

# NI Scan Engine VIs

June 2008, 372522A-01

**Installed With:** RT Module. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the NI Scan Engine VIs to interface with the NI Scan Engine.

<b>Palette Object</b>	<b>Description</b>
<a href="#">Get Scan Engine Mode</a>	Returns the operating <a href="#">mode</a> of the NI Scan Engine on the target.
<a href="#">Get Scan Engine Period</a>	Returns the <a href="#">scan period</a> of the <a href="#">NI Scan Engine</a> on the target.
<a href="#">Set Scan Engine Mode</a>	Sets the operating <a href="#">mode</a> of the <a href="#">NI Scan Engine</a> on the target.
<a href="#">Set Scan Engine Period</a>	Sets the <a href="#">scan period</a> of the <a href="#">NI Scan Engine</a> on the target.
<a href="#">Synchronize to Scan Engine</a>	Synchronizes execution timing to the <a href="#">scan period</a> of the <a href="#">NI Scan Engine</a> . This VI waits for each scan to complete and triggers subsequent code to execute once the scan engine has updated all values.

<b>Subpalette</b>	<b>Description</b>
<a href="#">Faults VIs</a>	Use the Faults VIs to view, set, and clear NI Scan Engine <a href="#">faults</a> programmatically.
<a href="#">Forcing VIs</a>	Use the Forcing VIs to <a href="#">force and unforce</a> I/O variables programmatically.

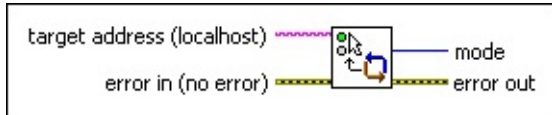
© 2008 National Instruments Corporation. All rights reserved.

# Get Scan Engine Mode VI

Owning Palette: [NI Scan Engine VIs](#)

Installed With: RT Module

Returns the operating **mode** of the NI Scan Engine on the target.



■ 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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.



**source** specifies 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.



**mode**—Returns the operating mode of the NI Scan Engine.

<b>Active Mode</b>	The mode in which the scan engine runs and updates values.
--------------------	--

<b>Configuration Mode</b>	The required mode when configuring scan engine settings.
<b>Fault Mode</b>	The mode triggered when a major or unrecoverable <b>fault</b> occurs.
<b>Initialization Mode</b>	Occurs only briefly during startup.



**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** front panel indicator 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 nonzero **error code**. If **status** is FALSE, **code** is 0 or a warning code.



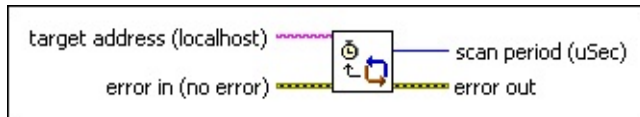
**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

# Get Scan Engine Period VI


Owning Palette: [NI Scan Engine VIs](#)


Installed With: RT Module


Returns the [scan period](#) of the [NI Scan Engine](#) on the target.





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


 **target address** specifies the IP address or DNS name of the target.

 **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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.

 **source** specifies 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.

 **scan period (uSec)** returns the period of the [scan engine](#) running on the target.



**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** front panel indicator 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 nonzero error code. If **status** is FALSE, **code** is 0 or a warning code.



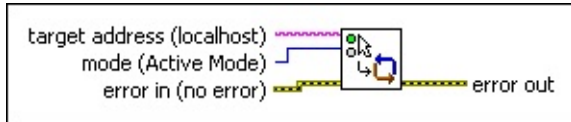
**source** describes the origin of the error or warning and is, in most cases, the name of the VI or function that produced the error or warning.

# Set Scan Engine Mode VI


Owning Palette: [NI Scan Engine VIs](#)


Installed With: RT Module

Sets the operating **mode** of the [NI Scan Engine](#) on the target.





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

 **target address** specifies the IP address or DNS name of the target.


 **mode** specifies the operating **mode** of the [scan engine](#) running on the target.


Active	The mode in which the scan engine runs and updates values.
Configuration	The required mode when configuring scan engine settings.
Fault	The mode triggered when a major or unrecoverable <b>fault</b> occurs.
Initialization	Occurs only briefly during startup.


 **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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.

 **source** specifies 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** front panel indicator 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 nonzero [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.

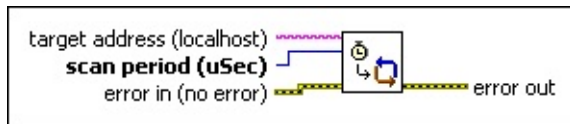
# Set Scan Engine Period VI

Owning Palette: [NI Scan Engine VIs](#)

Installed With: RT Module

Sets the [scan period](#) of the [NI Scan Engine](#) on the target.

[Details](#) [Example](#)



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

**target address** specifies the IP address or DNS name of the target.

**scan period (uSec)** specifies the period for the scan engine running on the target.

**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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.

**source** specifies 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** front panel indicator 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 nonzero **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.

## Set Scan Engine Period Details

Changing the scan period while the scan engine is running can take a significant amount of time, and might cause the scan engine to run late, resulting in a major [fault](#). If changing the scan period results in a fault, you can safely clear the fault after the new scan period takes effect.



**Note** If the target includes an expansion I/O driver, you can set the scan period only when the NI Scan Engine is in Configuration Mode. You can use the [Set Scan Engine Mode](#) VI before this VI to put the NI Scan Engine on the target into Configuration Mode.

## **Example**

Refer to the Scan Engine.lvproj in the labview\examples\scanengine directory for an example of using the Set Scan Engine Period VI.

■ Open example ■ Browse related examples

# Synchronize to Scan Engine VI

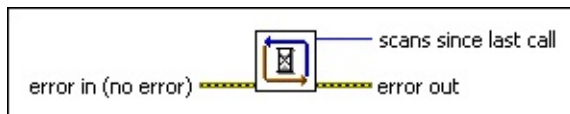
Owning Palette: [NI Scan Engine VIs](#)

Installed With: RT Module


Synchronizes execution timing to the [scan period](#) of the [NI Scan Engine](#). This VI waits for each scan to complete and triggers subsequent code to execute once the scan engine has updated all values.


Use the [Scan Engine](#) page of the [RT Target Properties](#) dialog box to configure the scan period. You also can use the [Set Scan Engine Period](#) VI to set the scan period programmatically.


[Details](#) [Example](#)



■ 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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.


 **source** specifies 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.

 **scans since last call** returns the number of times the scan engine has completed a scan since the last call of this VI. You can use this information to identify missed scan iterations and determine whether the synchronized code consistently finishes executing on time.

 **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** front panel indicator 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 nonzero [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.

## **Synchronize to Scan Engine Details**

Use this VI for code running outside of a [timed structure](#). For most NI Scan Engine use cases, NI recommends using a timed structure with the **Synchronize to Scan Engine** [timing source](#).

This VI must run on a target with the [NI Scan Engine](#) installed.

## **Example**

Refer to the Scan Engine.lvproj in the labview\examples\scanengine directory for an example of using the Synchronize to Scan Engine VI.

■ [Open example](#) ■ [Browse related examples](#)

# Faults VIs

Owning Palette: [NI Scan Engine VIs](#)

**Installed With:** RT Module. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Faults VIs to view, set, and clear NI Scan Engine [faults](#) programmatically.

The VIs on this palette can return [general LabVIEW error codes](#).

Palette Object	Description
<a href="#">Clear All Faults</a>	Clears all active faults on the target.
<a href="#">Clear Fault</a>	Clears NI Scan Engine <a href="#">faults</a> . If the fault you specify using the <b>fault code</b> input is active, this VI clears the fault. If the fault is not active, this VI takes no action and generates no error.
<a href="#">Get Fault List</a>	Returns a list of active <a href="#">faults</a> .
<a href="#">Set Fault</a>	Creates a <a href="#">fault</a> . When the <b>major fault</b> input is TRUE, this VI triggers <a href="#">fault mode</a> . Use the <b>code</b> element of the <b>error in (user fault)</b> input cluster to specify the fault code.



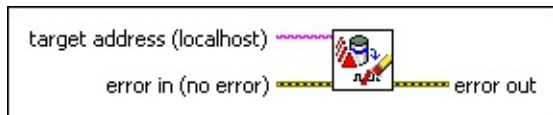
# Clear All Faults VI

Owning Palette: [Faults VIs](#)

Installed With: RT Module

Clears all active faults on the target.

[Details](#) [Example](#)



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

**target address** specifies the IP address or DNS name of the target.

**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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.


**source** specifies 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** front panel indicator 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 nonzero **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.

## Clear All Faults Details

This VI does not allocate memory, so you can use this VI in a time-critical section of an application without significantly affecting determinism.

This VI applies only to targets with the [NI Scan Engine](#) installed. You must either run this VI on a target with the NI Scan Engine installed or use the **target address** input to point to a target with the NI Scan Engine installed.

## Example

Refer to the Scan Engine.lvproj in the labview\examples\scanengine directory for an example of using the Clear All Faults VI.

■ Open example ■ Browse related examples

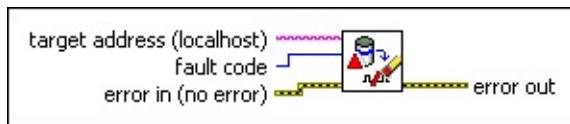
# Clear Fault VI

Owning Palette: [Faults VIs](#)

Installed With: RT Module

Clears NI Scan Engine [faults](#). If the fault you specify using the **fault code** input is active, this VI clears the fault. If the fault is not active, this VI takes no action and generates no error.

[Details](#) [Example](#)



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


**target address** specifies the IP address or DNS name of the target.


**fault code** is the code associated with the fault. The VI ignores this input if **clear all?** is TRUE.


**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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.

 **source** specifies 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** front panel indicator 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 nonzero [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.

## Clear Fault Details

This VI does not allocate memory, so you can use this VI in a time-critical section of an application without significantly affecting determinism.

This VI applies only to targets with the [NI Scan Engine](#) installed. You must either run this VI on a target with the NI Scan Engine installed or use the **target address** input to point to a target with the NI Scan Engine installed.

## Example

Refer to the Scan Engine.lvproj in the labview\examples\scanengine directory for an example of using the Clear Fault VI.

■ Open example ■ Browse related examples



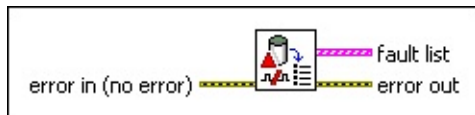
# Get Fault List VI

Owning Palette: [Faults VIs](#)


Installed With: RT Module


Returns a list of active [faults](#).


[Details](#) [Example](#)




■ 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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.


 **source** specifies 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.


 **fault list** contains the list of active faults.


 **code** is the fault code.


 **level** is the severity level of the fault.


 **first reported time** returns the time the fault first occurred since it was last cleared.

 **number reported** is the number of times the fault has occurred since it was last cleared.

 **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** front panel indicator 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 nonzero **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.

## Get Fault List Details

This VI allocates memory and can affect the determinism of the application. This VI must run on a target with the [NI Scan Engine](#) installed.

To obtain the fault list of a remote target from a host computer, you can use the [DataSocket Read](#) VI to subscribe to the following URL:

```
psp:\\<target IP address>\NI_SystemState\ControllerStatus\FaultList
```

The data type returned by the URL above is identical to that of the **fault list** output terminal of this VI.

You also can determine whether a fault is present on a remote target from a host computer by using the DataSocket Read VI to subscribe to the following URL:

```
psp:\\<target IP address>\NI_SystemState\ControllerStatus\FaultPresent
```

The data type returned by the URL above is Boolean.

## Example

Refer to the Scan Engine.lvproj in the labview\examples\scanengine directory for an example of using the Get Fault List VI.

■ Open example ■ Browse related examples

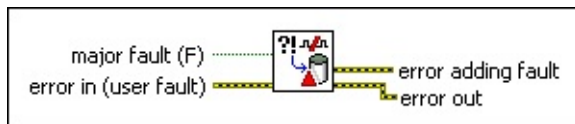
# Set Fault VI

Owning Palette: [Faults VIs](#)


Installed With: RT Module


Creates a [fault](#). When the **major fault** input is TRUE, this VI triggers [fault mode](#). Use the **code** element of the **error in (user fault)** input cluster to specify the fault code.


[Details](#) [Example](#)




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


 **major fault** specifies whether the fault is major (TRUE) or minor (FALSE).

 **error in (user fault)** specifies the fault code. If you do not specify a fault code, the fault code defaults to -66480.

 **status** indicates whether **code** corresponds to an error or a warning. This VI triggers a fault based on the presence of a non-zero error **code**, regardless of the value of **status**.


 **code** is the fault code.

 **source** specifies the source of the fault. This is an optional parameter that you can use to provide information about the fault.

 **error adding fault** contains error information related to the creation of the fault. Fault creation can fail if the NI Scan Engine is not installed on the target or if the target has already reached the maximum number of active faults (100). Right-click the **error out** front panel indicator and select **Explain Error** from the shortcut menu for more information about the error.

 **status** indicates whether an error occurred while this VI ran.

 **code** is the fault code.

 **source** indicates the origin of the error. The default is an empty string.



**error out** returns **error in (user fault)**.

## **Set Fault Details**

This VI does not allocate memory, so you can use this VI in a time-critical section of an application without significantly affecting determinism.

This VI must run on a target with the [NI Scan Engine](#) installed.

## Example

Refer to the Scan Engine.lvproj in the labview\examples\scanengine directory for an example of using the Set Fault VI.

■ Open example ■ Browse related examples



# Forcing VIs

Owning Palette: [NI Scan Engine VIs](#)

**Installed With:** RT Module. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Forcing VIs to [force and unforce](#) I/O variables programmatically.

The VIs on this palette can return [general LabVIEW error codes](#).

Palette Object	Description
<a href="#">Clear Variable Forcing</a>	Unforces all currently <a href="#">forced I/O variables</a> and <a href="#">I/O aliases</a> on the target.
<a href="#">Disable Variable Forcing</a>	Disables <a href="#">forcing</a> for <a href="#">I/O variables</a> and <a href="#">I/O aliases</a> on the target. Unless you <a href="#">clear forcing</a> , the previous forced values return when you re-enable forcing.
<a href="#">Enable Variable Forcing</a>	Enables <a href="#">forcing</a> for <a href="#">I/O variables</a> and <a href="#">I/O aliases</a> on the target.
<a href="#">Force Variable</a>	<a href="#">Forces</a> an <a href="#">I/O variable</a> or <a href="#">I/O alias</a> to assume <b>forced value</b> until you <a href="#">disable forcing</a> , reboot the target, or force the variable to assume a new <b>forced value</b> .
<a href="#">Unforce Variable</a>	Discontinues forcing on an I/O variable or I/O alias and returns control of the I/O value to the <a href="#">NI Scan Engine</a> .

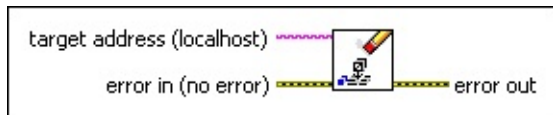
# Clear Variable Forcing VI

Owning Palette: [Forcing VIs](#)


Installed With: RT Module


Unforces all currently [forced I/O variables](#) and [I/O aliases](#) on the target.


[Details](#) [Example](#)





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

 **target address** specifies the IP address or DNS name of the target.

 **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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.


 **source** specifies 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** front panel indicator 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 nonzero **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.

## Clear Variable Forcing Details

This VI does not disable forcing on the target. Use the [Disable Variable Forcing](#) VI to prevent I/O variable and I/O alias forcing from taking effect on the target.

## Example

Refer to the Programmatic Forcing.lvproj in the labview\examples\scanengine directory for an example of using the Clear Variable Forcing VI.

■ Open example ■ Browse related examples

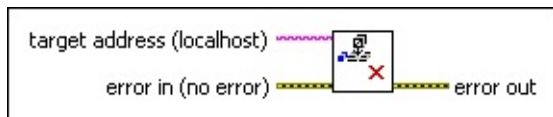
# Disable Variable Forcing VI

Owning Palette: [Forcing VIs](#)


Installed With: RT Module


Disables [forcing](#) for [I/O variables](#) and [I/O aliases](#) on the target. Unless you [clear forcing](#), the previous forced values return when you re-enable forcing.


## Details





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

 **target address** specifies the IP address or DNS name of the target.

 **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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.

 **source** specifies 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** front panel indicator 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 nonzero **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.

## **Disable Variable Forcing Details**

Forcing must be enabled on the target for individual I/O variable or I/O alias forces to take effect. Use the [Enable Variable Forcing](#) VI to re-enable forcing on the target.



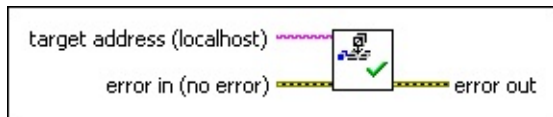
# Enable Variable Forcing VI

Owning Palette: [Forcing VIs](#)

Installed With: RT Module

Enables [forcing](#) for [I/O variables](#) and [I/O aliases](#) on the target.

[Details](#) [Example](#)



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

**target address** specifies the IP address or DNS name of the target.

**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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.


**source** specifies 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** front panel indicator 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 nonzero **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.

## **Enable Variable Forcing Details**

Forcing must be enabled on the target for individual I/O variable or I/O alias forces to take effect. This VI affects the equivalent of a master switch for all I/O variable and I/O alias forcing on the target.

## Example

Refer to the Programmatic Forcing.lvproj in the labview\examples\scanengine directory for an example of using the Enable Variable Forcing VI.

■ Open example ■ Browse related examples

# Force Variable VI

Owning Palette: [Forcing VIs](#)


Installed With: RT Module


**Forces** an [I/O variable](#) or [I/O alias](#) to assume **forced value** until you disable forcing, reboot the target, or force the variable to assume a new **forced value**.


[Details](#) [Example](#)




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


 **variable path (URL)** specifies the path to the I/O variable or I/O alias. The path can be absolute or relative. The format of the variable path is \\<TargetName>\<ModuleName>\<VariableName>. You can view the variable path for an I/O variable in the **Context Help** window when you hover over the I/O variable node on the block diagram.


 **forced value** is the value for the I/O variable or I/O alias to assume while forcing is enabled on the target and the **enable forcing** input of this VI is TRUE. To ensure data integrity, you must wire a value of the same data type as the I/O variable. To verify the data type of an I/O variable, right-click the I/O variable in the **Project Explorer** window and select **Properties** to display the [Shared Variable Properties](#) dialog box.


 **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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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 nonzero **error code**. If **status** is FALSE, **code** is 0 or a warning code.

 **source** specifies 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** front panel indicator 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 nonzero **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.

## Force Variable Details

To perform batch forcing on a set of I/O variables, [disable forcing](#) on the target, force each I/O variable, then [enable forcing](#) on the target.

## Example

Refer to the Programmatic Forcing.lvproj in the labview\examples\scanengine directory for an example of using the Force Variable VI.

■ Open example ■ Browse related examples



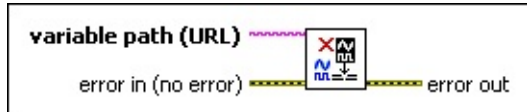
# Unforce Variable VI

Owning Palette: [Forcing VIs](#)

Installed With: RT Module

Discontinues forcing on an I/O variable or I/O alias and returns control of the I/O value to the [NI Scan Engine](#).

## Example




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


**abc** **variable path (URL)** specifies the path to the I/O variable or I/O alias. The path can be absolute or relative. The format of the variable path is \\<TargetName>\<ModuleName>\<VariableName>. You can view the variable path for an I/O variable in the **Context Help** window when you hover over the I/O variable node on the block diagram.


**TF** **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 [exception control](#) to treat what is normally an error as no error or to treat a warning as an error. 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.


**TF** **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.


**IS2** **code** is the error or warning code. The default is 0. If **status** is TRUE, **code** is a nonzero [error code](#). If **status** is FALSE, **code** is 0 or a warning code.

 **source** specifies 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** front panel indicator 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 nonzero [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.

## **Example**

Refer to the Programmatic Forcing.lvproj in the labview\examples\scanengine directory for an example of using the Unforce Variable VI.

■ [Open example](#) ■ [Browse related examples](#)