NI-Modinst Help

May 2008, 371474C-01

The NI-ModInst API provides a way to programmatically query for devices installed in your system that are supported by a specific instrument driver. This help file provides references for both <u>LabVIEW VIs</u> and C functions.

To navigate this help file, use the **Contents**, **Index**, and **Search** tabs to the left of this window.

To comment on National Instruments documentation, refer to the <u>National Instruments Web site</u>.

© 2005–2008 National Instruments Corporation. All rights reserved.

NI-ModInst Overview

NI-ModInst is a component that you can use to query information about National Instruments Modular Instruments installed in your system. NI-ModInst provides a programmatic way to determine the number, names, models, and other attributes of devices in your system that are supported by a specified driver. You can use the NI-ModInst API in applications you develop using LabVIEW, LabWindows™/CVI™, Measurement Studio, or C.

In general, you provide the name of a National Instruments instrument driver to NI-ModInst. Examples of instrument drivers include NI-SCOPE, NI-FGEN, and NI-DMM. NI-ModInst searches the system and finds all the devices that are supported by that driver. You can query NI-ModInst for the number of installed devices, the device names, product models, serial numbers, chassis numbers, slot numbers, bus numbers, and socket numbers.

Installed Files

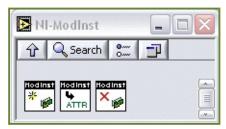
The following table lists files you may need when using NI-ModInst. These files are installed to the default locations by the NI-ModInst installer.

Application Development Environment	File	Location
LabVIEW	niModInst Open Installed Devices	NI-ModInst VI palette:
	Session.vi, niModInst Get Installed Device Attribute (poly).vi, niModInst Close Installed Devices Session.vi	Functions»Instrument I/O»Instrument Drivers»NI-ModInst
	niModInst EX Find Installed Devices.vi (Example Program)	<labview>\examples\instr\niModInst</labview>
LabWindows/CVI	niModInst.fp	<ivi directory="">\Drivers\niModInst\</ivi>
	niModInst.h	<ivi directory="">\Include\</ivi>
C/C++ ADEs	niModInst.h	<ivi directory="">\Include\</ivi>
	niModInst.lib	<ivi directory="">\Lib\msc\</ivi>

NI-ModInst VIs

Use the VIs located on the **NI-ModInst** palette to query information about National Instruments Modular Instruments devices installed in your system.

Click the icons for VI and function descriptions.



niModInst Open Installed Devices Session

Creates a handle to a list of installed devices supported by the specified driver. Call this VI and select the name of a National Instruments instrument driver, such as NI-SCOPE, from the **driver** control. This VI searches the system and constructs a list of all the installed devices that are supported by that driver, and returns both a handle and the number of devices found. The handle is used with other functions to query for attributes such as device name and model, and to safely discard the list when finished.



Note This handle reflects the system state when the handle is created (that is, when you call this VI). If you remove devices from the system or rename them in Measurement & Automation Explorer (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using niModInst_Close Installed Devices Session and create a new handle using this VI.



abc

driver specifies the driver whose supported devices you want to find. This string is not case-sensitive. Some examples are:

NI-SCOPE niScope

NI-FGEN

niFgen

NI-HSDIO

niHSDIO

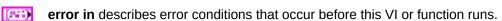
NI-DMM

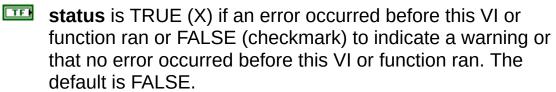
niDMM

NI-SWITCH niSwitch



Note If you do not wire an input or an empty string for this parameter, NI-ModInst creates a list of all Modular Instruments devices installed in the system.





code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.

- source identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **session** is the NI-ModInst session handle, which acts as a handle to the list of installed devices and is used in other NI-ModInst functions.
- **device count** returns the number of devices found in the system that are supported by the driver specified in the **driver** parameter.
- **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.
 - 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 identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niModinst Get Installed Device Attribute (poly)

Returns an attribute (either an integer or a string) specified by the **attribute ID** parameter for a device specified by the **session in** and **index in** parameters.

The **session in** parameter is expected to be a valid handle returned by <u>niModInst Open Installed Devices Session</u>. Therefore, this parameter acts as a handle to a list of installed devices. The **index in** parameter specifies the device in the list for which you want the attribute.

Use the pull-down menu to select an instance of this VI.

Select an instance

String

Returns a string attribute specified by the **attribute ID** parameter for a device specified by the **session in** and **index in** parameters.



- session in is the NI-ModInst session handle created by <u>niModInst Open Installed Devices Session</u>.
- index in is a zero-based index that specifies the device for which you want the attribute.

 The index in parameter should be between 0 and (device count 1), inclusive, where device count is the number of installed devices returned by niModinst Open Installed Devices Session.
- **attribute ID** is the ID of the string attribute you want to query.

Valid Values

Device Name—the name of the device, which can be used to open an instrument driver session for that device

Device Model—the model of the device (for example, NI PXI-5122)

Serial Number—the serial number of the device

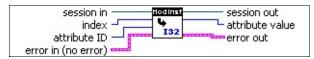
- error in describes error conditions that occur before this VI or function runs.
 - status is TRUE (X) if an error occurred before this VI or function ran or FALSE (checkmark) to indicate a warning or that no error occurred before this VI or function ran. The default is FALSE.
 - code is the error or warning code. The default is 0. If status is TRUE, code is a negative error code. If status is FALSE, code is 0 or a warning code.
 - source identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- session out has the same value as session in.
- index out has the same value as index in.
- **attribute value** returns the value of the requested attribute.
- **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.
 - 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** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

132

132

Returns an integer attribute specified by the **attribute ID** parameter for a device specified by the **session in** and **index in** parameters.



- session in is the NI-ModInst session handle created by niModInst Open Installed Devices Session.
- **index in** is a zero-based index that specifies the device for which you want the attribute. The index in parameter should be between 0 and (device count - 1), inclusive, where device count is the number of installed devices returned by niModinst Open Installed Devices Session.
- I32 attribute ID is the ID of the integer attribute you want to query.

Valid Values

Slot Number—the slot (for example, in a PXI chassis) in which the device is installed. This attribute can only be queried for PXI devices installed in a chassis that has been properly identified in MAX.

Chassis Number—the number of the chassis in which the device is installed. This attribute can only be gueried for PXI devices installed in a chassis that has been properly identified in MAX.

Bus Number—the bus on which the device has been enumerated.

Socket Number—the socket number on which the device has been enumerated.

PCI Express Link Width—the PCI express link width negotiated between the device and the system.

PCI Express Max Link Width—the maximum PCI express link width supported by the device.



Notes

- The bus number and socket number can be used to form a VISA resource string for the device, of the form PXIn:x:INSTR, where *n* is the bus number and *x* is the socket number. For example, for bus number 2 and socket number 13, use PXI2::13::INSTR.
- Traditional NI-DAQ devices do not support the Bus Number and Socket Number attributes.
- error in describes error conditions that occur before this VI or function runs.
 - **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 negative error code. If status is FALSE, code is 0 or a warning code.
- source identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- session out has the same value as session in.
- index out has the same value as index in.
- **attribute value** returns the value of the requested attribute.
- **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.
 - 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** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

niModinst Close Installed Devices Session

Cleans up the NI-ModInst session created by a call to <u>niModInst Open Installed Devices Session</u>. Call this VI when you are finished using the session handle and do not use this handle again.



- **session** is the NI-ModInst session handle, which acts as a handle to the list of installed devices and is used in other NI-ModInst functions.
- error in describes error conditions that occur before this VI or function runs.
 - 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 negative error code. If status is FALSE, code is 0 or a warning code.
 - source identifies where an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.
- **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.
 - 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** identifies where and why an error occurred. The source string includes the name of the VI that produced the error, what inputs are in error, and how to eliminate the error.

NI-ModInst Function Reference

The NI-ModInst API contains the following C/C++ functions:

 $\underline{niModinst_OpenInstalledDevicesSession}$

niModInst_GetInstalledDeviceAttributeViString

niModInst_GetInstalledDeviceAttributeViInt32

niModinst CloseInstalledDevicesSession

<u>niModInst_GetExtendedErrorInfo</u>

niModInst_OpenInstalledDevicesSession

Specific Function

C Function Prototype

ViStatus niModInst_OpenInstalledDevicesSession (ViConstString driver, ViSession* handle, ViInt32* deviceCount);

Description

Creates a handle to a list of installed devices supported by the specified driver. Call this function and pass in the name of a National Instruments instrument driver, such as "NI-SCOPE". This function searches the system and constructs a list of all the installed devices that are supported by that driver, and then returns both a handle to this list and the number of devices found. The handle is used with other functions to query for attributes such as device name and model, and to safely discard the list when finished.



Note This handle reflects the system state when the handle is created (that is, when you call this function). If you remove devices from the system or rename them in Measurement & Automation Explorer (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle using nimedolfostero (MAX), this handle may not refer to an accurate list of devices. You should destroy the handle of the handle o

Parameters

Name	Туре	Description	
driver	ViConstString	A string specifying the driver whose supported devices you want to find. This string is not case-sensitive. Some examples are:	
		NI-SCOPE niScope	
		NI-FGEN niFgen	
		NI-HSDIO niHSDIO	
		NI-DMM niDMM	
		NI-SWITCH niSwitch	
		Note If you use the empty string for this parameter, NI-ModInst creates a list of all Modular Instruments devices installed in the system.	
handle	ViSession*	A pointer to a ViSession variable that receives the value of the NI- ModInst session handle. This value acts as a handle to the list of installed devices and is used in other NI-ModInst functions.	
deviceCount	ViInt32*	A pointer to an integer variable that receives the number of devices found in the system that are supported by the driver specified in the driver parameter.	

Return Value

$niModInst_GetInstalledDeviceAttributeViString$

Specific Function

C Function Prototype

ViStatus niModInst_GetInstalledDeviceAttributeViString (ViSession handle, ViInt32 index, ViInt32 attributeID, ViInt32 attributeValueBufferSize, ViChar attributeValue[]);

Description

Returns a string attribute specified by the **attributeID** parameter for a device specified by the **handle** and **index** parameters. The **handle** parameter is expected to be a valid handle returned by niModInst_OpenInstalledDevicesSession. Therefore, it acts as a handle to a list of installed devices. The **index** parameter specifies for which device in the list you want the attribute.

To find out the length of the device name string before you allocate a buffer for it, simply call this function and pass 0 as the **attributeValueBufferSize** parameter or NULL as the **attributeValue** parameter. When you do this, the function returns the size of the buffer required to hold the attribute value string as its return value. You can then allocate an appropriately sized character buffer and call this function again.

Parameters

Name	Туре	Description
handle	ViSession	The NI-ModInst session handle created by niModInst_OpenInstalledDevicesSession .
index	Vilnt32	A zero-based index that specifies the device for which you want the attribute. This index parameter should be between 0 and (deviceCount – 1), inclusive, where deviceCount is the number of installed devices returned by niModInst_OpenInstalledDevicesSession .
attributeID	ViInt32	The ID of the string attribute you want to query.
		Valid Values
		NIMODINST_ATTR_DEVICE_NAME—the name of the device, which can be used to open an instrument driver session for that device
		NIMODINST_ATTR_DEVICE_MODEL—the model of the device (for example, NI PXI-5122)
		NIMODINST_ATTR_SERIAL_NUMBER—the serial number of the device
attributeValueBufferSize	Vilnt32	The size of the buffer allocated and passed in as the attributeValue parameter. The buffer should be large enough to hold the attribute value string (including a NULL terminating character). Refer to the Description section for information on how to determine the exact buffer size required.
attributeValue	ViChar[]	The character buffer into which the attribute value string is copied.

Return Value

Reports the return status of the function call. To obtain a text description of the status code, call <u>niModInst_GetExtendedErrorInfo</u>.



Note The return value for this function has a second purpose. If the **attributeValueBufferSize** is less than the size required to hold the attribute value string, or if the **attributeValue** parameter is NULL, the return value is the size of the buffer required to hold the attribute value string, including a NULL terminating character.

$niModInst_GetInstalledDeviceAttributeViInt 32$

Specific Function

C Function Prototype

ViStatus niModInst_GetInstalledDeviceAttributeViInt32 (ViSession handle, ViInt32 index, ViInt32 attributeID, ViInt32* attributeValue);

Description

Returns an integer attribute specified by the **attributeID** parameter for a device specified by the **handle** and index parameters. The **handle** parameter is expected to be a valid handle returned by niModInst_OpenInstalledDevicesSession. It therefore acts as a handle to a list of installed devices. The **index** parameter specifies the device in the list for which you want the attribute.

Input Parameters

Name	Туре	Description
handle	ViSession	The NI-ModInst session handle created by niModInst_OpenInstalledDevicesSession .
index	Vilnt32	A zero-based index that specifies the device for which you want the attribute. This index parameter should be between 0 and (deviceCount – 1), inclusive, where deviceCount is the number of installed devices returned by niModInst_OpenInstalledDevicesSession .
attributeID	ViInt32	The ID of the integer attribute you want to query.
		Valid Values
		NIMODINST_ATTR_SLOT_NUMBER—the slot (for example, in a PXI chassis) in which the device is installed. This attribute can only be queried for PXI devices installed in a chassis that has been properly identified in MAX.
		NIMODINST_ATTR_CHASSIS_NUMBER—the number of the chassis in which the device is installed. This attribute can only be queried for PXI devices installed in a chassis that has been properly identified in MAX.
		NIMODINST_ATTR_BUS_NUMBER—the bus on which the device has been enumerated.
		NIMODINST_ATTR_SOCKET_NUMBER—the socket number on which the device has been enumerated.
		NIMODINST_ATTR_PCIEXPRESS_LINK_WIDTH—the PCI express link width negotiated between the device and the system.
		NIMODINST_ATTR_MAX_PCIEXPRESS_LINK_WIDTH—the maximum PCI



Notes

express link width supported by the device.

- The bus number and socket number can be used to form a VISA resource string for this device, of the form "PXI<bus number>::<socket number>::INSTR".
- Traditional NI-DAQ devices do not support the chassis number, bus number, and socket number attributes.

attributeValue ViInt32*

A pointer to a signed 32-bit integer variable that receives the value of the requested attribute.

Return Value

$niModInst_CloseInstalledDevicesSession$

Specific Function

C Function Prototype

 $ViStatus\ niModInst_CloseInstalledDevicesSession\ (ViSession\ handle);$

Description

Closes the NI-ModInst session created by a call to niModInst_OpenInstalledDevicesSession. Call this function when you are finished using the session handle and do not use this handle again.

Parameters

Name Type Description

handle ViSession The NI-ModInst session handle created by

niModInst_OpenInstalledDevicesSession.

Return Value

$niModInst_GetExtendedErrorInfo$

Specific Function

C Function Prototype

ViStatus niModInst_GetExtendedErrorInfo (ViInt32 errorInfoBufferSize, ViChar errorInfo[]);

Description

Returns detailed information about the last error that occurred in the current thread during a call to one of the NI-ModInst functions. When one of the other functions returns a negative value as its return value, immediately call this function to get detailed information about the error. Because error information is stored on a thread-by-thread basis, be sure to call this function in the same thread that called the function that returned an error.

The extended error information is returned as a string. To find out the length of the error information string before you allocate a buffer for it, call this function and pass 0 as the **errorInfoBufferSize** parameter or NULL as the **errorInfo** parameter. When you do this, the function returns the size of the buffer required to hold the error information string as its return value. You can then allocate an appropriately sized string character buffer and call this function again.

Parameters

Name Type Description

errorInfoBufferSize ViInt32 The size of the buffer allocated and passed in as the errorInfo

parameter. The buffer should be large enough to hold the **errorInfo** string (including a NULL terminating character). Refer to the *Description* section to find out how to determine the exact buffer size required.

errorInfo ViChar[] The character buffer into which the error information string is copied.

Return Value

Return Value

Reports the return status of the function call. To obtain a text description of the status code, call <u>niModInst_GetExtendedErrorInfo</u>.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Value	Warning
Negative Value	Error