Real-Time Watchdog VIs

August 2005, 370713B-01

Use the Real-Time Watchdog VIs to monitor and control networked RT Series devices.

Click the icons for VI descriptions.



The VIs on this palette can return general LabVIEW error codes or specific watchdog error codes.

© 2003–2005 National Instruments Corporation. All rights reserved.

Watchdog Clear

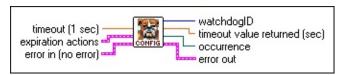
Resets the watchdog hardware and associated attributes and actions to the default power-on state and closes the specified watchdog session.



- Place on the block diagram. Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Configure

Sets the specified timeout value and expiration actions of the watchdog. This VI returns the actual value to which the timeout was set because the timeout value might be coerced for certain watchdog hardware. It also returns an occurrence refnum for the occurrence expiration action.



- ☐ Place on the block diagram. ☐ Find on the **Functions** palette.
- **timeout** is the timeout period for the watchdog timer specified in seconds. The timeout period specifies the amount of time that must elapse before the watchdog timer expires. The timeout period is coerced to a value determined from available timebase and initial count values. The default is 1 s.
- expiration actions specifies which actions to take when the watchdog timer expires. Valid expiration actions are reset, occurrence, trigger line 0, trigger line 1, trigger line 2, trigger line 3, trigger line 4, trigger line 5, trigger line 6, trigger line 7, and SMB line 0. You can use any of these actions in combination.
 - reset reboots the PXI controller.
 - **occurrence** sets the occurrence when the watchdog timer expires.
 - trigger line is the trigger line on the PXI bus to be asserted upon watchdog timer expiration. The following are valid values for trigger line.S

Controller	Supported PXI Triggers	SMB Connectors
8150B Series	4, 5, 6	
8140RT Series	4, 5	8 (front panel "WD TRIG")
8170RT Series	0, 1, 2, 3	_
8176RT	0, 1, 2, 3, 4, 5, 6, 7	8 (front panel "TRG")
FieldPoint 20xx	_	_
7041RT	_	8 (I/O board SMB)

trigger protocol is the protocol used for trigger action on the PXI bus when the watchdog timer expires.

0	Active high
1	Active low
2	Open collector (7041RT only)

- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID identifies the watchdog timer application.
- **timeout value returned** is the timeout period for the watchdog timer specified in seconds. The timeout period specifies the amount of time that must elapse before the watchdog timer expires. The timeout period is coerced to a value determined from available timebase and initial count values. The default is 1.
- **occurrence** is the occurrence associated with the interrupt action upon expiration of the watchdog timer.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.

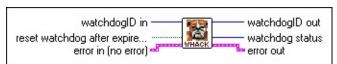
Pabe

source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Whack

Whacks the watchdog. Whacking the watchdog resets the watchdog counter to the specified count. If the specified count equals 0, the watchdog counter is reset to the previously specified count.

Details Examples



- Place on the block diagram. Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- reset watchdog after expire? determines whether a whack after the watchdog timer has expired resets the watchdog timer. The default value is FALSE, which specifies that the watchdog timer remains expired after the first expiration.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- watchdog status is the current status of the watchdog timer. The following are valid values for watchdog status:
 - 0 Disabled
 - 1 Expired
 - 2 Running
- error out contains error information. If error in indicates that an error occurred before this VI or

function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.

- status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Whack Details

The Watchdog Whack VI also starts the watchdog timer if the current status is disabled. You can use the Watchdog Whack VI to restart the watchdog timer after it has expired.

Examples

Refer to the Watchdog Occurrence - RT Engine VI and Watchdog Reset - RT Engine VI in the labview\examples\Real-Time\RT Watchdog (PXI).llb directory for examples of using the Watchdog Whack VI.

- Open Watchdog Occurrence RT Engine VI example
- Open Watchdog Reset RT Engine VI example
- Browse related examples

Advanced Watchdog VIs

Use the Advanced Watchdog VIs for high-level monitoring and control of networked RT Series devices.

Click the icons for VI descriptions.



The VIs and functions on this palette can return general LabVIEW error codes or specific watchdog error codes.

Watchdog Acknowledge

Whacks the watchdog. Whacking the watchdog resets the watchdog counter to the specified count. If the specified count equals 0, the watchdog counter is reset to the previously specified count.



☐ Place on the block diagram. ☐ Find on the **Functions** palette.

watchdogID in identifies the watchdog timer application.

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

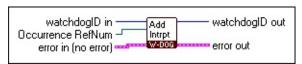
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- watchdog status is the current status of the watchdog timer. The following are valid values for watchdog status:
 - 0 Disabled1 Expired2 Running
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE

(checkmark) to indicate a warning or that no error occurred.

- code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
- **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Add Interrupt Action

Registers the interrupt action invoked when this watchdog expires.



- ☐ Place on the block diagram. ☐ Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- Occurrence RefNum is the occurrence to associate with the interrupt action upon expiration of the watchdog timer.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced

the error or warning.

Watchdog Add Reset Action

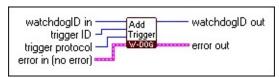
Registers the reset action invoked when this watchdog expires.



- Place on the block diagram. Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Add Trigger Action

Registers the specified trigger action invoked when this watchdog expires.



☐ Place on the block diagram. ☐ Find on the **Functions** palette.

U32

watchdogID in identifies the watchdog timer application.

U32 I

trigger ID is the trigger line on the PXI bus to be asserted upon watchdog timer expiration. The following are valid values for **trigger ID**:

FieldPoint 20xx

3			
Controller	Supported PXI Triggers	SMB Connectors	
8150B Series	4, 5, 6	_	
8140RT Series	4, 5	8 (front panel "WD TRIG")	
8170RT Series	0, 1, 2, 3	_	
8176RT	0, 1, 2, 3, 4, 5, 6, 7	8 (front panel "TRG")	

7041RT	_	8 (I/O
		board
		SMB)

trigger protocol is the protocol used for trigger action on the PXI bus when the watchdog timer expires.

0	Active high
1	Active low
2	Open collector (7041RT only)

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or

function passes the **error in** value to **error out**. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in **error out**. Use the <u>Simple Error Handler</u> or <u>General Error Handler</u> VIs to display the description of the error code. Use **error in** and **error out** to check errors and to specify execution order by wiring **error out** from one node to **error in** of the next node.

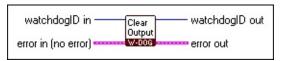
- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in. error out contains error information. If error in indicates that an error occurred before this VI or function ran, error out contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the error out indicator on the front panel and select Explain Error from the shortcut menu for more information about the error.
- **status** is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
- code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning

code.

source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Clear Outputs

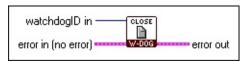
Clears any asserted outputs of the watchdog. Currently, this unasserts any triggers.



- ☐ Place on the block diagram. ☐ Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
 - error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Close

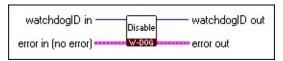
Closes the specified watchdog session.



- Place on the block diagram. Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Disable

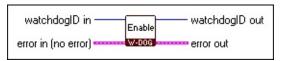
Disables the watchdog hardware. This VI clears any asserted outputs of the watchdog.



- ☐ Place on the block diagram. ☐ Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Enable

Enables the watchdog hardware. This method implicitly whacks the watchdog.



- ☐ Place on the block diagram. ☐ Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Get Attribute

Returns the value for the specified attribute of this watchdog. The value consists of the return value and the unit parameter.



☐ Place on the block diagram. ☐ Find on the **Functions** palette.

0

attribute is the watchdog timer attribute to get. The following are valid attributes:

0	Current count
1	Hardware expire
2	Hardware ID
3	Hardware lock
4	Initial count
5	Instance
6	Resolution
7	Time step
8	Whack input source
9	Total timeout

Attribute 0: Returns the current value of the watchdog counter.



Note This attribute is not supported on PXI Targets.

Attribute 1: Attribute not supported on any targets.

Attribute 2: Returns the Hardware ID# of the watchdog hardware.

Attribute 3: Attribute returns whether the watchdog hardware has been already configured or not.

Attribute 4: Returns value of the initial count (value of the counter before the countdown has begun).

Attribute 5: Returns watchdog instance in the system (useful if you have multiple watchdogs in the system).

Attribute 6: Attribute not supported on any targets.

Attribute 7: Returns rate at which counter decrements.



Note The time step corresponds to the ClockDivisor value set in the watchdog hardware.

Attribute 8: Returns the trigger line configured as the input for the watchdog hardware.

Attribute 9: Returns the total timeout value (TotalTimeout = InitialCount value * TimeStep value). Total timeout value is the amount of time needed to pass before watchdog hardware takes action (performs reset, etc.).

watchdogID in identifies the watchdog timer application.

200

error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.

- status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
- code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
- source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **attribute value returned** is the value of the specified **attribute**. This value is undefined if **attribute** is not supported. Time values are specified in microseconds.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Open

Opens a watchdog session to the first available watchdog in the system. A watchdog cannot have more than one open session at a time.



- Place on the block diagram.
 Find on the Functions palette.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID identifies the watchdog timer application.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Reset

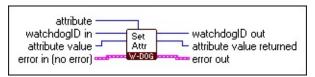
Resets the watchdog hardware and associated attributes and actions to the default power-on state.



- ☐ Place on the block diagram. ☐ Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Set Attribute

Sets the specified attribute of this watchdog to the specified value. This VI returns the actual value to which the specified attribute was set because the value might be coerced for certain watchdog hardware.



■ Place on the block diagram. ■ Find on the **Functions** palette.

attribute is the watchdog timer attribute to set. The following are valid attributes:

0	Current count
1	Hardware expire
2	Hardware ID
3	Hardware lock
4	Initial count
5	Instance
6	Resolution
7	Time step
8	Whack input source
9	Total timeout

Attribute 0: Read-only attribute value.

Attribute 1: Attribute not supported on any targets.

Attribute 2: Read-only attribute value.

Attribute 3: Sets value to indicate watchdog hardware has already been configured.

Attribute 4: Sets value of the initial count (maximum value is 15). This is the value of the counter before the countdown has begun.

Attribute 5: Read-only attribute value.

Attribute 6: Attribute not supported on any targets.

Attribute 7: Sets rate at which counter decrements.



Note The time step corresponds to the ClockDivisor value set in the watchdog hardware.

Attribute 8: Sets the trigger line input source for the watchdog hardware.

Attribute 9: Sets total timeout value (TotalTimeout = InitialCount value * TimeStep value). Total timeout value is the amount of time needed to pass before watchdog hardware takes action

(performs reset, etc.).

- watchdogID in identifies the watchdog timer application.
- **attribute value** is the value to which the specified **attribute** is set. Time values are specified in microseconds.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **attribute value returned** is the value of the specified **attribute**. This value is undefined if **attribute** is not supported. Time values are specified in microseconds.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If **status** is TRUE, **code** is a non-zero <u>error code</u>. If **status** is FALSE, **code** is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Start

Commits the attributes and actions associated with this watchdog to the watchdog hardware.



- ☐ Place on the block diagram. ☐ Find on the **Functions** palette.
- watchdogID in identifies the watchdog timer application.
- error in describes error conditions that occur before this VI or function runs. The default is no error. If an error occurred before this VI or function runs, the VI or function passes the error in value to error out. This VI or function runs normally only if no error occurred before this VI or function runs. If an error occurs while this VI or function runs, it runs normally and sets its own error status in error out. Use the Simple Error Handler or General Error Handler VIs to display the description of the error code. Use error in and error out to check errors and to specify execution order by wiring error out from one node to error in of the next node.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - source describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning. The default is an empty string.
- watchdogID out has the same value as watchdogID in.
- **error out** contains error information. If **error in** indicates that an error occurred before this VI or function ran, **error out** contains the same error information. Otherwise, it describes the error status that this VI or function produces. Right-click the **error out** indicator on the front panel and select **Explain Error** from the shortcut menu for more information about the error.
 - status is TRUE (X) if an error occurred or FALSE (checkmark) to indicate a warning or that no error occurred.
 - code is the error or warning code. If status is TRUE, code is a non-zero error code. If status is FALSE, code is 0 or a warning code.
 - **source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

Watchdog Error Codes

The <u>Watchdog VIs</u> can return the following error codes. Refer to the <u>KnowledgeBase</u> for more information about correcting these errors.

Code	Name	Description
-22015	kNIWDNumberTooBig	Unable to return a value because it would exceed the width of the return data type.
-22014	kNIWDWatchdogIsLocked	This watchdog has had kNIWatchdogHardwareLock enabled. No further changes can be made to its configuration until a reboot.
-22013	kNIWDCannotConnectToKernel	The kernel portion of the watchdog driver is not present, or failed to load correctly. Watchdog hardware cannot be used.
-22012	kNIWDUnsupportedProtocol	This watchdog cannot drive the specified trigger line in the requested manner. Not all watchdog hardware can drive triggers in all possible ways.
-22011	kNIWDHardwareInUse	The watchdog hardware you requested is already in use, or has been permanently locked.
-22010	kNIWD Unsupported Trigger Line	The trigger line specified cannot be driven by this watchdog. Not all watchdog hardware can assert on all, if any, trigger lines.
-22009	kNIWDOutOfMemory	A memory allocation attempted by the watchdog driver failed. Check available memory and disk space.
-22008	kNIWDInvalidSession	The session handle or object pointer for this watchdog is either invalid or illegal.
-22007	kNIWDTooManyActions	The specified action cannot be added to the watchdog. Some actions can only be added once, and others have a maximum number that can be added.
-22006	kNIWDPALFailure	The watchdog driver has experienced an unexpected, low-level error. This watchdog hardware cannot be used.
-22005	kNIWDHardwareNotPresent	The watchdog hardware you requested is either not in the system, or has not been registered with the driver.
-22004	kNIWDUnsupportedParamAction	The action you tried to register cannot be handled by this watchdog hardware. You might be using a newer version of the API with an older implementation, or you might be working with limited hardware.
-22003	kNIWDReadOnlyRegister	The attribute you tried to set is read only. For example, you cannot set IDs and instance numbers.
-22002	kNIWDUnknownAttribute	The attribute you tried to get or set is not recognized as a known attribute by this watchdog hardware. You might be using a newer version of the API with an older implementation.
-22001	kNIWDNotSupported	This method or attribute is not supported by the watchdog hardware on this system. Not all parameters and methods apply to all types of watchdogs.
-22000	kNIWDStatusError	An unexpected error occurred with the watchdog hardware on this system. No detailed information is available for the

error.		
		lerror.