diagrams for issues related to LabVIEW style. Refer to the <u>LabVIEW Style Checklist</u> for information about LabVIEW style.

- Backwards Wires—Checks whether wires flow from left to right.
- **Control Terminal Label Visible**—Checks if labels of block diagram control terminals are visible. To easily identify control terminals on the block diagram, always make labels visible.
- **Control Terminal Wiring**—Checks whether wires exit control terminals from the right side and enter indicator terminals on the left side.
- **Sequence Structure Usage**—Checks whether the block diagram includes <u>Stacked Sequence</u> structures that contain more than the user-specified maximum number of frames. The test does not check <u>Flat Sequence</u> structures.
- String Constant Style—Checks the style of string constants on the block diagram. String constants containing no characters fail the test. Replace them with an empty string constant. String constants set to Normal Display that contain only white space—such as spaces, tabs, or line feeds—fail the test. Set the constants to '\' Codes Display to improve block diagram readability.
- **Unused Code**—Checks for unnecessary code on the block diagram.
- Wire Bends—Compares the total number of bends on a wire to a user-specified maximum and compares the length of each wire segment to a user-specified minimum.
- Wires Under Objects—Checks for wires that run under objects or other wires.

Warnings

The tests in the following list analyze block diagrams for potential design problems.

- Adding Array Size Elements—Checks whether the <u>Add Array Elements</u> function connects to the output of the <u>Array Size</u> function to determine the size of a multidimensional array. Wire the <u>Multiply Array Elements</u> function to the <u>size(s)</u> output of the Array Size function to determine whether a multidimensional array is empty.
- **Breakpoint Detection**—Checks for breakpoints on the block diagram, including breakpoints on wires, nodes, and subdiagrams.
- Bundling Duplicate Names—Checks element names in the <u>Bundle By Name</u> and <u>Unbundle By Name</u> functions for duplicates. Duplicate elements can cause confusing and sometimes incorrect block diagram behavior.
- Error Cluster Wired—Checks that the error output on a block diagram node is wired. You can set the test to ignore VIs with automatic error handling disabled. You also can ignore nodes that usually have unwired error outputs.
- For Loop Iteration Count—Checks For Loops to ensure that the VI does not use both auto-indexing arrays and the N terminal to govern the number of iterations the For Loop runs. You also can check for multiple auto-indexing arrays governing the iteration count.
- Globals and Locals—Checks whether a block diagram contains global and local variables.
- **Hidden Objects in Structures**—Checks whether any objects in structures are hidden outside the visible bounds of the structure.
- Hidden Tunnels—Checks <u>tunnels</u>, <u>shift registers</u>, and other structure border elements to see if they overlap each other. Overlapping tunnels can make your block diagram difficult to read. The test currently does not check dynamic event tunnels, tunnels on Timed Loops with frames, or tunnels on Timed Sequence Structures.
- Indexer Datatype—Checks functions that index array elements

- and string characters to ensure that signed or unsigned 8-bit or 16-bit integer data types do not index string or array elements. The test ignores functions that use constants for indexing.
- Pattern Label—Checks whether a file dialog box that uses file patterns specifies a pattern label. The items you can check are the <u>File Dialog Express VI</u> and the <u>browse options on a path</u> control.
- Reentrant VI Issues—Checks for uninitialized shift registers and non-reentrant subVIs in a reentrant VI. Because reentrant VIs maintain multiple data spaces, unexpected results can occur when using uninitialized shift registers. Additionally, parallelism could be inhibited if the reentrant VI calls non-reentrant subVIs.
- Typedef Cluster Constants—Checks cluster constants on the block diagram that are linked to type definition controls to determine whether their values match the default value of the type definition control. If a constant has a non-default value and the structure of the type definition changes, the value of the constant might reset. Use the Bundle By Name function to change the value of any elements inside a cluster constant linked to a type definition control.

Documentation Tests (VI Analyzer Toolkit)



Note The VI Analyzer tests are available only if you install and activate the LabVIEW VI Analyzer Toolkit. Refer to the <u>National Instruments Web site</u> for more information about the VI Analyzer Toolkit.

Tests in the Documentation category of <u>VI Analyzer tests</u> check for documentation issues within VIs. The VI Analyzer groups the Documentation tests into the Developer and User subcategories.

Developer

The tests in the following list ensure that VIs contain documentation that benefits other developers.

- Comment Usage—Checks whether the block diagram contains a minimum user-specified number of comments. The test also can check whether all subdiagrams of multiframe structures, such as Case, Event, and sequence structures, contain at least one comment.
- Label Call Library Nodes—Checks Call Library Function Nodes
 on the block diagram for labels. You can improve the readability
 of a VI by using the label of a Call Library Function Node to
 describe the function you are calling in a shared library. The test
 fails if a Call Library Function Node has no label or the default
 label.
- **Revision History**—Checks for revision history comments. Clear the revision history when you complete a VI so users cannot see developer comments.

User

The tests in the following list ensure that VIs contain documentation that benefits users.

- **Spell Check**—Spell checks VIs, front panels, and block diagrams.
- VI Documentation—Checks for text in the VI description, control description, and/or tip strip fields on all controls.

Front Panel Tests (VI Analyzer Toolkit)



Note The VI Analyzer tests are available only if you install and activate the LabVIEW VI Analyzer Toolkit. Refer to the <u>National Instruments Web site</u> for more information about the VI Analyzer Toolkit.

Tests in the Front Panel category of <u>VI Analyzer tests</u> analyze front panel design and style. The VI Analyzer groups the Front Panel tests into the SubVI and User Interface subcategories.

SubVI

The tests in the following list check the appearance and arrangement of front panel controls on VIs used as subVIs.

- Array Default Values—Checks charts, graphs, and arrays on a
 front panel for empty default values. Saving non-empty default
 values inside charts, graphs, or arrays uses memory
 unnecessarily. When the VI runs, it overwrites values wired to
 indicators on the block diagram. If a subVI is using the VI, the VI
 overwrites values wired to controls connected to the connector
 pane.
- Cluster Sized to Fit—Checks that front panel clusters are set to Size to Fit, Arrange Horizontally, or Arrange Vertically. If you do not select one of these autosizing options, cluster objects might not be visible to the user.
- **Control Alignment**—Checks that the alignment of controls on the front panel roughly matches the alignment of controls on the connector pane.

User Interface

The tests in the following list analyze user interface design.

- Clipped Text—Checks that any visible text on the front panel is not cut off. This includes text in control labels, control captions, free labels, and text controls such as strings and paths. The test cannot check the text inside listboxes, tables, tree controls, and tab controls.
- **Dialog Controls**—Checks that front panel controls are from the System palette. The test ignores controls that do not have a dialog counterpart.
- **Duplicate Control Labels**—Checks that controls on the front panel do not share the same label.
- Empty List Items—Checks listbox, multicolumn listbox, table, and tree controls to ensure that they are empty. The VI populates the contents of these controls when it runs, so saving a VI with contents in these controls uses memory unnecessarily.
- **Font Usage**—Checks that front panel controls, indicators, and free labels use user-specified symbolic fonts, such as application, system, or dialog fonts.
- Overlapping Controls—Checks that front panel controls do not overlap. The test does not analyze front panel decorations.
- Panel Size and Position—Checks that a front panel completely resides within the bounds of the screen. The test also checks whether the front panel is larger than the maximum specified width and height. If you are using a multi-monitor system, the test fails if the panel does not reside entirely within the bounds of the primary monitor. This test works only on standard, control, and global VIs.
- **Transparent Labels**—Checks that free labels, control labels, and control captions all have transparent backgrounds.

General Tests (VI Analyzer Toolkit)



Note The VI Analyzer tests are available only if you install and activate the LabVIEW VI Analyzer Toolkit. Refer to the <u>National Instruments Web site</u> for more information about the VI Analyzer Toolkit.

Tests in the General category of <u>VI Analyzer tests</u> analyze aspects of VIs not covered in the <u>Block Diagram</u>, <u>Documentation</u>, or <u>Front Panel</u> categories. The VI Analyzer groups the General tests into three subcategories: File Properties, Icon and Connector Pane, and VI Properties.

File Properties

The tests in the following list analyze properties of the VI as a file, such as filename and size.

- **SubVI and TypeDef Locations**—Checks that subVIs and type definitions reside in one of an arbitrary number of user-specified locations (paths and LLBs). If you do not specify a location, the test passes. The test does not check Express VIs on a block diagram.
- VI Extension—Checks the filename extension. The test fails for any VI that does not have a .vi or .vit extension, or for any custom control that does not have a .ctl or .ctt extension. The test is case insensitive.
- VI Name—Checks the name of a VI for potentially invalid characters.
- VI Saved Version—Checks that the VI is saved in the most current version of LabVIEW.
- VI Size—Compares the file size of a VI to the maximum allowable size you specify.

Icon and Connector Pane

The tests in the following list analyze VI icon and connector pane style issues.

- **Connected Pane Terminals**—Checks that control and indicator terminals on the connector pane are wired on the block diagram.
- Connector Pane Alignment—Checks that inputs wire to connector pane terminals on the left side and that outputs wire to connector pane terminals on the right side.
- **Connector Pane Pattern**—Checks that the connector pane pattern, regardless of rotation, matches one of the user-specified connector pane patterns.
- **Default Icon**—Checks that VI icons of standard and polymorphic VIs are neither default nor empty.
- Error Style—Checks that error connections appear in the lower-left and lower-right corners of the connector pane. This part of the test runs only if the connector pane wires a single error in and a single error out terminal. The test also checks whether an error case appears around the contents of the block diagram.
- **Icon Size and Border**—Checks that the icon image is 32 × 32 pixels in size and has a solid border. The test does not check the icons of control or global VIs.
- **Polymorphic Terminals**—Checks that terminals on all instances of a polymorphic VI appear in the same position on the connector pane of the instance VI. The test does not analyze broken polymorphic VIs.
- **Terminal Connection Type**—Checks that controls and indicators on the connector pane that match user-specified names or patterns include the user-specified terminal connection type, such as required, recommended, or optional.
- **Terminal Positions**—Checks that controls and indicators connected to a connector pane that match user-specified names or patterns are located in certain positions.

VI Properties

The tests in the following list analyze the overall setup of a VI.

- Broken VI—Checks for broken VIs.
- Built Application Compatibility—Checks for properties or methods that are not supported in built applications. Run this test on any VIs you want to build into an application to ensure they will function properly.
- **Driver Usage**—Checks for subVIs, functions, or property nodes that are part of National Instruments driver software packages. This test does not check for Express VIs that are part of the driver software packages.
- **Platform Portability**—Checks for potential problems that might occur when you attempt to port a VI from one operating system to another.
- Removed Diagram—Checks whether the block diagram is present. Do not remove a block diagram from a VI because you cannot recover it. You can <u>password protect a VI</u> if you do not want users to view the block diagram.
- **Toolkit Usage**—Checks whether subVIs are National Instrument toolkit VIs. When you distribute a VI that includes toolkit VIs as subVIs, each computer that runs the VI must have the toolkit VIs installed or the VI does not run correctly.
- VI Lock State—Checks the lock state of a VI against the user-specified lock state(s).

Analyzing an Open VI



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

You can analyze a VI that you are currently editing if you have saved the VI. Complete the following steps to analyze an open VI.

- 1. Select Tools»VI Analyzer»Analyze VIs. In the VI Analyzer Select Task dialog box, the VI Analyzer automatically selects the Analyze the VI you are currently editing option.
- 2. Click the **Next** button. The <u>VI Analyzer Select Tests</u> dialog box appears.
- 3. Select the tests and settings you want for the VI Analyzer task.
- 4. Click the **Save** button if you want to <u>save the configuration file</u>.
- 5. Click the **Analyze** button. The **Analyzing VIs** dialog box displays the progress of the VI Analyzer task. After the analysis runs, the <u>VI Analyzer Results Window</u> appears.

Exporting VI Analyzer Results



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

You can export VI Analyzer results in a format you can use with other applications. Complete the following steps to create a report of the results in ASCII or HTML format.

- From the <u>VI Analyzer Results Window</u>, click the **Export** button. The <u>Export VI Analyzer Results</u> dialog box appears.
- 2. Enter the path and filename for the report in the **Export File Path** control. The default is a filename that consists of the current date and time, located in the LabVIEW directory.
- 3. From the **Export Type** list, select the type of file you want. You can export the results either as a tab-delimited ASCII file, which is useful with spreadsheet applications, or as an HTML file, which is useful for viewing in Web browsers.
- 4. From the **Sort by** list, select whether you want the results in the report sorted by VI name or by test name.
- 5. Place checkmarks in the checkboxes next to the results you want to include in the report. By default, the **Summary Data**, **Failed Tests**, and **Errors** checkboxes contain checkmarks.
- 6. Click the **Export** button to generate the file.

You also can <u>save results</u> into a VI Analyzer results file (.rsl) that you can load into the **VI Analyzer Results Window** at a later time.

Saving and Loading Configuration Files



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

You can save the settings you configure for a VI Analyzer task into a configuration file (.cfg) that you can load into the VI Analyzer. The configuration file contains the path to the VIs you want to analyze, the tests you want to run, and the settings you select for specific tests.

You can distribute a configuration file to other users. You can password protect a configuration file that contains settings you do not want other users to see.

Creating a Configuration File

Complete the following steps to create a configuration file.



Tip You can set the test criteria you want and save the configuration file without specifying VIs in the VI Analyzer - Select VIs dialog box. You can use this configuration file to run the same VI Analyzer task on different groups of VIs.

- 1. Create and configure a VI Analyzer task.
- 2. From any dialog box in the VI Analyzer, click the **Save** button. A file dialog box appears.
- 3. Navigate to the location where you want to save the configuration file.
- 4. Enter the filename you want for the configuration file and click the **OK** button. The <u>Password Protect Configuration File?</u> dialog box appears.
- 5. If you want to assign a password to the configuration file, click the **Yes** radio button.
- 6. If you are password protecting the file, enter the password in the **Enter New Password** text box. Re-enter the password in the **Enter New Password Again to Verify** text box.
- 7. Click the **OK** button.

Loading a Configuration File

Complete the following steps to load a configuration file into the VI Analyzer.

- 1. Select **Tools»VI Analyzer»Analyze VIs**. The <u>VI Analyzer Select</u> <u>Task</u> dialog box appears.
- 2. Select the **Load a previously saved analysis configuration file** option and click the **Next** button. A file dialog box appears.
- 3. Navigate to the configuration file you want to load and select it. You are prompted to enter the password if the configuration file is password-protected. The **VI Analyzer Select VIs** dialog box appears with the settings you saved in the configuration file. The settings you saved also appear in the <u>VI Analyzer Select Tests</u> and <u>VI Analyzer Exclude Tests from VIs</u> dialog boxes.

You also can load a configuration file into the **Path to Configuration File** input of the <u>VIAn Easy Analyze</u> VI or the <u>VIAn New Task</u> VI.

Refer to the *Loading VI Analyzer Configuration Files* section of the <u>LabVIEW VI Analyzer Toolkit User Guide</u> for more information about saving and loading configuration files.

Creating and Configuring a VI Analyzer Task



Note The VI Analyzer functionality is available only if you install and activate the LabVIEW VI Analyzer Toolkit or if you install and activate an add-on that contains VI Analyzer tests.

A VI Analyzer task contains information about the tests you want to run and the VIs you want to analyze. Complete the following steps to create a VI Analyzer task.

- 1. From the **Tools** menu, select **VI Analyzer»Analyze VIs**.
- 2. In the VI Analyzer Select Task dialog box, select the Start a new VI Analyzer task option and click the Next button.
- 3. In the <u>VI Analyzer Select VIs</u> dialog box, click the **Add Top-level Item** button. Navigate to the LLB, folder, or VI you want to analyze. To select an LLB, double-click the <u>VI folder icon</u> to add all VIs contained in the LLB to the VI Analyzer task.
- If you selected a folder or LLB to analyze, expand the folders in the Items to Analyze hierarchy. Use the Include/Exclude Item(s) button to specify the exact VIs you want in the VI Analyzer task.
- 5. Click the **Next** button. The <u>VI Analyzer Select Tests</u> dialog box allows you to select the tests you want to run. For each test, you can configure settings to determine specific criteria for passing or failing the test. Expand items in the **Tests** hierarchy to see <u>all available VI Analyzer tests</u>.
- 6. Select the tests and settings you want for the analysis and click the **Next** button. By default, the VI Analyzer selects all tests.
- 7. Change the settings in the <u>VI Analyzer Exclude Tests from VIs</u> dialog box if you want to <u>exclude specific tests from specific VIs</u>.
- 8. <u>Save the configuration file</u> if you want to reuse the settings with future VI Analyzer tasks.
- 9. Click the **Analyze** button. The **Analyzing VIs** dialog box displays the progress of the VI Analyzer task.

Refer to the *Creating and Configuring a VI Analyzer Task* section of the <u>LabVIEW VI Analyzer Toolkit User Guide</u> for more information about VI Analyzer tasks.

Password Protect Configuration File? Dialog Box

Installed With: Base Package

Save a VI Analyzer configuration file (.cfg) and click the **OK** button in the file dialog box to display this dialog box.

Use this dialog box to password-protect a configuration file so that users must enter a password to use the configuration file with the VI Analyzer.

This dialog box includes the following components:

- No—Does not password protect the file.
- **Yes**—Password protects the file.
 - Enter New Password
 —Specifies the password you want to assign to the configuration file. The text box is available only if you select the Yes option.
 - Enter New Password Again to Verify—Verifies the password specified in the Enter New Password text box. The text box is available only if you select the Yes option.