

NI-SWITCH Instrument Driver Quick Reference Guide











Easy Programming for National Instruments Switches





ICON	VI/FUNCTION NAME	DESCRIPTION
Examples		
	<code>niSwitch EasyIO Connect</code>	Performs the standard connect operation on a channel pair. Easy I/O versions of NI-SWITCH operations perform the initialization and destruction of a communication session to the hardware.
	<code>niSwitch EasyIO Disconnect</code>	Performs the standard disconnect operation on a channel pair. Easy I/O versions of NI-SWITCH operations perform the initialization and destruction of a communication session to the hardware.
	<code>niSwitch EasyIO Scanning Acquisition</code>	Takes all the primary settings for configuring a scan list and initiates the scan. Easy I/O versions of the NI-SWITCH operations perform the initialization and destruction of a communication session to the hardware.
	<code>niSwitch EasyIO Reset</code>	Resets the instrument to a known state and sends initialization commands to the instrument. Easy I/O versions of the NI-SWITCH operations perform the initialization and destruction of a communication session to the hardware.

ICON	VI/FUNCTION NAME AND DESCRIPTION†	TYPE	PARAMETER	VALUE TO SET, COMMENTS
Initialize Functions				
	<p><code>niSwitch Initialize</code> (<code>niSwitch_init</code>)</p> <p>Creates a new instrument driver session to the specified device using the interface and address you specify for the Resource Name parameter, sends initialization commands, and returns an instrument handle.</p> <p>If the ID Query parameter is set to True, this function queries the instrument ID and verifies that the ID is valid for this instrument driver. If the Reset parameter is set to True, this function resets the instrument to a known state.</p> <p><i>Note: Only a single session can be open to any unique piece of hardware at a time. Further calls to this function return the original ViSession until the first session is closed.</i></p>	ViRsrc ViBoolean ViBoolean ViSession*	Resource Name ID Query Reset Device vi	PXI::16::INSTR VI_TRUE, VI_FALSE VI_TRUE, VI_FALSE Reference to new session handle
	<p><code>niSwitch Initialize With Options</code> (<code>niSwitch_InitWithOptions</code>)</p> <p>Creates a new instrument driver session to the specified device in the Resource Name parameter, sets the initial state of the session attributes specified in the Option String parameter, sends initialization commands, and returns an instrument handle.</p> <p>If the ID Query parameter is set to True, this function queries the instrument ID and checks that the ID is valid for this instrument driver. If the Reset parameter is set to True, this function resets the instrument to a known state.</p> <p><i>Note: Only a single session can be open to any unique piece of hardware at a time. Further calls to this function return the original ViSession until the first session is closed.</i></p>	ViRsrc ViBoolean ViBoolean ViString ViSession	Resource Name ID Query Reset Device Option String *newVi	PXI::16::INSTR VI_TRUE, VI_FALSE VI_TRUE, VI_FALSE Simulate=0,RangeCheck=1,QueryInstrStatus=1,Cache=1,DriverSetup=PXI-2503 2-Wire Mux Reference to new session handle





† Function names for C, C++, LabWindows/CVI, and Visual Basic are in parentheses.







IVI™, National Instruments™, and ni.com™ are trademarks of National Instruments Corporation. Product and company names mentioned herein are trademarks or trade names of their respective companies.

ICON	VI/FUNCTION NAME AND DESCRIPTION†	TYPE	PARAMETER	VALUE TO SET, COMMENTS
Initialize Functions (Continued)				
	niSwitch Close (niSwitch_close) Closes the instrument I/O session, destroys the instrument driver session and all its attributes, and deallocates any memory resources that the driver was using.	ViSession	vi	Session handle
Application Functions				
	niSwitch Connect And Wait Performs the standard connect operation and has the ability to wait for the switch to settle.	ViSession ViConstString ViConstString ViInt32	vi Channel 1 Channel 2 Maximum Time	Session handle Default = None Default = None Default = 1,000 ms
	niSwitch Disconnect And Wait Performs the standard disconnect operation and has the ability to wait for the switch to settle.	ViSession ViConstString ViConstString ViInt32	vi Channel 1 Channel 2 Maximum Time	Session handle Default = None Default = None Default = 1,000 ms
	niSwitch Scanning Acquisition Takes all the primary settings for configuring a scan list and initiates the scan.	ViSession ViBoolean ViInt32 ViInt32 ViReal64 ViConstString ViInt32	vi Continuous Scan Trigger Input Scan Advanced Output Scan Delay Scan List Scan Mode	Session handle VI_TRUE, VI_FALSE Default = External Default = External Default = 0.00 s Default = None Default = Break Before Make
Routing Functions				
	niSwitch Connect Channels (niSwitch_Connect) Creates a path between channel 1 and channel 2. The driver calculates the shortest path between the two channels. If a path is not available, the function returns an error. Note: The paths are bidirectional. For example, if a path exists between CH1 and CH2, then the path between CH2 and CH1 also exists. Also, this function does not support National Instruments switch modules configured for INDEP mode. To use INDEP mode, use the low-level control function niSwitch Control A Single Switch.	ViSession ViConstString ViConstString	vi Channel1 Channel2	Session handle Default = None Default = None
	niSwitch Disconnect Channels (niSwitch_Disconnect) Destroys the path between two channels that you created with niSwitch Connect Channels or niSwitch Set Path. Note: This function does not support National Instruments switch modules configured for INDEP mode. To use INDEP mode, use the low-level control function niSwitch Control A Single Switch	ViSession ViConstString ViConstString	vi Channel 1 Channel 2	Session handle Default = None Default = None
	niSwitch Disconnect All Channels (niSwitch_DisconnectAll) Disconnects all existing paths. Note: If the switch module is not capable of disconnecting all paths, this function returns the WARN_PATH_REMAINS warning. Refer to the NI-SWITCH User Manual for a list of supported modules.	ViSession	vi	Session handle
	niSwitch Can Connect Channels? (niSwitch_CanConnect) Verifies that the switch module can create a path between the two channels you specify with the Channel 1 and Channel 2 parameters. If the switch module can create a path, this function indicates whether the path is currently available given the existing connections. Note: If the path is not available due to the currently existing connections, but the implicit connection between the two channels already exists, the function returns the WARN_IMPLICIT_CONNECTION_EXISTS warning.	ViSession ViConstString ViConstString ViInt32	vi Channel 1 Channel 2 *Path Capability Reference	Session handle Default = None Default = None Default = None




ICON	VI/FUNCTION NAME AND DESCRIPTION†	TYPE	PARAMETER	VALUE TO SET, COMMENTS
	niSwitch Switch Is Debounced? (niSwitch_IsDebounced) Returns the state of the switch module. It indicates if all the paths that you created have settled. <i>Note: Refer to the NI-SWITCH User Manual for a list of supported modules.</i>	ViSession ViBoolean	vi Is Debounced	Session handle VI_TRUE, VI_FALSE
	niSwitch Wait For Debounce (niSwitch_WaitForDebounce) Returns after all the paths that you created have settled.	ViSession ViInt32	vi Maximum Time	Session handle Default = 1,000 ms
	niSwitch Get Path (niSwitch_GetPath) Returns a string that uniquely identifies the path you create with niSwitch Connect Channels. You can pass this string to niSwitch Set Path to establish the exact same path in the future. <i>Note: This function returns only those paths that you explicitly create by calling niSwitch Connect Channels and niSwitch Set Path. For example, if you connect CH1 and CH3, and then connect CH2 and CH3, the explicit path between CH1 and Ch2 does not exist and this function returns an error.</i>	ViSession ViConstString ViConstString ViInt32 ViChar	vi Channel 1 Channel 2 Buffer Size Path List []	Session handle Default = None Default = None Default = None Default = None
	niSwitch Set Path (niSwitch_SetPath) Connects two channels by establishing the exact path you specify with the Path List parameter. Use this function for applications where repeatability of the path is important, such as in calibrated signal paths. If repeatability is not necessary, use niSwitch Connect Channels. To obtain the exact path for a given connection, use niSwitch Get Path.	ViSession ViConstString	vi Path List	Session handle Default = None





Measurement Functions

	niSwitch Scan (niSwitch_Scan) Takes the scan list provided, programs the switching hardware, and initiates the scan. If Handshaking Initiation is set to Measurement Device, the VI waits until the switches have settled and the switch is waiting for a trigger, then returns. If Handshaking Initiation is set to Switch, the VI returns immediately after initiating the scan.	ViSession ViConstString ViInt16	Instrument Handle Scan List Handshaking Initiation	Session handle Default = None Default = Measurement Device
	niSwitch Initiate Scan (niSwitch_InitiateScan) Initiates a scan using the scan triggers and list that you configured with niSwitch Configure Scan Trigger and niSwitch Configure Scan List. niSwitch Initiate Scan returns immediately upon starting the scan. Once you start the scanning operation, you cannot perform any other operation—other than Abort Scan or Send Software Trigger, as well as the retrieval of attributes. All other VIs return the SCAN_IN_PROGRESS error. To stop the scanning operation, call niSwitch Abort Scan.	ViSession	vi	Session handle
	niSwitch Abort Scan (niSwitch_AbortScan) Aborts a previously initiated scan. You initiate a scan with niSwitch Initiate Scan. <i>Note: If you abort a scan, the device returns to a disconnect all state (equivalent to calling niSwitch Disconnect All). If the instrument is not currently scanning, this function returns the NO_SCAN_IN_PROGRESS error.</i>	ViSession	vi	Session handle
	niSwitch Switch Is Scanning? (niSwitch_IsScanning) Returns the state of the switch module. It indicates if the instrument is currently scanning or idle.	ViSession ViBoolean	vi Is Scanning	Session handle VI_TRUE, VI_FALSE




ICON	VI/FUNCTION NAME AND DESCRIPTION†	TYPE	PARAMETER	VALUE TO SET, COMMENTS
	niSwitch Wait For Scan To Complete (niSwitch_WaitForScanComplete) Waits until the instrument stops scanning. <i>Note: Refer to the NI-SWITCH User Manual for a list of supported modules.</i>	ViSession ViInt32	vi Maximum Time	Session handle Default = 1,000 ms
	niSwitch Send Software Trigger (niSwitch_SendSoftwareTrigger) Sends the software trigger to the switch module controlled by the session.	ViSession	vi	Session handle
	niSwitch Continue From Breakpoint (niSwitch_ContinueFromBreakpoint) Continues the scan from the breakpoint. When the scan has been paused by the breakpoint, niSwitch Continue From Breakpoint causes the scan to continue. To determine whether a breakpoint has occurred, you can check the Is At Breakpoint attribute. For more information on scan list syntax, refer to the <i>NI-SWITCH Software User Manual</i> . <i>Note: Refer to the NI-SWITCH User Manual for a list of supported modules.</i>	ViSession	vi	Session handle
	niSwitch Configure Scan Trigger (niSwitch_ConfigureScanTrigger) Configures the scan triggers for the current scan.	ViSession ViReal64 ViInt32 ViInt32	vi Scan Delay Trigger Input Scan Advanced Output	Session handle Default = 0.00 s Default = External Default = External
	niSwitch Configure Scan List (niSwitch_ConfigureScanList) Configures the switch module for scanning by setting the scan list and indicating the mode to scan in. The scan list itself is comprised of a list of channel connections separated by semicolons. For example, the following scan list would scan the first three channels of a multiplexer: com0->ch0; com0->ch1; com0->ch2; To see the status of the scan, you can call either niSwitch Is Scanning or niSwitch Wait For Scan Complete. Use niSwitch Configure Scan Trigger to configure the scan trigger. Use niSwitch Initiate Scan to start the scan. For more information on scan list syntax, refer to the <i>NI-SWITCH Software User Manual</i> .	ViSession ViConstString ViInt32	vi Scan List Scan Mode	Session handle Default = None Default = Break Before Make
	niSwitch Set Continuous Scan (niSwitch_SetContinuousScan) Tells the driver whether to continuously loop the scan list (True) or to stop scanning after one pass through the scan list (False). <i>Note: If the scan is continuous, you must call niSwitch Abort Scan to halt the scan. Refer to the NI-SWITCH User Manual for a list of supported modules.</i>	ViSession ViBoolean	vi Continuous Scan	Session handle VI_TRUE, VI_FALSE



Utility Functions

	niSwitch Revision Query (niSwitch_revision_query) Returns the revision numbers of the instrument driver and instrument firmware.	ViSession ViChar ViChar	vi Instrument Driver Revision [] Firmware Revision []	Session handle Output Output
	niSwitch Self Test (niSwitch_self_test) Runs the instrument's self-test routine and returns the test result(s).	ViSession ViInt16 ViChar	vi *Self Test Result Self Test Message []	Session handle Output Output
	niSwitch Reset (niSwitch_reset) Resets the instrument to a known state and sends initialization commands to the instrument.	ViSession	vi	Session handle

ICON	VI/FUNCTION NAME AND DESCRIPTION†	TYPE	PARAMETER	VALUE TO SET, COMMENTS
	<p>niSwitch Control A Single Switch (niSwitch_SingleSwitchControl)</p> <p>Opens or closes individual switches; however, the primary switch control VIs are niSwitch Connect Channels and niSwitch Disconnect Channels. Note: If you use niSwitch Control A Single Switch to change the state of an individual switch, the routing and scanning functions do not recognize these changes. Due to state-caching in the driver, you must clear any low-level changes by calling niSwitch Reset or niSwitch Disconnect All Channels. Also, this function will work only if the switch module is configured for INDEP mode.</p>	ViSession ViConstString ViInt16	vi Switch Name Switch Action	Session handle Default = None Default = Switch Open (0)
	<p>niSwitch Query A Single Switch (niSwitch_SingleSwitchQuery)</p> <p>Queries the state of individual switches; however, the primary switch control VIs are niSwitch Connect Channels and niSwitch Disconnect Channels. Note: This function will work only if the switch module is configured for INDEP mode.</p>	ViSession ViConstString ViInt16	vi Switch Name Switch State	Session handle Default = None Default = Switch Open (0)
	<p>niSwitch Write Calibration Data (niSwitch_CalibrationDataWrite)</p> <p>Takes the measurement data, typically in terms of the amplifier offset, and stores the value in the EEPROM. The operation automatically stores today's date along with the data in the EEPROM. Note: Refer to the NI-SWITCH User Manual for a list of supported modules.</p>	ViSession ViConstString ViInt32 ViReal64*	vi Channel Calibration Field Calibration Data	Session handle Default = None Default = Channel Amp Calibration Default = 0.0000
	<p>niSwitch Read Calibration Data (niSwitch_CalibrationDataRead)</p> <p>Retrieves the measurement data, typically in terms of the amplifier offset, that was stored in the EEPROM. Also retrieves the stored calibration date. Note: Refer to the NI-SWITCH User Manual for a list of supported modules.</p>	ViSession ViConstString ViInt32 ViReal64* ViInt32 ViInt32 ViInt32	vi Channel Calibration Field Calibration Data *Year *Month *Day	Session handle Default = None Default = Channel Amp Calibration Output Output Output Output

Error Functions

	<p>niSwitch Error Query (niSwitch_error_query)</p> <p>Reads an error code and a message from the instrument error queue. Note: Refer to the NI-SWITCH User Manual for a list of supported modules.</p>	ViSession ViInt32 ViChar	vi *Error Code Error Message []	Session handle Output Output
	<p>niSwitch Get Error Info (niSwitch_GetErrorInfo)</p> <p>Returns the error information associated with the instrument handle, retrieves, and clears the error information for the session. If you leave the instrument handle unwired, this VI retrieves and clears the error information for the process. The error information includes a primary error, secondary error, and an error elaboration string. Leaving the instrument handle unwired is useful if you do not have a session handle to pass to niSwitch Get Error Info or niSwitch Clear Error Info, which occurs when a call to niSwitch Initialize or niSwitch Initialize With Options fails. You can call niSwitch Error Message to obtain a text description of the primary or secondary error value.</p>	ViSession ViStatus ViStatus ViChar	vi *Primary Error *Secondary Error Error Elaboration [256]	Session handle Output Output Output
	<p>niSwitch Error Handler</p> <p>Takes the error cluster and creates a popup error message if the Message Box input is wired to TRUE and an error exists.</p>	ViSession ViBoolean ViChar []	vi Message Box Error Message	Session handle VI_TRUE, VI_FALSE Output

ICON	VI/FUNCTION NAME AND DESCRIPTION†	TYPE	PARAMETER	VALUE TO SET, COMMENTS
Error Functions (Continued)				
	niSwitch Clear Error Info (niSwitch_ClearErrorInfo) Clears the error information for the instrument handle. If you do not wire the instrument handle, this VI clears the error information only for the current execution thread. The error information includes a primary error code, secondary error code, and an error elaboration string. This VI sets the primary and secondary error codes to VI_SUCCESS (0), and sets the error elaboration string to "".	ViSession	vi	Session handle
	niSwitch Error Message (niSwitch_ErrorMessage) Converts a status code returned by an instrument driver function into a user-readable string.	ViSession ViStatus* ViChar	vi Error Code Error Message [256]	Session handle Output Output

Programming Flow

