



Revision A  
August 2004  
Copyright© 2004  
by Argantix Corporation  
All rights reserved  
P/N 5005-982

---

**KDC Series  
DC Power Supply  
IVI Driver Manual**

TEL: +1 (858) 404 6936  
FAX: +1 (858) 677-0940  
Email: [sales@argantix.com](mailto:sales@argantix.com)  
Web Site: <http://www.argantix.com>

---

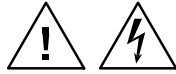
IVI Driver Manual  
P/N 5005-982  
DC Power Supplies  
Argantix

KDC Series

© Copyright 2003-2004, California Instruments, Rev A

Argantix™ is a registered trademark of California Instruments Corporation.

*Specifications subject to change without notice.*



## SAFETY SUMMARY

**This power supply contains high voltage and current circuits that are potentially lethal. Because of its size and weight, electrical and mechanical stability must be ensured. The following safety guidelines must be followed when operating or servicing this equipment. These guidelines are not a substitute for vigilance and common sense. If this power supply is not used as specified in this manual, the protection provided by this equipment may be impaired. California Instruments assumes no liability for the customer's failure to comply with these requirements.**

### BEFORE APPLYING POWER

1. Check the units input rating on the type label and verify the correct voltage is applied to the unit (for example 208 V L-L, 3 Phase).
2. The chassis and cabinet of this power supply must be grounded to minimize shock hazard. A chassis ground is provided at the input terminal block. This is located at the back of the cabinet on the lower right hand side (looking at the back of the unit). The chassis ground must be connected to an electrical ground through an insulated wire (green/yellow) of sufficient gauge.
3. The KDS Series power supplies do not have internal AC mains disconnect. Instead, protection is provided by fuses. The on/off button only engages or disengages the bias supplies, it does not disconnect the AC mains to the unit. **It is strongly recommended to use a suitably rated circuit breaker or mains disconnect device with branch protection rating per local electrical codes between point of connection and the KDS unit AC input to enable complete disconnection of AC input power.**

### FUSES

Use only fuses of the specified current, voltage, and protection speed.

### DO NOT OPERATE IN A VOLATILE ATMOSPHERE

Do not operate the power supply in the presence of flammable gases or fumes. This product is designed to operate in a controlled environment. Do not expose to rain or snow.

### DO NOT TOUCH ENERGIZED CIRCUITS

Disconnect the power cable before servicing this equipment. Even with the power cable disconnected, high voltage can still exist on some circuits. Discharge these voltages before servicing. Only qualified service personnel may remove covers, replace components or make adjustments.

### DO NOT SERVICE ALONE

Do not remove covers, replace components, or make adjustments unless another person, who can administer first aid, is present.

### DO NOT EXCEED INPUT RATINGS

Do not exceed the rated input voltage or frequency. Additional hazards may be introduced because of component failure or improper operation.

### DO NOT MODIFY INSTRUMENT OR SUBSTITUTE PARTS

Do not modify this instrument or substitute parts. Additional hazards may be introduced because of component failure or improper operation.

### MOVING THE POWER SUPPLY

When moving the power supply, observe the following:

1. Remove all AC power to unit.
2. Use two people to prevent injury.

### SURFACE STABILITY

Operate the power supply only on a level surface.

## Disclaimer

### DISCLAIMER OF WARRANTY

=====

THIS INSTRUMENT DRIVER, DOCUMENTATION AND ANY ACCOMPANYING FILES ARE PROVIDED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OR MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. Because of the various hardware and software environments into which this instrument driver may be put, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. The user must assume the entire risk of using the program.

### LIMITATION OF LIABILITY

=====

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL ARGANTIX CORP. OR ITS REPRESENTATIVES BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE PRODUCT OR THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF ARGANTIX CORP. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, ARGANTIX CORP. ASSUMES NO LIABILITY UNDER ANY PROVISION OF THIS AGREEMENT FOR ANY DAMAGES. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

## Table of Contents

Disclaimer.....	2
Introduction:.....	7
Assumptions:.....	7
Error and Status Information:.....	8
How To Use This Document:.....	8
axKDC_Abort .....	11
axKDC_CheckAttributeViBoolean.....	13
axKDC_CheckAttributeViInt32.....	16
axKDC_CheckAttributeViReal64.....	19
axKDC_CheckAttributeViSession .....	22
axKDC_CheckAttributeViString.....	25
axKDC_ClearError .....	28
axKDC_ClearInterchangeWarnings.....	30
axKDC_close.....	32
axKDC_ConfigureCurrentLimit.....	34
axKDC_ConfigureOutputEnabled .....	36
axKDC_ConfigureOutputMode .....	38
axKDC_ConfigureOVP.....	41
axKDC_ConfigureTriggerDelay.....	43
axKDC_ConfigureTriggeredCurrentLimit .....	45
axKDC_ConfigureTriggeredVoltageLevel.....	47
axKDC_ConfigureTriggeredVoltageMode.....	49
axKDC_ConfigureTriggerList .....	51
axKDC_ConfigureTriggerTTLOutputList.....	55
axKDC_ConfigureTriggerVoltageList.....	57
axKDC_ConfigureTrigSlewVoltageList .....	59
axKDC_ConfigureTrigSlewVoltageMode.....	61
axKDC_ConfigureTTLTriggerOutput.....	63
axKDC_ConfigureVoltageLevel .....	65
axKDC_Disable .....	67
axKDC_error_message.....	69
axKDC_error_query .....	75
axKDC_Fetch.....	77
axKDC_GetAttributeViBoolean .....	79
axKDC_GetAttributeViInt32.....	82
axKDC_GetAttributeViReal64.....	85

axKDC_GetAttributeViSession.....	88
axKDC_GetAttributeViString .....	91
axKDC_GetError .....	95
axKDC_GetNextCoercionRecord.....	98
axKDC_GetNextInterchangeWarning .....	101
axKDC_init .....	104
axKDC_Initiate .....	108
axKDC_InitWithOptions.....	110
axKDC_InvalidateAllAttributes .....	115
axKDC_LockSession.....	117
axKDC_Measure .....	120
axKDC_QueryMaxVoltageLevel .....	124
axKDC_QueryOutputState .....	126
axKDC_QueryTrnsListStatus .....	129
axKDC_QuestionableStatusRegister .....	131
axKDC_ReadInstrData.....	133
axKDC_reset .....	135
axKDC_ResetInterchangeCheck .....	137
AXKDC_RESETOUTPUTPROTECTION .....	139
axKDC_ResetPeakCurrent .....	141
axKDC_ResetWithDefaults .....	143
axKDC_revision_query.....	145
axKDC_self_test.....	147
axKDC_SendSoftwareTrigger.....	149
axKDC_SetAttributeViBoolean.....	151
axKDC_SetAttributeViInt32 .....	154
axKDC_SetAttributeViReal64.....	157
axKDC_SetAttributeViSession .....	160
axKDC_StoreRecallRegister .....	166
axKDC_UnlockSession .....	168
axKDC_WriteInstrData .....	171

---

## Argantix KDC Series Power Supply

---

### Introduction:

This instrument driver provides programming support for Argantix KDC Series Power Supply. It contains functions for opening, configuring, taking measurements from, and closing the instrument.

---

### Assumptions:

To successfully use this module, the following conditions must be met:

For GPIB instrument drivers:

- the instrument is connected to the GPIB.
- the GPIB address supplied to the initialize function must match the GPIB address of the instrument.

For VXI instrument drivers:

- the instrument is installed in the VXI mainframe and you are using one of the following controller options:
  - Embedded controller
  - MXI
  - MXI2
  - GPIB-VXI
- the logical address supplied to the initialize function must match the logical address of the instrument.

For RS-232 instrument drivers:

- the instrument is connected to the RS-232 interface.
  - the COM port, baud rate, parity, and timeout supplied to the initialize function must match the settings of the instrument.
-

## Error and Status Information:

Each function in this instrument driver returns a status code that either indicates success or describes an error or warning condition. Your program should examine the status code from each call to an instrument driver function to determine if an error occurred.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

The description of each instrument driver function lists possible error codes and their meanings

## How To Use This Document:

Use this document as a programming reference manual. It describes each function in the

Argantix KDC Series Power Supply

instrument. The functions appear in alphabetical order, with a description of the function and its C syntax, a description of each parameter, and a list of possible error codes.

## Function Tree Layout:

Class/Panel Name:	Function Name:
Initialize	axKDC_init
Initialize With Options Configuration	axKDC_InitWithOptions
Output	
Configure Output Enabled	axKDC_ConfigureOutputEnabled
Configure Output Mode	axKDC_ConfigureOutputMode
Configure Current Limit	axKDC_ConfigureCurrentLimit
Configure OVP	axKDC_ConfigureOVP
Configure Voltage Level	axKDC_ConfigureVoltageLevel
Triggering	
Configure Trigger Source	axKDC_ConfigureTriggerSource
Configure Triggered Voltage Lev	
axKDC_ConfigureTriggeredVoltageLevel	
Configure Triggered Current Lim	
axKDC_ConfigureTriggeredCurrentLimit	
Configure Trigger Delay	axKDC_ConfigureTriggerDelay
Configure TTL Trigger Outp	axKDC_ConfigureTTLTriggerOutput
Configure Trigger Voltage Mode	
axKDC_ConfigureTriggeredVoltageMode	



Configure Trigger Slew Vol Mode	
axKDC_ConfigureTrigSlewVoltageMode	
Configure Trigger List	axKDC_ConfigureTriggerList
Configure Trigger Voltage List	axKDC_ConfigureTriggerVoltageList
Configure Trigger Slew Vol List	
axKDC_ConfigureTrigSlewVoltageList	
Configure Trigger TTL Outp List	
axKDC_ConfigureTriggerTTLOutputList	
Set/Get/Check Attribute	
Set Attribute	
Set Attribute ViInt32	axKDC_SetAttributeViInt32
Set Attribute ViReal64	axKDC_SetAttributeViReal64
Set Attribute ViString	axKDC_SetAttributeViString
Set Attribute ViBoolean	axKDC_SetAttributeViBoolean
Set Attribute ViSession	axKDC_SetAttributeViSession
Get Attribute	
Get Attribute ViInt32	axKDC_GetAttributeViInt32
Get Attribute ViReal64	axKDC_GetAttributeViReal64
Get Attribute ViString	axKDC_GetAttributeViString
Get Attribute ViBoolean	axKDC_GetAttributeViBoolean
Get Attribute ViSession	axKDC_GetAttributeViSession
Check Attribute	
Check Attribute ViInt32	axKDC_CheckAttributeViInt32
Check Attribute ViReal64	axKDC_CheckAttributeViReal64
Check Attribute ViString	axKDC_CheckAttributeViString
Check Attribute ViBoolean	axKDC_CheckAttributeViBoolean
Check Attribute ViSession	axKDC_CheckAttributeViSession
Action/Status	
Initiate	axKDC_Initiate
Abort	axKDC_Abort
Send Software Trigger	axKDC_SendSoftwareTrigger
Query Output State	axKDC_QueryOutputState
Query Max Current Limit	axKDC_QueryMaxCurrentLimit
Query Max Voltage Level	axKDC_QueryMaxVoltageLevel
Query Transient List Status	axKDC_QueryTrnsListStatus
Questionable Status Register	axKDC_QuestionableStatusRegister
Reset Output Protection	axKDC_ResetOutputProtection
Reset Peak Current	axKDC_ResetPeakCurrent
Store Recall Register	axKDC_StoreRecallRegister
Fetch	axKDC_Fetch
Measure	axKDC_Measure
Utility	
Self-Test	axKDC_self_test
Reset	axKDC_reset
Reset With Defaults	axKDC_ResetWithDefaults
Disable	axKDC_Disable
Revision Query	axKDC_revision_query
Error-Query	axKDC_error_query
Error Message	axKDC_error_message
Invalidate All Attributes	axKDC_InvalidateAllAttributes
Error	
Get Error	axKDC_GetError
Clear Error	axKDC_ClearError
Coercion Info	
Get Next Coercion Record	axKDC_GetNextCoercionRecord
Interchangeability Info	
Get Next Interchange Warning	axKDC_GetNextInterchangeWarning

---

Clear Interchange Warnings	axKDC_ClearInterchangeWarnings
Reset Interchange Check	axKDC_ResetInterchangeCheck
Locking	
Lock Session	axKDC_LockSession
Unlock Session	axKDC_UnlockSession
Instrument I/O	
Write Instrument Data	axKDC_WriteInstrData
Read Instrument Data	axKDC_ReadInstrData
Close	axKDC_close

---

#### Argantix KDC Series Power Supply

This instrument driver provides programming support for the Argantix KDC Series Power Supplies. The driver contains all the functions that IVI and VXIplug&play require. In addition, the driver contains high-level functions that configure the power supply and generate output in a single operation. The driver also contains lower level functions that configure the power supply and initiate the output changes in separate operations.

Note: This driver requires the VISA and IVI libraries.

---

The following functions are in alphabetical order.

---

**axKDC\_Abort**

```
ViStatus axKDC_Abort (ViSession instrumentHandle);
```

## Purpose

This function causes the instrument to no longer wait for a trigger.

## Parameter List

```
instrumentHandle
```

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

## Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	

---

3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_CheckAttributeViBoolean***

```
ViStatus axKDC_CheckAttributeViBoolean (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViAttr attributeID,  
                                       ViBoolean attributeValue);
```

### Purpose

This function checks the validity of a value you specify for a ViBoolean attribute.

### Parameter List

#### instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type            ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to check the attribute value. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type            ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see

all IVI attributes or only the attributes of the ViBoolean type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViBoolean are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViBoolean

Pass the value which you want to verify as a valid value for the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

-----  
No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_CheckAttributeViInt32***

```
ViStatus axKDC_CheckAttributeViInt32 (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViAttr attributeID,  
                                       ViInt32 attributeValue);
```

### Purpose

This function checks the validity of a value you specify for a ViInt32 attribute.

### Parameter List

#### instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type            ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to check the attribute value. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type            ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.



A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViInt32 type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box.

Attributes with data types other than ViInt32 are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViInt32

Pass the value which you want to verify as a valid value for the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

-----  
No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_CheckAttributeViReal64***

```
ViStatus axKDC_CheckAttributeViReal64 (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViAttr attributeID,  
                                       ViReal64 attributeValue);
```

### Purpose

This function checks the validity of a value you specify for a ViReal64 attribute.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type          ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to check the attribute value. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type          ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see

all IVI attributes or only the attributes of the ViReal64 type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViReal64 are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViReal64

Pass the value which you want to verify as a valid value for the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

-----  
No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_CheckAttributeViSession***

```
ViStatus axKDC_CheckAttributeViSession (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViAttr attributeID,  
                                       ViSession attributeValue);
```

### Purpose

This function checks the validity of a value you specify for a ViSession attribute.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type          ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to check the attribute value. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type          ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViSession type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViSession are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type          ViSession

Pass the value which you want to verify as a valid value for the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

-----  
No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---



## ***axKDC\_CheckAttributeViString***

```
ViStatus axKDC_CheckAttributeViString (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViAttr attributeID,  
                                       ViChar _VI_FAR attributeValue[]);
```

### Purpose

This function checks the validity of a value you specify for a ViString attribute.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type          ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to check the attribute value. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type          ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see

all IVI attributes or only the attributes of the ViString type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViString are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type      ViChar[]

Pass the value which you want to verify as a valid value for the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

-----  
No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ClearError***

```
ViStatus axKDC_ClearError (ViSession instrumentHandle);
```

### Purpose

This function clears the error code and error description for the IVI session. If the user specifies a valid IVI session for the `instrument_handle` parameter, this function clears the error information for the session. If the user passes `VI_NULL` for the `Vi` parameter, this function clears the error information for the current execution thread. If the `Vi` parameter is an invalid session, the function does nothing and returns an error.

The function clears the error code by setting it to `VI_SUCCESS`. If the error description string is non-NULL, the function de-allocates the error description string and sets the address to `VI_NULL`.

Maintaining the error information separately for each thread is useful if the user does not have a session handle to pass to the `axKDC_GetError` function, which occurs when a call to `axKDC_Init` or `axKDC_InitWithOptions` fails.

### Parameter List

`instrumentHandle`

Variable Type	ViSession
---------------	-----------

The `ViSession` handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: None

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
WARNINGS:	
3FFA4001	Current decreased to satisfy voltage requirements.
3FFA4002	Voltage decreased to satisfy current requirements.
<p>This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:</p>	
-----	
Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ClearInterchangeWarnings***

```
ViStatus axKDC_ClearInterchangeWarnings (ViSession instrumentHandle);
```

### Purpose

This function clears the list of current interchange warnings.

### Parameter List

instrumentHandle

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
WARNINGS:	
3FFA4001	Current decreased to satisfy voltage requirements.
3FFA4002	Voltage decreased to satisfy current requirements.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

---

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

**axKDC\_close**

```
ViStatus axKDC_close (ViSession instrumentHandle);
```

## Purpose

This function performs the following operations:

- Closes the instrument I/O session.
- Destroys the instrument driver session and all of its attributes.
- Deallocates any memory resources the driver uses.

## Notes:

- (1) You must unlock the session before calling axKDC\_close.
- (2) After calling axKDC\_close, you cannot use the instrument driver again until you call axKDC\_init or axKDC\_InitWithOptions.

## Parameter List

instrumentHandle

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

## Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
-----	-----
0	Success
Positive Values	Warnings
Negative Values	Errors



This driver defines the following status codes:

Status	Description
--------	-------------

-----  
No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_ConfigureCurrentLimit***

```
ViStatus axKDC_ConfigureCurrentLimit (ViSession instrumentHandle,  
                                     ViChar _VI_FAR channelName[],  
                                     ViInt32 behavior, ViReal64 limit);
```

### Purpose

This function configures the current limit. You specify the output current limit value and the behavior of the power supply when the output current is greater than or equal to that value.

### Parameter List

#### instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

#### channelName

Variable Type            ViChar[]

Pass the name of the channel on which to configure the current limit.

Valid Channel Name:    "1"

#### behavior

Variable Type            ViInt32

Pass the behavior you want the power supply to exhibit when the output current is greater than or equal to the value of the limit parameter. The driver uses this value to set the AXKDC\_ATTR\_CURRENT\_LIMIT\_BEHAVIOR attribute.

Valid Values:

AXKDC\_VAL\_CURRENT\_REGULATE - Regulatory limit

Default Value:

AXKDC\_VAL\_CURRENT\_REGULATE

#### limit

Variable Type            ViReal64

Pass the current limit you want to use. The driver uses this value to set the AXKDC\_ATTR\_CURRENT\_LIMIT attribute.

Units: Amps

Valid Range: (depends on the model type)

Default Value: 1.0 Amps

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ConfigureOutputEnabled***

```
ViStatus axKDC_ConfigureOutputEnabled (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViBoolean enabled);
```

### Purpose

Configures whether the signal that the power supply produces on a channel appears at the output connector.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel you want to enable or disable.

Valid Channel Name: "1"

#### enabled

Variable Type      ViBoolean

Pass whether you want the signal the power supply produces on a channel to appear at the output connector. The driver uses this value to set the AXKDC\_ATTR\_OUTPUT\_ENABLED attribute.

Valid Values:

VI\_TRUE - Enable the output

VI\_FALSE - Disable the output (Default)

Default Value: VI\_FALSE

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ConfigureOutputMode***

```
ViStatus axKDC_ConfigureOutputMode (ViSession instrumentHandle,
                                     ViChar _VI_FAR channelName[],
                                     ViInt32 outputMode,
                                     ViReal64 constantPowerLevel);
```

### Purpose

Configures the power supply's output mode. You specify whether you want to configure the output to be constant voltage, constant current and constant power.

### Notes:

- 1) When configure the output mode to be constant power, a constant power level must be specified.
- 2) When configure the output mode to be constant voltage or constant current, the constant power level is irrelevant.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to specify the output mode.

Valid Channel Name: "1"

#### outputMode

Variable Type      ViInt32

Pass the type of output mode you want to configure.

#### Valid Values:

CONSTANT VOLTAGE - 0  
 CONSTANT CURRENT - 1  
 CONSTANT POWER - 2

Default Value: 1 (CONSTANT CURRENT)

contantPowerLevel

Variable Type            ViReal64

Pass the constant power level in which you want the power supply to operate.

Unit: Watts

Valid Range:    (depends on the model types)

1 - 15,000 Watts

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors

---

BFFA0000 to BFFA1FFF	IVI	Errors
BFFF0000 to BFFFFFFF	VISA	Errors
BFFC0000 to BFFCFFFF	VXIPnP	Driver Errors

---



## ***axKDC\_ConfigureOVP***

```
ViStatus axKDC_ConfigureOVP (ViSession instrumentHandle,  
                             ViChar _VI_FAR channelName[],  
                             ViBoolean enabled, ViReal64 limit);
```

### Purpose

This function configures the power supply's over-voltage protection. You specifies the over-voltage limit and the behavior of the power supply when the output voltage is greater than or equal to that value.

When the enabled parameter is VI\_FALSE, the driver ignores the limit parameter and does not set the AXKDC\_ATTR\_OVP\_LIMIT attribute.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the OVP limit.

Valid Channel Name: "1"

#### enabled

Variable Type      ViBoolean

Pass whether you want to enable or disable the OVP limit. The driver uses this value to set the AXKDC\_ATTR\_OVP\_ENABLED attribute.

Valid Values:

VI\_TRUE - Enable OVP limit (Default)

VI\_FALSE - Disable OVP limit

Default Value: VI\_TRUE

#### limit

Variable Type      ViReal64

Pass the over-voltage protection limit you want to use. The driver uses this value to set the AXKDC\_ATTR\_OVP\_LIMIT attribute.

Units: Volts

Valid Range :

OVP MODE RANGE : 0.000 - 110%  
 OVP MODE SVOLT : 105.0 - 110%  
 OVP MODE ABS : 0.000 - 110%

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Error

## ***axKDC\_ConfigureTriggerDelay***

```
ViStatus axKDC_ConfigureTriggerDelay (ViSession instrumentHandle,  
                                     ViChar _VI_FAR channelName[],  
                                     ViReal64 triggeredDelay_sec);
```

### Purpose

This function configures the trigger delay time the power supply attempts to wait for after it receives a trigger.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the trigger delay.

Valid Channel Name: "1"

#### triggeredDelay\_sec

Variable Type      ViReal64

Pass the delay time you want the power supply to attempt to wait for after it receives a trigger. The driver sets the AXKDC\_ATTR\_TRIGGER\_DELAY attribute to this value.

Units: seconds

Valid Range: 0.0 - 3600.0 (seconds)

Default Value: 0.0 (seconds)

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ConfigureTriggeredCurrentLimit***

```
ViStatus axKDC_ConfigureTriggeredCurrentLimit
        (ViSession instrumentHandle, ViChar _VI_FAR channelName[],
         ViReal64 triggeredLimit);
```

### Purpose

This function configures the current limit the power supply attempts to use after it receives a trigger.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the triggered current limit.

Valid Channel Name: "1"

#### triggeredLimit

Variable Type      ViReal64

Pass the current limit you want the power supply to attempt to use after it receives a trigger. The driver sets the AXKDC\_ATTR\_TRIGGERED\_CURRENT\_LIMIT attribute to this value.

Units: Amps

Valid Range: (depends on the model type)

Default Value: 1.0

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver

function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ConfigureTriggeredVoltageLevel***

```
ViStatus axKDC_ConfigureTriggeredVoltageLevel
    (ViSession instrumentHandle, ViChar _VI_FAR channelName[],
     ViReal64 triggeredLevel);
```

### Purpose

This function configures the DC voltage level the power supply attempts to generate after it receives a trigger.

### Parameter List

#### instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type            ViChar[]

Pass the name of the channel on which to configure the triggered voltage level.

Valid Channel Name: "1"

#### triggeredLevel

Variable Type            ViReal64

Pass the DC voltage level you want the power supply to attempt to generate after it receives a trigger. The driver sets the AXKDC\_ATTR\_TRIGGERED\_VOLTAGE\_LEVEL attribute to this value.

Units: Volts

Valid Range: (depends on the model type)

Default Value: 0.0 Volts

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You

examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors



### ***axKDC\_ConfigureTriggeredVoltageMode***

```

    ViStatus axKDC_ConfigureTriggeredVoltageMode (ViSession instrumentHandle,
                                                ViChar _VI_FAR
channelName[],
                                                ViInt32
triggeredVoltageMode);

```

#### Purpose

This function configures the triggered voltage mode to the following options :

```

AXKDC_VAL_TRIGGER_MODE_FIX      - "FIX"
AXKDC_VAL_TRIGGER_MODE_STEP     - "STEP"
AXKDC_VAL_TRIGGER_MODE_PULSE    - "PULS"
AXKDC_VAL_TRIGGER_MODE_LIST     - "LIST"

```

#### Parameter List

instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the trigger delay.

Valid Channel Name: "1"

triggeredVoltageMode

Variable Type      ViInt32

Pass this parameter to set the triggered voltage mode to following options :

```

AXKDC_VAL_TRIGGER_MODE_FIX      - 0
AXKDC_VAL_TRIGGER_MODE_STEP     - 1
AXKDC_VAL_TRIGGER_MODE_PULSE    - 2
AXKDC_VAL_TRIGGER_MODE_LIST     - 3

```

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ConfigureTriggerList***

```
ViStatus axKDC_ConfigureTriggerList (ViSession instrumentHandle,
                                     ViInt32 repeatCount,
                                     ViInt32 stepMode, ViInt32 listSize,
                                     ViReal64 _VI_FAR dwellTime[],
                                     ViInt32 _VI_FAR repeatTime[]);
```

### Purpose

This function configures how many times the DC power supply sequences through a list before that list is completed, specifies the time interval that each value (point) of a list is to remain in effect and determines if a trigger causes a list to advance only to its next point or to sequence through all of its points.

### Parameter List

#### instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

#### repeatCount

Variable Type            ViInt32

Pass this parameter(how many times) to loop the transient list.

#### stepMode

Variable Type            ViInt32

Pass this parameter to determine the transient list mode for the following options :

```
AXKDC_VAL_TRIGGER_LIST_STEP_ONCE - 0
AXKDC_VAL_TRIGGER_LIST_STEP_AUTO - 1
```

#### listSize

Variable Type            ViInt32

Pass this parameter for the total data point that the transient list will execute.

#### dwellTime

Variable Type            ViReal64[]

Pass this specified dwell time for each data that the transient list will execute.

repeatTime

Variable Type            ViInt32[]

Pass this specified repeat time that each data point will execute in the transient list.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ConfigureTriggerSource***

```
ViStatus axKDC_ConfigureTriggerSource (ViSession instrumentHandle,
                                       ViChar _VI_FAR channelName[],
                                       ViInt32 source);
```

### Purpose

Configures the trigger source the power supply waits for before changing the output signal. When the power supply receives a trigger signal on the source you specify, it changes its current limit and voltage level to the values of the `AXKDC_ATTR_CURRENT_LIMIT` and `AXKDC_ATTR_VOLTAGE_LEVEL` attributes, respectively.

```
{AXKDC_VAL_TRIG_IMMEDIATE, 0, 0, "IMM", 0},
 {AXKDC_VAL_TRIG_EXTERNAL, 0, 0, "EXT", 0},
 {AXKDC_VAL_SOFTWARE_TRIG, 0, 0, "BUS", 0},
```

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the trigger source.

Valid Channel Name: "1"

#### source

Variable Type      ViInt32

Pass the trigger source to you want the power supply to wait for before changing the output signal. The driver sets the `AXKDC_ATTR_TRIGGER_SOURCE` attribute to this value.

Valid Values:

`AXKDC_VAL_TRIG_IMMEDIATE` - The power supply does not wait for a trigger to change the output signal.

`AXKDC_VAL_SOFTWARE_TRIG` - The power supply waits until you call the

axKDC\_SendSoftwareTrigger function.

Default Value: AXKDC\_VAL\_SOFTWARE\_TRIG

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_ConfigureTriggerTTLOutputList***

```

ViStatus axKDC_ConfigureTriggerTTLOutputList (ViSession instrumentHandle,
                                              ViInt32 listSize,
                                              ViBoolean _VI_FAR
transientOutputTTL[]);

```

#### Purpose

This function configures the output TTL list.

#### Parameter List

##### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

##### listSize

Variable Type      ViInt32

Pass this parameter for the total data point that the transient list will execute.

##### transientOutputTTL

Variable Type      ViBoolean[]

Pass the transient output TTLT for each data point in the transient list.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
-------	---------

```

-----
0                Success
Positive Values  Warnings
Negative Values  Errors

```

This driver defines the following status codes:

```

Status   Description
-----

```

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

```

Numeric Range (in Hex)  Status Code Types
-----
3FFA2000 to 3FFA3FFF    IviPower Warnings
3FFA0000 to 3FFA1FFF    IVI      Warnings
3FFF0000 to 3FFFFFFF    VISA     Warnings
3FFC0000 to 3FFCFFFF    VXIPnP  Driver Warnings

BFFA2000 to BFFA3FFF    IviPower Errors
BFFA0000 to BFFA1FFF    IVI      Errors
BFFF0000 to BFFFFFFF    VISA     Errors
BFFC0000 to BFFCFFFF    VXIPnP  Driver Errors

```



## ***axKDC\_ConfigureTriggerVoltageList***

```
ViStatus axKDC_ConfigureTriggerVoltageList (ViSession instrumentHandle,
                                           ViChar _VI_FAR channelName[],
                                           ViInt32 listSize,
                                           ViReal64 _VI_FAR
                                           transientVoltage[]);
```

### Purpose

This function configures the transient voltage list.

```
ViStatus _VI_FUNC axKDC_ConfigureTriggerVoltageList (
    ViSession      vi,
    ViConstString  channelName,
    ViInt32        listSize,
    ViReal64 _VI_FAR voltage[]
)
```

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the trigger delay.

Valid Channel Name: "1"

#### listSize

Variable Type      ViInt32

Pass this parameter for the total data point that the transient list will execute.

#### transientVoltage

Variable Type      ViReal64[]

Pass the transient voltage for each data point in the transient

list.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_ConfigureTrigSlewVoltageList***

```
ViStatus axKDC_ConfigureTrigSlewVoltageList (ViSession instrumentHandle,
                                             ViChar _VI_FAR channelName[],
                                             ViInt32 listSize,
                                             ViReal64 _VI_FAR
transientSlewVoltage[]);
```

#### Purpose

This function configures transient voltage slew rate list.

```
ViStatus _VI_FUNC axKDC_ConfigureTrigSlewVoltageList (
    ViSession      vi,
    ViConstString  channelName,
    ViInt32        listSize,
    ViReal64 _VI_FAR slewVoltage[]
)
```

#### Parameter List

##### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

##### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the trigger delay.

Valid Channel Name: "1"

##### listSize

Variable Type      ViInt32

Pass this parameter for the total data point that the transient list will execute.

##### transientSlewVoltage

Variable Type            ViReal64[]

Pass the transient voltage slew rate for each data point in the transient list.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_ConfigureTrigSlewVoltageMode***

```

ViStatus axKDC_ConfigureTrigSlewVoltageMode (ViSession instrumentHandle,
                                             ViChar _VI_FAR channelName[],
                                             ViInt32
triggeredVoltageSlewMode);

```

#### Purpose

This function configures the triggered voltage slew mode to the following options :

```

AXKDC_VAL_TRIGGER_MODE_FIX      - "FIX"
AXKDC_VAL_TRIGGER_MODE_STEP     - "STEP"
AXKDC_VAL_TRIGGER_MODE_PULSE    - "PULS"
AXKDC_VAL_TRIGGER_MODE_LIST     - "LIST"

```

#### Parameter List

##### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

##### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the trigger delay.

Valid Channel Name: "1"

##### triggeredVoltageSlewMode

Variable Type      ViInt32

Pass this parameter to set the triggered voltage slew mode to the following options "

```

AXKDC_VAL_TRIGGER_MODE_FIX      - 0
AXKDC_VAL_TRIGGER_MODE_STEP     - 1
AXKDC_VAL_TRIGGER_MODE_PULSE    - 2
AXKDC_VAL_TRIGGER_MODE_LIST     - 3

```

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_ConfigureTTLTriggerOutput***

```
ViStatus axKDC_ConfigureTTLTriggerOutput (ViSession instrumentHandle,
                                          ViBoolean state,
                                          ViInt32 source);
```

#### Purpose

This function configures the trigger output ttl to be ON or OFF, and trigger output ttl source to be BOT, EOT, LIST.

#### Parameter List

##### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

##### state

Variable Type          ViBoolean

Set this parameter wheter to output TTLT ON or TTLT OFF.

Default Value : OFF

##### source

Variable Type          ViInt32

Set the trigger source to the following options:

```
AXKDC_VAL_OUTPUT_TRIGGER_SOURCE_BOT - "BOT"
AXKDC_VAL_OUTPUT_TRIGGER_SOURCE_EOT, - "EOT"
AXKDC_VAL_OUTPUT_TRIGGER_SOURCE_LIST, - "LIST"
```

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors



## ***axKDC\_ConfigureVoltageLevel***

```
ViStatus axKDC_ConfigureVoltageLevel (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViReal64 level);
```

### Purpose

This function configures the DC voltage level the power supply attempts to generate.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to configure the voltage level.

Valid Channel Name: "1"

#### level

Variable Type      ViReal64

Pass the DC voltage you want the power supply to attempt to generate. The driver uses this value to set the AXKDC\_ATTR\_VOLTAGE\_LEVEL attribute.

Units: Volts

Valid Range:      (depends on the model type)

Default Value: 0.0 (Volts)

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_Disable***

```
ViStatus axKDC_Disable (ViSession instrumentHandle);
```

### Purpose

This function places the instrument in a quiescent state where it has minimal or no impact on the system to which it is connected.

### Parameter List

instrumentHandle

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
WARNINGS:	
3FFA4001	Current decreased to satisfy voltage requirements.
3FFA4002	Voltage decreased to satisfy current requirements.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

---

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

**axKDC\_error\_message**

```
ViStatus axKDC_error_message (ViSession instrumentHandle,
                             ViStatus errorCode,
                             ViChar _VI_FAR errorMessage[]);
```

## Purpose

This function converts a status code returned by an instrument driver function into a user-readable string.

## Parameter List

## instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

You can pass VI\_NULL for this parameter. This is useful when one of the initialize functions fail.

Default Value: VI\_NULL

## errorCode

Variable Type            ViStatus

Pass the Status parameter that is returned from any of the instrument driver functions.

Default Value: 0 (VI\_SUCCESS)

## axKDC Status Codes:

Status	Description
-----	
No defined status codes.	

No defined status codes.

## IviPower Status Codes:

Status	Description
-----	
ERRORS:	
BFFA1001	The trigger source is not software trigger.

## ERRORS:

BFFA1001 The trigger source is not software trigger.

## IVI Engine Status Codes:

Status	Description
-----	
ERRORS:	
BFFA0001	Instrument error. Call axKDC_error_query.
BFFA0002	Cannot open file.
BFFA0003	Error reading from file.
BFFA0004	Error writing to file.
BFFA0005	Driver module file not found.
BFFA0006	Cannot open driver module file for reading.

## ERRORS:

BFFA0001 Instrument error. Call axKDC\_error\_query.

BFFA0002 Cannot open file.

BFFA0003 Error reading from file.

BFFA0004 Error writing to file.

BFFA0005 Driver module file not found.

BFFA0006 Cannot open driver module file for reading.

BFFFA0007 Driver module has invalid file format or invalid data.  
BFFFA0008 Driver module contains undefined references.  
BFFFA0009 Cannot find function in driver module.  
BFFFA000A Failure loading driver module.  
BFFFA000B Invalid path name.  
BFFFA000C Invalid attribute.

BFFFA000D IVI attribute is not writable.  
BFFFA000E IVI attribute is not readable.  
BFFFA000F Invalid parameter.  
BFFFA0010 Invalid value.  
BFFFA0011 Function not supported.  
BFFFA0012 Attribute not supported.  
BFFFA0013 Value not supported.  
BFFFA0014 Invalid type.  
BFFFA0015 Types do not match.  
BFFFA0016 Attribute already has a value waiting to be updated.  
BFFFA0017 Specified item already exists.  
BFFFA0018 Not a valid configuration.  
BFFFA0019 Requested item does not exist or value not available.  
BFFFA001A Requested attribute value not known.  
BFFFA001B No range table.  
BFFFA001C Range table is invalid.  
BFFFA001D Object or item is not initialized.  
BFFFA001E Non-interchangeable behavior.  
BFFFA001F No channel table has been built for the session.  
BFFFA0020 Channel name specified is not valid.  
BFFFA0021 Unable to allocate system resource.  
BFFFA0022 Permission to access file was denied.  
BFFFA0023 Too many files are already open.  
BFFFA0024 Unable to create temporary file in target directory.  
BFFFA0025 All temporary filenames already used.  
BFFFA0026 Disk is full.  
BFFFA0027 Cannot find configuration file on disk.  
BFFFA0028 Cannot open configuration file.  
BFFFA0029 Error reading configuration file.  
BFFFA002A Invalid ViInt32 value in configuration file.  
BFFFA002B Invalid ViReal64 value in configuration file.  
BFFFA002C Invalid ViBoolean value in configuration file.  
BFFFA002D Entry missing from configuration file.  
BFFFA002E Initialization failed in driver DLL.  
BFFFA002F Driver module has unresolved external reference.  
BFFFA0030 Cannot find CVI Run-Time Engine.  
BFFFA0031 Cannot open CVI Run-Time Engine.  
BFFFA0032 CVI Run-Time Engine has invalid format.  
BFFFA0033 CVI Run-Time Engine is missing required function(s).  
BFFFA0034 CVI Run-Time Engine initialization failed.  
BFFFA0035 CVI Run-Time Engine has unresolved external reference.  
BFFFA0036 Failure loading CVI Run-Time Engine.  
BFFFA0037 Cannot open DLL for read exports.  
BFFFA0038 DLL file is corrupt.  
BFFFA0039 No DLL export table in DLL.  
BFFFA003A Unknown attribute name in default configuration file.  
BFFFA003B Unknown attribute value in default configuration file.  
BFFFA003C Memory pointer specified is not known.  
BFFFA003D Unable to find any channel strings.  
BFFFA003E Duplicate channel string.

```

BFFA003F Duplicate virtual channel name.
BFFA0040 Missing virtual channel name.
BFFA0041 Bad virtual channel name.
BFFA0042 Unassigned virtual channel name.
BFFA0043 Bad virtual channel assignment.
BFFA0044 Channel name required.
BFFA0045 Channel name not allowed.
BFFA0046 Attribute not valid for channel.
BFFA0047 Attribute must be channel based.
BFFA0048 Channel already excluded.
BFFA0049 Missing option name (nothing before the '=').
BFFA004A Missing option value (nothing after the '=').
BFFA004B Bad option name.
BFFA004C Bad option value.
BFFA004D Operation only valid on a class driver session.
BFFA004E "ivi.ini" filename is reserved.
BFFA004F Duplicate run-time configuration entry.
BFFA0050 Index parameter is one-based.
BFFA0051 Index parameter is too high.
BFFA0052 Attribute is not cacheable.
BFFA0053 You cannot export a ViAddr attribute to the end-user.
BFFA0054 Bad channel string in channel string list.
BFFA0055 Bad prefix name in default configuration file.

```

#### VISA Status Codes:

```
Status Description
```

```
-----
```

#### WARNINGS:

```

3FFF0002 Event enabled for one or more specified mechanisms.
3FFF0003 Event disabled for one or more specified mechanisms.
3FFF0004 Successful, but queue already empty.
3FFF0005 Specified termination character was read.
3FFF0006 Number of bytes transferred equals input count.
3FFF0077 Configuration non-existent or could not be loaded.
3FFF007D Open successful, but the device not responding.
3FFF0080 Wait successful, but more event objects available.
3FFF0082 Specified object reference is uninitialized.
3FFF0084 Attribute value not supported.
3FFF0085 Status code could not be interpreted.
3FFF0088 Specified I/O buffer type not supported.
3FFF0098 Successful, but invoke no handlers for this event.
3FFF0099 Successful but session has nested shared locks.
3FFF009A Successful but session has nested exclusive locks.
3FFF009B Successful but operation not asynchronous.

```

#### ERRORS:

```

BFFF0000 Unknown system error (miscellaneous error).
BFFF000E Session or object reference is invalid.
BFFF000F Resource is locked.
BFFF0010 Invalid expression specified for search.
BFFF0011 Resource is not present in the system.
BFFF0012 Invalid resource reference specified. Parsing error.
BFFF0013 Invalid access mode.
BFFF0015 Timeout expired before operation completed.
BFFF0016 Unable to deallocate session data structures.
BFFF001B Specified degree is invalid.
BFFF001C Specified job identifier is invalid.

```

BFFF001D Attribute is not supported by the referenced object.  
BFFF001E Attribute state not supported by the referenced object.  
BFFF001F Specified attribute is read-only.  
BFFF0020 Lock type lock not supported by this resource.  
BFFF0021 Invalid access key.  
BFFF0026 Specified event type not supported by the resource.  
BFFF0027 Invalid mechanism specified.  
BFFF0028 A handler was not installed.  
BFFF0029 Handler reference either invalid or was not installed.  
BFFF002A Specified event context invalid.  
BFFF002D Event queue for specified type has overflowed.  
BFFF002F Event type must be enabled in order to receive.  
BFFF0030 User abort during transfer.  
BFFF0034 Violation of raw write protocol during transfer.  
BFFF0035 Violation of raw read protocol during transfer.  
BFFF0036 Device reported output protocol error during transfer.  
BFFF0037 Device reported input protocol error during transfer.  
BFFF0038 Bus error during transfer.  
BFFF0039 Unable to queue asynchronous operation.  
BFFF003A Unable to start operation because setup is invalid.  
BFFF003B Unable to queue the asynchronous operation.  
BFFF003C Insufficient resources to perform memory allocation.  
BFFF003D Invalid buffer mask specified.  
BFFF003E I/O error.  
BFFF003F Format specifier invalid.  
BFFF0041 Format specifier not supported.  
BFFF0042 Trigger line is currently in use.  
BFFF004A Service request not received for the session.  
BFFF004E Invalid address space specified.  
BFFF0051 Invalid offset specified.  
BFFF0052 Invalid access width specified.  
BFFF0054 Offset not accessible from this hardware.  
BFFF0055 Source and destination widths are different.  
BFFF0057 Session not currently mapped.  
BFFF0059 Previous response still pending.  
BFFF005F No listeners condition detected.  
BFFF0060 Interface not currently the controller in charge.  
BFFF0061 Interface not the system controller.  
BFFF0067 Session does not support this operation.  
BFFF006A A parity error occurred during transfer.  
BFFF006B A framing error occurred during transfer.  
BFFF006C An overrun error occurred during transfer.  
BFFF0070 Offset not properly aligned for operation access width.  
BFFF0071 Specified user buffer not valid.  
BFFF0072 Resource valid, but VISA cannot access it.  
BFFF0076 Width not supported by this hardware.  
BFFF0078 Invalid parameter value, parameter unknown.  
BFFF0079 Invalid protocol.  
BFFF007B Invalid window size.  
BFFF0080 Session currently contains a mapped window.  
BFFF0081 Operation not implemented.  
BFFF0083 Invalid length.  
BFFF0091 Invalid mode.  
BFFF009C Session did not have a lock on the resource.  
BFFF009D The device does not export any memory.  
BFFF009E VISA-required code library not located or not loaded.



## VXIIPnP Driver Status Codes:

Status	Description
-----	
WARNINGS:	
3FFC0101	Instrument does not have ID Query capability.
3FFC0102	Instrument does not have Reset capability.
3FFC0103	Instrument does not have Self-Test capability.
3FFC0104	Instrument does not have Error Query capability.
3FFC0105	Instrument does not have Revision Query capability.
ERRORS:	
BFFC0001	Parameter 1 out of range, or error trying to set it.
BFFC0002	Parameter 2 out of range, or error trying to set it.
BFFC0003	Parameter 3 out of range, or error trying to set it.
BFFC0004	Parameter 4 out of range, or error trying to set it.
BFFC0005	Parameter 5 out of range, or error trying to set it.
BFFC0006	Parameter 6 out of range, or error trying to set it.
BFFC0007	Parameter 7 out of range, or error trying to set it.
BFFC0008	Parameter 8 out of range, or error trying to set it.
BFFC0011	Instrument failed the ID Query.
BFFC0012	Invalid response from instrument.

## WARNINGS:

3FFC0101	Instrument does not have ID Query capability.
3FFC0102	Instrument does not have Reset capability.
3FFC0103	Instrument does not have Self-Test capability.
3FFC0104	Instrument does not have Error Query capability.
3FFC0105	Instrument does not have Revision Query capability.

## ERRORS:

BFFC0001	Parameter 1 out of range, or error trying to set it.
BFFC0002	Parameter 2 out of range, or error trying to set it.
BFFC0003	Parameter 3 out of range, or error trying to set it.
BFFC0004	Parameter 4 out of range, or error trying to set it.
BFFC0005	Parameter 5 out of range, or error trying to set it.
BFFC0006	Parameter 6 out of range, or error trying to set it.
BFFC0007	Parameter 7 out of range, or error trying to set it.
BFFC0008	Parameter 8 out of range, or error trying to set it.
BFFC0011	Instrument failed the ID Query.
BFFC0012	Invalid response from instrument.

## errorMessage

Variable Type	ViChar[]
---------------	----------

Returns the user-readable message string that corresponds to the status code you specify.

You must pass a ViChar array with at least 256 bytes.

## Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

---

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_error\_query***

```
ViStatus axKDC_error_query (ViSession instrumentHandle,  
                           ViPInt32 errorCode,  
                           ViChar _VI_FAR errorMessage[]);
```

### Purpose

This function reads an error code and a message from the instrument's error queue.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

#### errorCode

Variable Type      ViInt32 (passed by reference)

Returns the error code read from the instrument's error queue.

#### errorMessage

Variable Type      ViChar[]

Returns the error message string read from the instrument's error message queue.

You must pass a ViChar array with at least 256 bytes.

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

**axKDC\_Fetch**

```
ViStatus axKDC_Fetch (ViSession instrumentHandle,
                    ViChar _VI_FAR channelName[],
                    ViInt32 measurementType, ViPReal64 measurement);
```

## Purpose

This function takes a measurement on the output signal and returns the fetched value.

## Parameter List

## instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

## channelName

Variable Type            ViChar[]

Pass the name of the channel on which to take a measurement.

Valid Channel Name: "1"

## measurementType

Variable Type            ViInt32

Pass the measurement you want the power supply to take.

## Valid Values:

AXKDC_VAL_MEASURE_VOLTAGE	- DC voltage measurement
AXKDC_VAL_MEASURE_CURRENT	- DC current measurement
AXKDC_VAL_MEASURE_PEAK_CURRENT	- Peak current measurement
AXKDC_VAL_MEASURE_POWER	- DC power measurement

Default Value: AXKDC\_VAL\_MEASURE\_VOLTAGE

## measurement

Variable Type            ViReal64 (passed by reference)

Returns the measured value.

Units: volts (for voltage measurement)  
 amps (for current measurement)  
 amps (for peak current measurement)  
 watts (for power measurement)

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_GetAttributeViBoolean***

```
ViStatus axKDC_GetAttributeViBoolean (ViSession instrumentHandle,  
                                     ViChar _VI_FAR channelName[],  
                                     ViAttr attributeID,  
                                     ViPBoolean attributeValue);
```

### Purpose

This function queries the value of a ViBoolean attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

#### channelName

Variable Type      ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name:    "1"

Default Value:        ""

#### attributeID

Variable Type      ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViBoolean type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViBoolean are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViBoolean (passed by reference)

Returns the current value of the attribute. Pass the address of a ViBoolean variable.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has named constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
-----	



0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_GetAttributeViInt32***

```
ViStatus axKDC_GetAttributeViInt32 (ViSession instrumentHandle,
                                     ViChar _VI_FAR channelName[],
                                     ViAttr attributeID,
                                     ViPInt32 attributeValue);
```

### Purpose

This function queries the value of a ViInt32 attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

#### channelName

Variable Type          ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name:    "1"

Default Value:        ""

#### attributeID

Variable Type          ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViInt32 type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViInt32 are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViInt32 (passed by reference)

Returns the current value of the attribute. Pass the address of a ViInt32 variable.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has named constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## **axKDC\_GetAttributeViReal64**

```
ViStatus axKDC_GetAttributeViReal64 (ViSession instrumentHandle,
                                     ViChar _VI_FAR channelName[],
                                     ViAttr attributeID,
                                     ViPReal64 attributeValue);
```

### Purpose

This function queries the value of a ViReal64 attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

#### channelName

Variable Type          ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name:    "1"

Default Value:        ""

#### attributeID

Variable Type          ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViReal64 type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViReal64 are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViReal64 (passed by reference)

Returns the current value of the attribute. Pass the address of a ViReal64 variable.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has named constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
-----	

0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_GetAttributeViSession***

```
ViStatus axKDC_GetAttributeViSession (ViSession instrumentHandle,
                                       ViChar _VI_FAR channelName[],
                                       ViAttr attributeID,
                                       ViPSession attributeValue);
```

### Purpose

This function queries the value of a ViSession attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

#### channelName

Variable Type          ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name:    "1"

Default Value:        ""

#### attributeID

Variable Type          ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.



- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViSession type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViSession are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViSession (passed by reference)

Returns the current value of the attribute. Pass the address of a ViSession variable.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has named constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_GetAttributeViString***

```
ViStatus axKDC_GetAttributeViString (ViSession instrumentHandle,
                                     ViChar _VI_FAR channelName[],
                                     ViAttr attributeID,
                                     ViInt32 bufferSize,
                                     ViChar _VI_FAR attributeValue[]);
```

#### Purpose

This function queries the value of a ViString attribute.

You can use this function to get the values of instrument- specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid.

You must provide a ViChar array to serve as a buffer for the value. You pass the number of bytes in the buffer as the Buffer Size parameter. If the current value of the attribute, including the terminating NUL byte, is larger than the size you indicate in the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you want to call this function just to get the required buffer size, you can pass 0 for the Buffer Size and VI\_NULL for the Attribute Value buffer.

If you want the function to fill in the buffer regardless of the number of bytes in the value, pass a negative number for the Buffer Size parameter.

#### Parameter List

instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

### channelName

Variable Type        ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to obtain the value of the attribute. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name:  "1"

Default Value:  ""

### attributeID

Variable Type        ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViString type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViString are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

### bufferSize

Variable Type        ViInt32

Pass the number of bytes in the ViChar array you specify for the Attribute Value parameter.

If the current value of the attribute, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size

you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI\_NULL for the Attribute Value buffer parameter.

#### attributeValue

Variable Type            ViChar[]

The buffer in which the function returns the current value of the attribute. The buffer must be of type ViChar and have at least as many bytes as indicated in the Buffer Size parameter.

If the current value of the attribute, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is

"123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you specify 0 for the Buffer Size parameter, you can pass VI\_NULL for this parameter.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has named constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

If the current value of the return buffer, including the terminating NUL byte, is larger than the size you indicate in the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about

the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_GetError***

```
ViStatus axKDC_GetError (ViSession instrumentHandle, ViPStatus code,
                        ViInt32 bufferSize,
                        ViChar _VI_FAR description[]);
```

### Purpose

This function retrieves and then clears the IVI error information for the session or the current execution thread. One exception exists: If the BufferSize parameter is 0, the function does not clear the error information. By passing 0 for the buffer size, the caller can ascertain the buffer size required to get the entire error description string and then call the function again with a sufficiently large buffer.

If the user specifies a valid IVI session for the InstrumentHandle parameter, Get Error retrieves and then clears the error information for the session. If the user passes VI\_NULL for the InstrumentHandle parameter, this function retrieves and then clears the error information for the current execution thread. If the InstrumentHandle parameter is an invalid session, the function does nothing and returns an error. Normally, the error information describes the first error that occurred since the user last called axKDC\_GetError or axKDC\_ClearError.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### code

Variable Type      ViStatus (passed by reference)

Returns the error code for the session or execution thread.

If you pass 0 for the Buffer Size, you can pass VI\_NULL for this parameter.

#### bufferSize

Variable Type      ViInt32

Pass the number of bytes in the ViChar array you specify for the Description parameter.

If the error description, including the terminating NUL byte,

contains more bytes than you indicate in this parameter, the function copies BufferSize - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI\_NULL for the Description buffer parameter.

Default Value: None

description

Variable Type      ViChar[]

Returns the error description for the IVI session or execution thread. If there is no description, the function returns an empty string.

The buffer must contain at least as many elements as the value you specify with the Buffer Size parameter. If the error description, including the terminating NUL byte, contains more bytes than you indicate with the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass 0 for the Buffer Size, you can pass VI\_NULL for this parameter.

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

If the current value of the return buffer, including the terminating NUL byte, is larger than the size you indicate in the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:



Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
WARNINGS:	
3FFA4001	Current decreased to satisfy voltage requirements.
3FFA4002	Voltage decreased to satisfy current requirements.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_GetNextCoercionRecord***

```
ViStatus axKDC_GetNextCoercionRecord (ViSession instrumentHandle,
                                       ViInt32 bufferSize,
                                       ViChar _VI_FAR coercionRecord[]);
```

### Purpose

This function returns the coercion information associated with the IVI session. This function retrieves and clears the oldest instance in which the instrument driver coerced a value you specified to another value.

If you set the AXKDC\_ATTR\_RECORD\_COERCIONS attribute to VI\_TRUE, the instrument driver keeps a list of all coercions it makes on ViInt32 or ViReal64 values you pass to instrument driver functions. You use this function to retrieve information from that list.

If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI\_NULL for the Coercion Record buffer parameter.

The function returns an empty string in the Coercion Record parameter if no coercion records remain for the session.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init function. The handle identifies a particular instrument session.

Default Value:    None

#### bufferSize

Variable Type      ViInt32

Pass the number of bytes in the ViChar array you specify for the Coercion Record parameter.

If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size

you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI\_NULL for the Coercion Record buffer parameter.

Default Value: None

#### coercionRecord

Variable Type      ViChar[]

Returns the next coercion record for the IVI session. If there are no coercion records, the function returns an empty string.

The buffer must contain at least as many elements as the value you specify with the Buffer Size parameter. If the next coercion record string, including the terminating NUL byte, contains more bytes than you indicate with the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

This parameter returns an empty string if no coercion records remain for the session.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

If the current value of the return buffer, including the terminating NUL byte, is larger than the size you indicate in the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_GetNextInterchangeWarning***

```
ViStatus axKDC_GetNextInterchangeWarning (ViSession instrumentHandle,  
                                          ViInt32 bufferSize,  
                                          ViChar _VI_FAR  
interchangeWarning[]);
```

#### Purpose

This function returns the interchangeability warnings associated with the IVI session. It retrieves and clears the oldest instance in which the class driver recorded an interchangeability warning. Interchangeability warnings indicate that using your application with a different instrument might cause different behavior. You use this function to retrieve interchangeability warnings.

The driver performs interchangeability checking when the `AXKDC_ATTR_INTERCHANGE_CHECK` attribute is set to `VI_TRUE`.

The function returns an empty string in the Interchange Warning parameter if no interchangeability warnings remain for the session.

In general, the instrument driver generates interchangeability warnings when an attribute that affects the behavior of the instrument is in a state that you did not specify.

#### Parameter List

##### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: None

##### bufferSize

Variable Type      ViInt32

Pass the number of bytes in the ViChar array you specify for the Interchange Warning parameter.

If the next interchangeability warning string, including the terminating NUL byte, contains more bytes than you indicate in this parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

If you pass a negative number, the function copies the value to the buffer regardless of the number of bytes in the value.

If you pass 0, you can pass VI\_NULL for the Interchange Warning buffer parameter.

Default Value: None

#### interchangeWarning

Variable Type            ViChar[]

Returns the next interchange warning for the IVI session. If there are no interchange warnings, the function returns an empty string.

The buffer must contain at least as many elements as the value you specify with the Buffer Size parameter. If the next interchangeability warning string, including the terminating NUL byte, contains more bytes than you indicate with the Buffer Size parameter, the function copies Buffer Size - 1 bytes into the buffer, places an ASCII NUL byte at the end of the buffer, and returns the buffer size you must pass to get the entire value. For example, if the value is "123456" and the Buffer Size is 4, the function places "123" into the buffer and returns 7.

This parameter returns an empty string if no interchangeability warnings remain for the session.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
WARNINGS:	
3FFA4001	Current decreased to satisfy voltage requirements.

3FFA4002 Voltage decreased to satisfy current requirements.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_init***

```
ViStatus axKDC_init (ViRsrc resourceName, ViBoolean IDQuery,
                    ViBoolean resetDevice, ViPSession instrumentHandle,
                    ViInt32 baudRate);
```

### Purpose

This function performs the following initialization actions:

- Creates a new IVI instrument driver session.
- Opens a session to the specified device using the interface and address you specify for the Resource Name parameter.
- If the ID Query parameter is set to VI\_TRUE, this function queries the instrument ID and checks that it is valid for this instrument driver.
- If the Reset parameter is set to VI\_TRUE, this function resets the instrument to a known state.
- Sends initialization commands to set the instrument to the state necessary for the operation of the instrument driver.
- Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.

Note: This function creates a new session each time you invoke it. Although you can open more than one IVI session for the same resource, it is best not to do so. You can use the same session in multiple program threads. You can use the `axKDC_LockSession` and `axKDC_UnlockSession` functions to protect sections of code that require exclusive access to the resource.

### Parameter List

resourceName

Variable Type	ViRsrc
---------------	--------

Pass the resource name of the device to initialize.

You can also pass the name of a virtual instrument or logical name that you configure with the IVI Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical Name identifies a particular virtual instrument.

Refer to the following table below for the exact grammar to use for this parameter. Optional fields are shown in square brackets ([]).

#### Syntax

```
-----
GPIB[board]::<primary address>[::secondary address]::INSTR
ASRL<port>::INSTR
```





### instrumentHandle

Variable Type            ViSession (passed by reference)

Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.

#### Notes:

(1) This function creates a new session each time you invoke it. This is useful if you have multiple physical instances of the same type of instrument.

(2) Avoid creating multiple concurrent sessions to the same physical instrument. Although you can create more than one IVI session for the same resource, it is best not to do so. A better approach is to use the same IVI session in multiple execution threads. You can use functions axKDC\_LockSession and axKDC\_UnlockSession to protect sections of code that require exclusive access to the resource.

### baudRate

Variable Type            ViInt32

Specify the baud rate of the serial port.

#### Baud Rate Ranges:

9600  
19200  
38400  
57600  
115200

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings

Negative Values                      Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI            Warnings
3FFF0000 to 3FFFFFFF	VISA          Warnings
3FFC0000 to 3FFCFFFF	VXIPnP      Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI            Errors
BFFF0000 to BFFFFFFF	VISA          Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_Initiate***

```
ViStatus axKDC_Initiate (ViSession instrumentHandle);
```

### Purpose

This function initiates output changes that you previously specified.

If the power supply is not currently waiting for a trigger, the power supply waits for the trigger you specify with the `axKDC_ConfigureTriggerSource` function after you call this function. After the power supply detects the trigger, it updates its voltage level and current limit to the values you specify with the `axKDC_ConfigureTriggeredVoltageLevel` and `axKDC_ConfigureTriggeredCurrentLimit` functions.

If the power supply is already waiting for a trigger, this function does nothing and returns `VI_SUCCESS`.

### Parameter List

`instrumentHandle`

Variable Type      `ViSession`

The `ViSession` handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: `None`

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

---

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_InitWithOptions***

```
ViStatus axKDC_InitWithOptions (ViRsrc resourceName, ViBoolean IDQuery,
                                ViBoolean resetDevice,
                                ViString optionString,
                                ViPSession instrumentHandle,
                                ViInt32 baudRate);
```

### Purpose

This function performs the following initialization actions:

- Creates a new IVI instrument driver and optionally sets the initial state of the following session attributes:

```
AXKDC_ATTR_RANGE_CHECK
AXKDC_ATTR_QUERY_INSTR_STATUS
AXKDC_ATTR_CACHE
AXKDC_ATTR_SIMULATE
AXKDC_ATTR_RECORD_COERCIONS
```

- Opens a session to the specified device using the interface and address you specify for the Resource Name parameter.
- If the ID Query parameter is set to VI\_TRUE, this function queries the instrument ID and checks that it is valid for this instrument driver.
- If the Reset parameter is set to VI\_TRUE, this function resets the instrument to a known state.
- Sends initialization commands to set the instrument to the state necessary for the operation of the instrument driver.
- Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.

**Note:** This function creates a new session each time you invoke it. Although you can open more than one IVI session for the same resource, it is best not to do so. You can use the same session in multiple program threads. You can use the `axKDC_LockSession` and `axKDC_UnlockSession` functions to protect sections of code that require exclusive access to the resource.

### Parameter List

resourceName

Variable Type	ViRsrc
---------------	--------

Pass the resource name of the device to initialize.

You can also pass the name of a virtual instrument or logical name that you configure with the IVI Configuration utility. The virtual instrument identifies a specific device and specifies the initial settings for the session. A logical Name identifies a particular

virtual instrument.

Refer to the following table below for the exact grammar to use for this parameter. Optional fields are shown in square brackets ([]).

#### Syntax

```
-----
GPIB[board]::<primary address>[::secondary address]::INSTR
ASRL<port>::INSTR
<LogicalName>
```

If you do not specify a value for an optional field, the following values are used:

#### Optional Field - Value

```
-----
board - 0
secondary address - none (31)
```

The following table contains example valid values for this parameter.

#### "Valid Value" - Description

```
-----
"GPIB::22::INSTR" - GPIB board 0, primary address 22 no
                    secondary address
"GPIB::22::5::INSTR" - GPIB board 0, primary address 22
                       secondary address 5
"GPIB1::22::5::INSTR" - GPIB board 1, primary address 22
                        secondary address 5
"ASRL2::INSTR" - COM port 2
"SampleInstr" - Logical name "SampleInstr"
```

```
Default Value:  "GPIB::1::INSTR"
                 "ASRL1::INSTR"
```

#### IDQuery

Variable Type            ViBoolean

Specify whether you want the instrument driver to perform an ID Query.

#### Valid Range:

```
VI_TRUE  (1) - Perform ID Query (Default Value)
VI_FALSE (0) - Skip ID Query
```

When you set this parameter to VI\_TRUE, the driver verifies that the instrument you initialize is a type that this driver supports.

Circumstances can arise where it is undesirable to send an ID Query command string to the instrument. When you set this parameter to VI\_FALSE, the function initializes the instrument without performing an ID Query.

#### resetDevice

Variable Type            ViBoolean

Specify whether you want the to reset the instrument during the initialization procedure.

Valid Range:

VI\_TRUE (1) - Reset Device (Default Value)

VI\_FALSE (0) - Don't Reset

#### optionString

Variable Type            ViString

You can use this control to set the initial value of certain attributes for the session. The following table lists the attributes and the name you use in this parameter to identify the attribute.

Name	Attribute Defined Constant
RangeCheck	AXKDC_ATTR_RANGE_CHECK
QueryInstrStatus	AXKDC_ATTR_QUERY_INSTRUMENT_STATUS
Cache	AXKDC_ATTR_CACHE
Simulate	AXKDC_ATTR_SIMULATE
RecordCoercions	AXKDC_ATTR_RECORD_COERCIONS

The format of this string is, "AttributeName=Value" where AttributeName is the name of the attribute and Value is the value to which the attribute will be set. To set multiple attributes, separate their assignments with a comma.

If you pass NULL or an empty string for this parameter and a VISA resource descriptor for the Resource Name parameter, the session uses the default values for the attributes. The default values for the attributes are shown below:

Attribute Name	Default Value
RangeCheck	VI_TRUE
QueryInstrStatus	VI_FALSE
Cache	VI_TRUE
Simulate	VI_FALSE
RecordCoercions	VI_FALSE

If you pass NULL or an empty string for this parameter and a virtual instrument or logical name for the Resource Name parameter, the session uses the values that you configure for virtual instrument or logical name with the IVI Configuration utility.

You can override the values of the attributes by assigning a value explicitly in a string you pass for this parameter. You do not have to specify all of the attributes and may leave any of them out. If you do not specify one of the attributes, its default value or the value that you configure with the IVI Configuration utility will be used.



The following are the valid values for ViBoolean attributes:

True: 1, True, or VI\_TRUE  
False: 0, False, or VI\_FALSE

You may also specify a DriverSetup option that allows you to select which DC power supply model the instrument driver emulates.

The format for the DriverSetup option is  
"DriverSetup=Tag1:Value1".

The driver emulates the model specified in the DriverSetup option even when a different model is returned by ID Query. When no model is specified in the DriverSetup option, driver emulates the model returned by ID Query. The only exception is when simulating or no ID Query is selected. For this case, the driver emulates the KDC30\_167.

Example:

```
"Simulate=1,DriverSetup=Model:KDC30_167"
```

Default Value:

```
"Simulate=0,RangeCheck=1,QueryInstrStatus=1,Cache=1"
```

#### instrumentHandle

Variable Type            ViSession (passed by reference)

Returns a ViSession handle that you use to identify the instrument in all subsequent instrument driver function calls.

Notes:

(1) This function creates a new session each time you invoke it. This is useful if you have multiple physical instances of the same type of instrument.

(2) Avoid creating multiple concurrent sessions to the same physical instrument. Although you can create more than one IVI session for the same resource, it is best not to do so. A better approach is to use the same IVI session in multiple execution threads. You can use functions axKDC\_LockSession and axKDC\_UnlockSession to protect sections of code that require exclusive access to the resource.

#### baudRate

Variable Type            ViInt32

Specify the baud rate of the serial port.

Baud Rate Ranges:

9600

19200  
38400  
57600  
115200

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_InvalidateAllAttributes***

```
ViStatus axKDC_InvalidateAllAttributes (ViSession instrumentHandle);
```

### Purpose

This function invalidates the cached values of all attributes for the session.

### Parameter List

instrumentHandle

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
WARNINGS:	
3FFA4001	Current decreased to satisfy voltage requirements.
3FFA4002	Voltage decreased to satisfy current requirements.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

---

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_LockSession***

```
ViStatus axKDC_LockSession (ViSession instrumentHandle,
                           ViPBoolean callerHasLock);
```

### Purpose

This function obtains a multithread lock on the instrument session. Before it does so, it waits until all other execution threads have released their locks on the instrument session.

Other threads might have obtained a lock on this session in the following ways:

- The user's application called `axKDC_LockSession`.
- A call to the instrument driver locked the session.
- A call to the IVI engine locked the session.

After your call to `axKDC_LockSession` returns successfully, no other threads can access the instrument session until you call `axKDC_UnlockSession`.

Use `axKDC_LockSession` and `axKDC_UnlockSession` around a sequence of calls to instrument driver functions if you require that the instrument retain its settings through the end of the sequence.

You can safely make nested calls to `axKDC_LockSession` within the same thread. To completely unlock the session, you must balance each call to

`axKDC_LockSession` with a call to `axKDC_UnlockSession`. If, however, you use the Caller Has Lock parameter in all calls to `axKDC_LockSession` and `axKDC_UnlockSession` within a function, the IVI Library locks the session only once within the function regardless of the number of calls you make to `axKDC_LockSession`. This allows you to call `axKDC_UnlockSession` just once at the end of the function.

### Parameter List

#### `instrumentHandle`

Variable Type      `ViSession`

The `ViSession` handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value:    None

#### `callerHasLock`

Variable Type      `ViBoolean` (passed by reference)

This parameter serves as a convenience. If you do not want to use

this parameter, pass VI\_NULL.

Use this parameter in complex functions to keep track of whether you obtain a lock and therefore need to unlock the session. Pass the address of a local ViBoolean variable. In the declaration of the local variable, initialize it to VI\_FALSE. Pass the address of the same local variable to any other calls you make to axKDC\_LockSession or axKDC\_UnlockSession in the same function.

The parameter is an input/output parameter. axKDC\_LockSession and axKDC\_UnlockSession each inspect the current value and take the following actions:

- If the value is VI\_TRUE, axKDC\_LockSession does not lock the session again. If the value is VI\_FALSE, axKDC\_LockSession obtains the lock and sets the value of the parameter to VI\_TRUE.
- If the value is VI\_FALSE, axKDC\_UnlockSession does not attempt to unlock the session. If the value is VI\_TRUE, axKDC\_UnlockSession releases the lock and sets the value of the parameter to VI\_FALSE.

Thus, you can, call axKDC\_UnlockSession at the end of your function without worrying about whether you actually have the lock.

Example:

```
ViStatus TestFunc (ViSession vi, ViInt32 flags)
{
    ViStatus error = VI_SUCCESS;
    ViBoolean haveLock = VI_FALSE;

    if (flags & BIT_1)
    {
        viCheckErr( axKDC_LockSession(vi, &haveLock));
        viCheckErr( TakeAction1(vi));
        if (flags & BIT_2)
        {
            viCheckErr( axKDC_UnlockSession(vi, &haveLock));
            viCheckErr( TakeAction2(vi));
            viCheckErr( axKDC_LockSession(vi, &haveLock));
        }
        if (flags & BIT_3)
            viCheckErr( TakeAction3(vi));
    }

    Error:
    /*
       At this point, you cannot really be sure that
       you have the lock. Fortunately, the haveLock
       variable takes care of that for you.
    */
    axKDC_UnlockSession(vi, &haveLock);
    return error;
}
```

Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## **axKDC\_Measure**

```
ViStatus axKDC_Measure (ViSession instrumentHandle,  
                       ViChar _VI_FAR channelName[],  
                       ViInt32 measurementType, ViPReal64 measurement);
```

### Purpose

This function takes a measurement on the output signal and returns the measured value.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type          ViChar[]

Pass the name of the channel on which to take a measurement.

Valid Channel Name: "1"

#### measurementType

Variable Type          ViInt32

Pass the measurement you want the power supply to take.

#### Valid Values:

AXKDC\_VAL\_MEASURE\_VOLTAGE - DC voltage measurement  
AXKDC\_VAL\_MEASURE\_CURRENT - DC current measurement  
AXKDC\_VAL\_MEASURE\_PEAK\_CURRENT - Peak current measurement  
AXKDC\_VAL\_MEASURE\_POWER - DC power measurement

Default Value: AXKDC\_VAL\_MEASURE\_VOLTAGE

#### measurement

Variable Type          ViReal64 (passed by reference)

Returns the measured value.



Units: volts (for voltage measurement)  
 amps (for current measurement)  
 amps (for peak current measurement)  
 watts (for power measurement)

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_QueryMaxCurrentLimit***

```
ViStatus axKDC_QueryMaxCurrentLimit (ViSession instrumentHandle,  
                                     ViChar _VI_FAR channelName[],  
                                     ViReal64 voltageLevel,  
                                     ViPReal64 maxCurrentLimit);
```

### Purpose

This function returns the maximum programmable current limit that the power supply accepts for a particular voltage level on a channel for the output range to which the power supply is currently configured.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to query for the maximum programmable current limit.

Valid Channel Name: "1"

#### voltageLevel

Variable Type      ViReal64

Pass the voltage level for which to determine the maximum programmable current limit.

Units: volts (V).

Valid Range: (depends on the model type)

Default Value: 0.0 (volts)

#### maxCurrentLimit

Variable Type            ViReal64 (passed by reference)

This parameter returns the maximum programmable current limit for the voltage level you specify.

Units: amps (A)

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

-----

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_QueryMaxVoltageLevel***

```
ViStatus axKDC_QueryMaxVoltageLevel (ViSession instrumentHandle,  
                                     ViChar _VI_FAR channelName[],  
                                     ViReal64 currentLimit,  
                                     ViPReal64 maxVoltageLevel);
```

### Purpose

This function returns the maximum programmable voltage level that the power supply accepts for a particular current limit on a channel for the output range to which the power supply is currently configured.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to query for the maximum programmable voltage level.

Valid Channel Name: "1"

#### currentLimit

Variable Type      ViReal64

Pass the current limit for which to determine the maximum programmable voltage level.

Units: amps (A).

Valid Range: (depends on the model type)

Default Value: 1.0 amps

#### maxVoltageLevel

Variable Type      ViReal64 (passed by reference)

This parameter returns the maximum programmable voltage level for the current limit you specify.

Units: volts (V)

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_QueryOutputState***

```
ViStatus axKDC_QueryOutputState (ViSession instrumentHandle,
                                ViChar _VI_FAR channelName[],
                                ViInt32 outputState,
                                ViPBoolean inState);
```

### Purpose

This function returns whether the power supply is in a particular output state.

A constant voltage condition occurs when the output voltage is equal to the value of the `AXKDC_ATTR_VOLTAGE_LEVEL` attribute and the current is less than or equal to the value of the `AXKDC_ATTR_CURRENT_LIMIT` attribute.

A constant current condition occurs when the output current is equal to the value of the `AXKDC_ATTR_CURRENT_LIMIT` attribute and the `AXKDC_ATTR_CURRENT_LIMIT_BEHAVIOR` attribute is set to `AXKDC_VAL_CURRENT_REGULATE`.

An unregulated condition occurs when the output voltage is less than the value of the `AXKDC_ATTR_VOLTAGE_LEVEL` attribute and the current is less than the value of the `AXKDC_ATTR_CURRENT_LIMIT` attribute.

An over-voltage condition occurs when the output voltage is equal to or greater than the value of the `AXKDC_ATTR_OVP_LIMIT` attribute and the `AXKDC_ATTR_OVP_ENABLED` attribute is set to `VI_TRUE`.

When an over-voltage condition occurs, the power supply's output protection disables the output. If the power supply is in an over-voltage, it does not produce power until the output protection is reset. The `axKDC_ResetOutputProtection` function resets the output protection. Once the output protection is reset, the power supply resumes generating a power signal.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel on which to query for an output state.

Valid Channel Name: "1"

#### outputState

Variable Type            ViInt32

Pass the output state for which you want to query.

#### Valid Values:

AXKDC\_VAL\_OUTPUT\_CONSTANT\_VOLTAGE - Constant Voltage State

AXKDC\_VAL\_OUTPUT\_CONSTANT\_CURRENT - Constant Current State

AXKDC\_VAL\_OUTPUT\_UNREGULATED - Unregulated State

AXKDC\_VAL\_OUTPUT\_OVER\_VOLTAGE - Over-voltage State

AXKDC\_VAL\_OUTPUT\_OVER\_TEMPERATURE - Over-temperature

#### Default Value:

AXKDC\_VAL\_OUTPUT\_CONSTANT\_VOLTAGE

#### Notes:

(1) When AXKDC\_VAL\_OUTPUT\_OVER\_TEMPERATURE is selected, channel name parameter is ignored.

#### inState

Variable Type            ViBoolean (passed by reference)

This parameter returns VI\_TRUE if the power supply is currently in the state you specify with the outputState parameter, and VI\_FALSE if it is not.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
-----	-----

0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors



## ***axKDC\_QueryTrnsListStatus***

```
ViStatus axKDC_QueryTrnsListStatus (ViSession instrumentHandle,
                                     ViChar _VI_FAR triggerStatus[]);
```

### Purpose

This function returns the transient list status that a transient list of the power supply is in process.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### triggerStatus

Variable Type          ViChar[]

Returns a transient status of a transient list execution.

Valid Return :

"BUSY", "WTRIG", "IDLE"

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	
<p>This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:</p>	
Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_QuestionableStatusRegister***

```
ViStatus axKDC_QuestionableStatusRegister (ViSession instrumentHandle,  
                                           ViInt32 eventCondition,  
                                           ViPInt32 returnedRegister);
```

### Purpose

This function returns the value of the Questionable Event register or the Questionable Condition register.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### eventCondition

Variable Type          ViInt32

Set this parameter whether to query the status of the register or to query and clear the register.

- 0 - EVEN (Query the status of the register)
- 1 - COND (Query and clear the status of the register)

#### returnedRegister

Variable Type          ViInt32 (passed by reference)

Return a register status of the power supply whether the power supply was in constant current, constant current etc...

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ReadInstrData***

```
ViStatus axKDC_ReadInstrData (ViSession instrumentHandle,  
                              ViInt32 number_ofBytesToRead,  
                              ViChar _VI_FAR readBuffer[],  
                              ViPInt32 numBytesRead);
```

### Purpose

This function reads data from the instrument.

### Parameter List

#### instrumentHandle

Variable Type          ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### number\_ofBytesToRead

Variable Type          ViInt32

Pass the maximum number of bytes to read from the instruments.

Valid Range: 0 to the number of elements in the Read Buffer.

Default: 0

#### readBuffer

Variable Type          ViChar[]

After this function executes, this parameter contains the data that was read from the instrument.

#### numBytesRead

Variable Type          ViInt32 (passed by reference)

Returns the number of bytes actually read from the instrument and stored in the Read Buffer.

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

**axKDC\_reset**

```
ViStatus axKDC_reset (ViSession instrumentHandle);
```

## Purpose

This function resets the instrument to a known state and sends initialization commands to the instrument. The initialization commands set instrument settings such as Headers Off, Short Command form, and Data Transfer Binary to the state necessary for the operation of the instrument driver.

## Parameter List

```
instrumentHandle
```

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

## Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the

particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors



***axKDC\_ResetInterchangeCheck***

```
ViStatus axKDC_ResetInterchangeCheck (ViSession instrumentHandle);
```

**Purpose**

When developing a complex test system that consists of multiple test modules, it is generally a good idea to design the test modules so that they can run in any order. To do so requires ensuring that each test module completely configures the state of each instrument it uses. If a particular test module does not completely configure the state of an instrument, the state of the instrument depends on the configuration from a previously executed test module. If you execute the test modules in a different order, the behavior of the instrument and therefore the entire test module is likely to change. This change in behavior is generally instrument specific and represents an interchangeability problem.

You can use this function to test for such cases. After you call this function, the interchangeability checking algorithms in the specific driver ignore all previous configuration operations. By calling this function at the beginning of a test module, you can determine whether the test module has dependencies on the operation of previously executed test modules.

This function does not clear the interchangeability warnings from the list of previously recorded interchangeability warnings. If you want to guarantee that the `axKDC_GetNextInterchangeWarning` function only returns those interchangeability warnings that are generated after calling this function, you must clear the list of interchangeability warnings. You can clear the interchangeability warnings list by repeatedly calling the `axKDC_GetNextInterchangeWarning` function until no more interchangeability warnings are returned. If you are not interested in the content of those warnings, you can call the `axKDC_ClearInterchangeWarnings` function.

**Parameter List**

`instrumentHandle`

Variable Type      `ViSession`

The `ViSession` handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value:    `None`

**Return Value**

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about

the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
WARNINGS:	
3FFA4001	Current decreased to satisfy voltage requirements.
3FFA4002	Voltage decreased to satisfy current requirements.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ResetOutputProtection***

```
ViStatus axKDC_ResetOutputProtection (ViSession instrumentHandle,  
                                     ViChar _VI_FAR channelName[]);
```

### Purpose

This function resets the power supply's output protection after an over-voltage.

An over-voltage condition occurs when the output voltage is equal to or greater than the value of the AXKDC\_ATTR\_OVP\_LIMIT attribute and the AXKDC\_ATTR\_OVP\_ENABLED attribute is set to VI\_TRUE.

When an over-voltage condition occurs, the output protection of the power supply disables the output. Once the output protection is reset, the power supply resumes generating a power signal.

You use the axKDC\_QueryOutputState function to determine if the power supply is in an output-protection state.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### channelName

Variable Type      ViChar[]

Pass the name of the channel you want to reset.

Valid Channel Name: "1"

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError

function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_ResetPeakCurrent***

```
ViStatus axKDC_ResetPeakCurrent (ViSession instrumentHandle);
```

### Purpose

This function resets the power supply's output protection after an over-voltage.

An over-voltage condition occurs when the output voltage is equal to or greater than the value of the `AXKDC_ATTR_OVP_LIMIT` attribute and the `AXKDC_ATTR_OVP_ENABLED` attribute is set to `VI_TRUE`.

When an over-voltage condition occurs, the output protection of the power supply disables the output. Once the output protection is reset, the power supply resumes generating a power signal.

You use the `axKDC_QueryOutputState` function to determine if the power supply is in an output-protection state.

### Parameter List

`instrumentHandle`

Variable Type      `ViSession`

The `ViSession` handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: `None`

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
-----	
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

---

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

---

## ***axKDC\_ResetWithDefaults***

```
ViStatus axKDC_ResetWithDefaults (ViSession instrumentHandle);
```

### Purpose

This function resets the instrument and applies initial user specified settings from the Logical Name which was used to initialize the session. If the session was created without a Logical Name, this function is equivalent to the `axKDC_reset` function.

### Parameter List

`instrumentHandle`

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: None

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
--------	-------------

#### WARNINGS:

3FFA4001	Current decreased to satisfy voltage requirements.
3FFA4002	Voltage decreased to satisfy current requirements.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the

particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviFgen Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviFgen Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors



## ***axKDC\_revision\_query***

```
ViStatus axKDC_revision_query (ViSession instrumentHandle,  
                               ViChar _VI_FAR instrumentDriverRevision[],  
                               ViChar _VI_FAR firmwareRevision[]);
```

### Purpose

This function returns the revision numbers of the instrument driver and instrument firmware.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### instrumentDriverRevision

Variable Type      ViChar[]

Returns the instrument driver software revision numbers in the form of a string.

You must pass a ViChar array with at least 256 bytes.

#### firmwareRevision

Variable Type      ViChar[]

Returns the instrument firmware revision numbers in the form of a string.

You must pass a ViChar array with at least 256 bytes.

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
-----	
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

**axKDC\_self\_test**

```
ViStatus axKDC_self_test (ViSession instrumentHandle,
                          ViPInt16 selfTestResult,
                          ViChar _VI_FAR selfTestMessage[]);
```

## Purpose

This function runs the instrument's self test routine and returns the test result(s).

## Parameter List

instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

selfTestResult

Variable Type      ViInt16 (passed by reference)

This control contains the value returned from the instrument self test. Zero means success. For any other code, see the device's operator's manual.

Self-Test Code	Description
-----	
0	Passed self test
1	Self test failed

selfTestMessage

Variable Type      ViChar[]

Returns the self-test response string from the instrument. See the device's operation manual for an explanation of the string's contents.

You must pass a ViChar array with at least 256 bytes.

## Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You

examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_SendSoftwareTrigger***

```
ViStatus axKDC_SendSoftwareTrigger (ViSession instrumentHandle);
```

### Purpose

This function sends a command to trigger the power supply. Call this function if you configure the power supply to respond to software triggers. If the power supply is not configured to respond to software triggers, this function returns the error `AXKDC_ERROR_TRIGGER_NOT_SOFTWARE`.

### Parameter List

instrumentHandle

Variable Type	ViSession
---------------	-----------

The ViSession handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: None

### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the

particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_SetAttributeViBoolean***

```
ViStatus axKDC_SetAttributeViBoolean (ViSession instrumentHandle,
                                     ViChar _VI_FAR channelName[],
                                     ViAttr attributeID,
                                     ViBoolean attributeValue);
```

### Purpose

This function sets the value of a ViBoolean attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

### Parameter List

instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type      ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to set the value of the attribute. If the

attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type            ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViBoolean type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViBoolean are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViBoolean

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the



current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## **axKDC\_SetAttributeViInt32**

```
ViStatus axKDC_SetAttributeViInt32 (ViSession instrumentHandle,  
                                   ViChar _VI_FAR channelName[],  
                                   ViAttr attributeID,  
                                   ViInt32 attributeValue);
```

### Purpose

This function sets the value of a ViInt32 attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

### Parameter List

instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

channelName

Variable Type      ViChar[]

If the attribute is channel-based, this parameter specifies the name

of the channel on which to set the value of the attribute. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type          ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViInt32 type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViInt32 are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type          ViInt32

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## **axKDC\_SetAttributeViReal64**

```
ViStatus axKDC_SetAttributeViReal64 (ViSession instrumentHandle,
                                     ViChar _VI_FAR channelName[],
                                     ViAttr attributeID,
                                     ViReal64 attributeValue);
```

### Purpose

This function sets the value of a ViReal64 attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

### Parameter List

instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type      ViChar[]

If the attribute is channel-based, this parameter specifies the name

of the channel on which to set the value of the attribute. If the attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type            ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViReal64 type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViReal64 are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViReal64

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_SetAttributeViSession***

```
ViStatus axKDC_SetAttributeViSession (ViSession instrumentHandle,  
                                       ViChar _VI_FAR channelName[],  
                                       ViAttr attributeID,  
                                       ViSession attributeValue);
```

### Purpose

This function sets the value of a ViSession attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

### Parameter List

instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type      ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to set the value of the attribute. If the



attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type            ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViSession type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViSession are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViSession

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the

current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_SetAttributeViString***

```
ViStatus axKDC_SetAttributeViString (ViSession instrumentHandle,  
                                     ViChar _VI_FAR channelName[],  
                                     ViAttr attributeID,  
                                     ViChar _VI_FAR attributeValue[]);
```

### Purpose

This function sets the value of a ViString attribute.

This is a low-level function that you can use to set the values of instrument-specific attributes and inherent IVI attributes. If the attribute represents an instrument state, this function performs instrument I/O in the following cases:

- State caching is disabled for the entire session or for the particular attribute.
- State caching is enabled and the currently cached value is invalid or is different than the value you specify.

This instrument driver contains high-level functions that set most of the instrument attributes. It is best to use the high-level driver functions as much as possible. They handle order dependencies and multithread locking for you. In addition, they perform status checking only after setting all of the attributes. In contrast, when you set multiple attributes using the SetAttribute functions, the functions check the instrument status after each call.

Also, when state caching is enabled, the high-level functions that configure multiple attributes perform instrument I/O only for the attributes whose value you change. Thus, you can safely call the high-level functions without the penalty of redundant instrument I/O.

### Parameter List

instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

channelName

Variable Type      ViChar[]

If the attribute is channel-based, this parameter specifies the name of the channel on which to set the value of the attribute. If the

attribute is not channel-based, then pass VI\_NULL or an empty string.

Valid Channel Name: "1"

Default Value: ""

#### attributeID

Variable Type            ViAttr

Pass the ID of an attribute.

From the function panel window, you can use this control as follows.

- Click on the control or press <ENTER>, <spacebar>, or <ctrl-down arrow>, to display a dialog box containing a hierarchical list of the available attributes. Attributes whose value cannot be set are dim. Help text is shown for each attribute. Select an attribute by double-clicking on it or by selecting it and then pressing <ENTER>.

Read-only attributes appear dim in the list box. If you select a read-only attribute, an error message appears.

A ring control at the top of the dialog box allows you to see all IVI attributes or only the attributes of the ViString type. If you choose to see all IVI attributes, the data types appear to the right of the attribute names in the list box. Attributes with data types other than ViString are dim. If you select an attribute data type that is dim, LabWindows/CVI transfers you to the function panel for the corresponding function that is consistent with the data type.

- If you want to enter a variable name, press <CTRL-T> to change this ring control to a manual input box.
- If the attribute in this ring control has named constants as valid values, you can view the constants by moving to the Attribute Value control and pressing <ENTER>.

#### attributeValue

Variable Type            ViChar[]

Pass the value to which you want to set the attribute.

From the function panel window, you can use this control as follows.

- If the attribute currently showing in the Attribute ID ring control has constants as valid values, you can view a list of the constants by pressing <ENTER> on this control. Select a value by double-clicking on it or by selecting it and then pressing <ENTER>.

Note: Some of the values might not be valid depending on the

current settings of the instrument session.

Default Value: none

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the

particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

## ***axKDC\_StoreRecallRegister***

```
ViStatus axKDC_StoreRecallRegister (ViSession instrumentHandle,
                                   ViBoolean storeRecall,
                                   ViInt32 registers);
```

### Purpose

This function resets the power supply's output protection after an over-voltage.

An over-voltage condition occurs when the output voltage is equal to or greater than the value of the AXKDC\_ATTR\_OVP\_LIMIT attribute and the AXKDC\_ATTR\_OVP\_ENABLED attribute is set to VI\_TRUE.

When an over-voltage condition occurs, the output protection of the power supply disables the output. Once the output protection is reset, the power supply resumes generating a power signal.

You use the axKDC\_QueryOutputState function to determine if the power supply is in an output-protection state.

### Parameter List

#### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value: None

#### storeRecall

Variable Type      ViBoolean

Pass this parameter to determine whether to store the setting to a register or to recall from a register.

0 - Store  
1 - Recall

#### registers

Variable Type      ViInt32

Assign a register number to be recalled or stored of the settings.

Valid registers : 0 - 7

## Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
No defined status codes.	

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_UnlockSession***

```
ViStatus axKDC_UnlockSession (ViSession instrumentHandle,
                             ViPBoolean callerHasLock);
```

#### Purpose

This function releases a lock that you acquired on an instrument session using `axKDC_LockSession`. Refer to `axKDC_LockSession` for additional information on session locks.

#### Parameter List

##### instrumentHandle

Variable Type            ViSession

The ViSession handle that you obtain from the `axKDC_init` or `axKDC_InitWithOptions` function. The handle identifies a particular instrument session.

Default Value: None

##### callerHasLock

Variable Type            ViPBoolean (passed by reference)

This parameter serves as a convenience. If you do not want to use this parameter, pass `VI_NULL`.

Use this parameter in complex functions to keep track of whether you obtain a lock and therefore need to unlock the session.

Pass the address of a local ViPBoolean variable. In the declaration of the local variable, initialize it to `VI_FALSE`. Pass the address of the same local variable to any other calls you make to `axKDC_LockSession` or `axKDC_UnlockSession` in the same function.

The parameter is an input/output parameter. `axKDC_LockSession` and `axKDC_UnlockSession` each inspect the current value and take the following actions:

- If the value is `VI_TRUE`, `axKDC_LockSession` does not lock the session again. If the value is `VI_FALSE`, `axKDC_LockSession` obtains the lock and sets the value of the parameter to `VI_TRUE`.

- If the value is `VI_FALSE`, `axKDC_UnlockSession` does not attempt to unlock the session. If the value is `VI_TRUE`, `axKDC_UnlockSession` releases the lock and sets the value of the parameter to `VI_FALSE`.

Thus, you can, call `axKDC_UnlockSession` at the end of your function without worrying about whether you actually have the lock.

#### Example:

```
ViStatus TestFunc (ViSession vi, ViInt32 flags)
```



```

{
    ViStatus error = VI_SUCCESS;
    ViBoolean haveLock = VI_FALSE;

    if (flags & BIT_1)
    {
        viCheckErr( axKDC_LockSession(vi, &haveLock));
        viCheckErr( TakeAction1(vi));
        if (flags & BIT_2)
        {
            viCheckErr( axKDC_UnlockSession(vi, &haveLock));
            viCheckErr( TakeAction2(vi));
            viCheckErr( axKDC_LockSession(vi, &haveLock));
        }
        if (flags & BIT_3)
            viCheckErr( TakeAction3(vi));
    }

Error:
    /*
    At this point, you cannot really be sure that
    you have the lock.  Fortunately, the haveLock
    variable takes care of that for you.
    */
    axKDC_UnlockSession(vi, &haveLock);
    return error;
}

```

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the `axKDC_error_message` function. To obtain additional information about the error condition, call the `axKDC_GetError` function. To clear the error information from the driver, call the `axKDC_ClearError` function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	
No defined status codes.	

This instrument driver also returns errors and warnings defined by

other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

### ***axKDC\_WriteInstrData***

```
ViStatus axKDC_WriteInstrData (ViSession instrumentHandle,
                               ViChar _VI_FAR writeBuffer[]);
```

#### Purpose

This function writes a user-specified string to the instrument.

**Note:** This function bypasses IVI attribute state caching. Therefore, when you call this function, the cached values for all attributes will be invalidated.

#### Parameter List

##### instrumentHandle

Variable Type      ViSession

The ViSession handle that you obtain from the axKDC\_init or axKDC\_InitWithOptions function. The handle identifies a particular instrument session.

Default Value:    None

##### writeBuffer

Variable Type      ViChar[]

Pass the string to be written to the instrument.

#### Return Value

Returns the status code of this operation. The status code either indicates success or describes an error or warning condition. You examine the status code from each call to an instrument driver function to determine if an error occurred.

To obtain a text description of the status code, call the axKDC\_error\_message function. To obtain additional information about the error condition, call the axKDC\_GetError function. To clear the error information from the driver, call the axKDC\_ClearError function.

The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

This driver defines the following status codes:

Status	Description
-----	

No defined status codes.

This instrument driver also returns errors and warnings defined by other sources. The following table defines the ranges of additional status codes that this driver can return. The table lists the different include files that contain the defined constants for the particular status codes:

Numeric Range (in Hex)	Status Code Types
3FFA2000 to 3FFA3FFF	IviPower Warnings
3FFA0000 to 3FFA1FFF	IVI Warnings
3FFF0000 to 3FFFFFFF	VISA Warnings
3FFC0000 to 3FFCFFFF	VXIPnP Driver Warnings
BFFA2000 to BFFA3FFF	IviPower Errors
BFFA0000 to BFFA1FFF	IVI Errors
BFFF0000 to BFFFFFFF	VISA Errors
BFFC0000 to BFFCFFFF	VXIPnP Driver Errors

Attribute Information for the Following Functions:

```
axKDC_SetAttributeViInt32
axKDC_GetAttributeViInt32
axKDC_CheckAttributeViInt32
axKDC_SetAttributeViReal64
axKDC_GetAttributeViReal64
axKDC_CheckAttributeViReal64
axKDC_SetAttributeViSession
axKDC_GetAttributeViSession
axKDC_CheckAttributeViSession
axKDC_SetAttributeViBoolean
axKDC_GetAttributeViBoolean
axKDC_CheckAttributeViBoolean
axKDC_SetAttributeViString
axKDC_GetAttributeViString
axKDC_CheckAttributeViString
```

#### Inherent IVI Attributes

```
User Options
  Range Check                                AXKDC_ATTR_RANGE_CHECK
  Query Instrument Status
AXKDC_ATTR_QUERY_INSTRUMENT_STATUS
  Cache                                       AXKDC_ATTR_CACHE
  Simulate                                    AXKDC_ATTR_SIMULATE
  Record Value Coercions                     AXKDC_ATTR_RECORD_COERCIONS
  Interchange Check                           AXKDC_ATTR_INTERCHANGE_CHECK
Driver Identification
  Description
AXKDC_ATTR_SPECIFIC_DRIVER_DESCRIPTION
  Driver Prefix
AXKDC_ATTR_SPECIFIC_DRIVER_PREFIX
```

```

    Driver Vendor
AXKDC_ATTR_SPECIFIC_DRIVER_VENDOR
    Revision
AXKDC_ATTR_SPECIFIC_DRIVER_REVISION
    Class Specification Major Version
AXKDC_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MAJOR_VERSION
    Class Specification Minor Version
AXKDC_ATTR_SPECIFIC_DRIVER_CLASS_SPEC_MINOR_VERSION
    Driver Capabilities
    Supported Instrument Models
AXKDC_ATTR_SUPPORTED_INSTRUMENT_MODELS
    Class Group Capabilities
AXKDC_ATTR_GROUP_CAPABILITIES
    Instrument Identification
    Manufacturer
AXKDC_ATTR_INSTRUMENT_MANUFACTURER
    Model
AXKDC_ATTR_INSTRUMENT_MODEL
    Firmware Revision
AXKDC_ATTR_INSTRUMENT_FIRMWARE_REVISION
    Advanced Session Information
    Logical Name
AXKDC_ATTR_LOGICAL_NAME
    I/O Resource Descriptor
AXKDC_ATTR_IO_RESOURCE_DESCRIPTOR
    Driver Setup
AXKDC_ATTR_DRIVER_SETUP
    Basic Operation
    Output Enabled
AXKDC_ATTR_OUTPUT_ENABLED
    OVP Limit
AXKDC_ATTR_OVP_LIMIT
    Current Limit Behavior
AXKDC_ATTR_CURRENT_LIMIT_BEHAVIOR
    Current Limit
AXKDC_ATTR_CURRENT_LIMIT
    Voltage Level
AXKDC_ATTR_VOLTAGE_LEVEL
    Trigger Subsystem
    Trigger Source
AXKDC_ATTR_TRIGGER_SOURCE
    Triggered Voltage Level
AXKDC_ATTR_TRIGGERED_VOLTAGE_LEVEL
    Triggered Current Limit
AXKDC_ATTR_TRIGGERED_CURRENT_LIMIT
    Miscellaneous Attributes
    ID Query Response
AXKDC_ATTR_ID_QUERY_RESPONSE
    AXKDC_ATTR_OUTPUT_RELAY_ENABLED
AXKDC_ATTR_OUTPUT_RELAY_ENABLED
    AXKDC_ATTR_VOLTAGE_MAXIMUM
AXKDC_ATTR_VOLTAGE_MAXIMUM
    AXKDC_ATTR_CURRENT_MAXIMUM
AXKDC_ATTR_CURRENT_MAXIMUM
    AXKDC_ATTR_CONSTANT_POWER_VALUE
AXKDC_ATTR_CONSTANT_POWER_VALUE
    AXKDC_ATTR_OUTPUT_MODE
AXKDC_ATTR_OUTPUT_MODE
    AXKDC_ATTR_OVP_MODE
AXKDC_ATTR_OVP_MODE
    AXKDC_ATTR_OVP_MIN_LIMIT
AXKDC_ATTR_OVP_MIN_LIMIT
    AXKDC_ATTR_OVP_MAX_LIMIT
AXKDC_ATTR_OVP_MAX_LIMIT
    AXKDC_ATTR_OUTPUT_TRIGGER_ENABLED
AXKDC_ATTR_OUTPUT_TRIGGER_ENABLED
    AXKDC_ATTR_OUTPUT_TRIGGER_SOURCE
AXKDC_ATTR_OUTPUT_TRIGGER_SOURCE
    AXKDC_ATTR_TRIGGERED_VOLTAGE_MODE
AXKDC_ATTR_TRIGGERED_VOLTAGE_MODE
    AXKDC_ATTR_TRIGGERED_SLEW_VOLTAGE_RATE_MODE
AXKDC_ATTR_TRIGGERED_SLEW_VOLTAGE_RATE_MODE

```

AXKDC_ATTR_TRIGGERED_SLEW_VOLTAGE_RATE	
AXKDC_ATTR_TRIGGERED_SLEW_VOLTAGE_RATE	
AXKDC_ATTR_TRIGGER_LIST_COUNT	AXKDC_ATTR_TRIGGER_LIST_COUNT
AXKDC_ATTR_TRIGGER_LIST_MODE	AXKDC_ATTR_TRIGGER_LIST_MODE
AXKDC_ATTR_TRIGGER_DELAY	AXKDC_ATTR_TRIGGER_DELAY

**AXKDC\_ATTR\_CACHE**

Data Type: ViBoolean

Description: Specifies whether to cache the value of attributes. When caching is enabled, the instrument driver keeps track of the current instrument settings and avoids sending redundant commands to the instrument. Thus, you can significantly increase execution speed.

The instrument driver can choose always to cache or never to cache particular attributes regardless of the setting of this attribute.

The default value is VI\_TRUE. Use the `axKDC_InitWithOptions` function to override this value.

**AXKDC\_ATTR\_CONSTANT\_POWER\_VALUE**

Data Type: ViReal64

Description:

**AXKDC\_ATTR\_CURRENT\_LIMIT**

Data Type: ViReal64

Description: This channel-based attribute specifies the output current limit. The units are Amps.

The value of the `AXKDC_ATTR_CURRENT_LIMIT_BEHAVIOR` attribute determines the behavior of the power supply when the output current is equal to or greater than the value of this attribute.

**AXKDC\_ATTR\_CURRENT\_LIMIT\_BEHAVIOR**

Data Type: ViInt32

Description: This channel-based attribute specifies the behavior of the power supply when the output current is equal to or greater than the value of the `AXKDC_ATTR_CURRENT_LIMIT` attribute.

Values:

AXKDC\_VAL\_CURRENT\_REGULATE

0

When the output current reaches the value of the `AXKDC_ATTR_CURRENT_LIMIT` attribute, the power supply restricts the output voltage such that the output current is not greater than the value of the `AXKDC_ATTR_CURRENT_LIMIT` attribute.

**AXKDC\_ATTR\_CURRENT\_MAXIMUM**

Data Type: ViReal64

Description:

**AXKDC\_ATTR\_DRIVER\_SETUP**

Data Type: ViString

Restrictions: Not settable.

Description: Some cases exist where you must specify instrument driver options at initialization time. An example of this is specifying a particular

instrument model from among a family of instruments that the driver supports. This is useful when using simulation. You can specify driver-specific options through the DriverSetup keyword in the optionsString parameter to the axKDC\_InitWithOptions function. If you open an instrument using a logical name, you can also specify the options through the IVI Configuration Utility.

The default value is an empty string.

#### AXKDC\_ATTR\_GROUP\_CAPABILITIES

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains a comma-separated list of class-extended groups that this driver implements.

#### AXKDC\_ATTR\_ID\_QUERY\_RESPONSE

Data Type: ViString

Restrictions: Not settable.

Description: Returns the ID Query response string. The instrument driver gets the value of this attribute when you pass VI\_TRUE for the ID Query parameter to the axKDC\_init or axKDC\_InitWithOptions function.

#### AXKDC\_ATTR\_INSTRUMENT\_FIRMWARE\_REVISION

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains the firmware revision information for the instrument you are currently using.

#### AXKDC\_ATTR\_INSTRUMENT\_MANUFACTURER

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains the name of the instrument manufacturer you are currently using.

#### AXKDC\_ATTR\_INSTRUMENT\_MODEL

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains the model number or name of the instrument that you are currently using.

#### AXKDC\_ATTR\_INTERCHANGE\_CHECK

Data Type: ViBoolean

Description: Specifies whether to perform interchangeability checking and retrieve interchangeability warnings.

The default value is VI\_FALSE.

Interchangeability warnings indicate that using your application with a different instrument might cause different behavior. You call axKDC Get Next Interchange Warning to extract interchange warnings. Call the axKDC\_ClearInterchangeWarnings function to clear the list of interchangeability warnings without reading them.

Interchangeability checking logs a warning for each attribute you have not set

that affects the behavior of the instrument.

#### AXKDC\_ATTR\_IO\_RESOURCE\_DESCRIPTOR

Data Type: ViString

Restrictions: Not settable.

Description: Indicates the resource descriptor the driver uses to identify the physical device.

If you initialize the driver with a logical name, this attribute contains the resource descriptor that corresponds to the entry in the IVI Configuration utility.

If you initialize the instrument driver with the resource descriptor, this attribute contains that value.

#### AXKDC\_ATTR\_LOGICAL\_NAME

Data Type: ViString

Restrictions: Not settable.

Description: A string containing the logical name you specified when opening the current IVI session.

You may pass a logical name to the `axKDC_init` or `axKDC_InitWithOptions` functions. The IVI Configuration utility must contain an entry for the logical name. The logical name entry refers to a virtual instrument section in the IVI Configuration file. The virtual instrument section specifies a physical device and initial user options.

#### AXKDC\_ATTR\_OUTPUT\_ENABLED

Data Type: ViBoolean

Description: This channel-based attribute specifies whether the signal the power supply produces appears at the output connector.

#### AXKDC\_ATTR\_OUTPUT\_MODE

Data Type: ViString

Description:

#### AXKDC\_ATTR\_OUTPUT\_RELAY\_ENABLED

Data Type: ViBoolean

Description:

#### AXKDC\_ATTR\_OUTPUT\_TRIGGER\_ENABLED

Data Type: ViBoolean

Description:

#### AXKDC\_ATTR\_OUTPUT\_TRIGGER\_SOURCE

Data Type: ViInt32

Description:

Values:

AXKDC_VAL_OUTPUT_TRIGGER_SOURCE_BOT	0
AXKDC_VAL_OUTPUT_TRIGGER_SOURCE_EOT	0
AXKDC_VAL_OUTPUT_TRIGGER_SOURCE_LIST	0



**AXKDC\_ATTR\_OVP\_LIMIT**

Data Type: ViReal64

Description: This channel-based attribute specifies the voltage the power supply allows. The units are Volts.

If the AXKDC\_ATTR\_OVP\_ENABLED attribute is set to VI\_TRUE, the power supply disables the output when the output voltage is greater than or equal to the value of this attribute.

If the AXKDC\_ATTR\_OVP\_ENABLED attribute is set to VI\_FALSE, this attribute does not affect the behavior of the instrument.

**AXKDC\_ATTR\_OVP\_MAX\_LIMIT**

Data Type: ViReal64

Description:

**AXKDC\_ATTR\_OVP\_MIN\_LIMIT**

Data Type: ViReal64

Description:

**AXKDC\_ATTR\_OVP\_MODE**

Data Type: ViString

Description:

**AXKDC\_ATTR\_QUERY\_INSTRUMENT\_STATUS**

Data Type: ViBoolean

Description: Specifies whether the instrument driver queries the instrument status after each operation. Querying the instrument status is very useful for debugging. After you validate your program, you can set this attribute to VI\_FALSE to disable status checking and maximize performance

The instrument driver can choose to ignore status checking for particular attributes regardless of the setting of this attribute.

The default value is VI\_FALSE. Use the axKDC\_InitWithOptions function to override this value.

**AXKDC\_ATTR\_RANGE\_CHECK**

Data Type: ViBoolean

Description: Specifies whether to validate attribute values and function parameters. If enabled, the instrument driver validates the parameter values that you pass to driver functions. Range checking parameters is very useful for debugging. After you validate your program, you can set this attribute to VI\_FALSE to disable range checking and maximize performance.

The default value is VI\_TRUE. Use the axKDC\_InitWithOptions function to override this value.

**AXKDC\_ATTR\_RECORD\_COERCIONS**

Data Type: ViBoolean

Description: Specifies whether the IVI engine keeps a list of the value coercions it makes for integer and real type attributes. You call axKDC Get

Next Coercion Record to extract and delete the oldest coercion record from the list.

The default value is VI\_FALSE. Use the axKDC\_InitWithOptions function to override this value.

#### AXKDC\_ATTR\_SIMULATE

Data Type: ViBoolean

Description: Specifies whether or not to simulate instrument driver I/O operations. If simulation is enabled, instrument driver functions perform range checking and call Ivi\_GetAttribute and Ivi\_SetAttribute functions, but they do not perform instrument I/O. For output parameters that represent instrument data, the instrument driver functions return calculated values.

The default value is VI\_FALSE. Use the axKDC\_InitWithOptions function to override this value.

#### AXKDC\_ATTR\_SPECIFIC\_DRIVER\_CLASS\_SPEC\_MAJOR\_VERSION

Data Type: ViInt32

Restrictions: Not settable.

Description: The major version number of the class specification with which this driver is compliant.

#### AXKDC\_ATTR\_SPECIFIC\_DRIVER\_CLASS\_SPEC\_MINOR\_VERSION

Data Type: ViInt32

Restrictions: Not settable.

Description: The minor version number of the class specification with which this driver is compliant.

#### AXKDC\_ATTR\_SPECIFIC\_DRIVER\_DESCRIPTION

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains a brief description of the specific driver.

#### AXKDC\_ATTR\_SPECIFIC\_DRIVER\_PREFIX

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains the prefix for the instrument driver. The name of each user-callable function in this driver starts with this prefix.

#### AXKDC\_ATTR\_SPECIFIC\_DRIVER\_REVISION

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains additional version information about this instrument driver.

#### AXKDC\_ATTR\_SPECIFIC\_DRIVER\_VENDOR

Data Type: ViString

Restrictions: Not settable.

Description: A string that contains the name of the vendor that supplies this driver.

**AXKDC\_ATTR\_SUPPORTED\_INSTRUMENT\_MODELS**

Data Type: ViString

Restrictions: Not settable.

Description: Contains a model code of the instrument. For drivers that support more than one device, this attribute contains a comma-separated list of supported instrument models.

**AXKDC\_ATTR\_TRIGGER\_DELAY**

Data Type: ViReal64

Description:

**AXKDC\_ATTR\_TRIGGER\_LIST\_COUNT**

Data Type: ViInt32

Description:

**AXKDC\_ATTR\_TRIGGER\_LIST\_MODE**

Data Type: ViInt32

Description:

Values:

AXKDC_VAL_TRIGGER_LIST_STEP_ONCE	0
----------------------------------	---

AXKDC_VAL_TRIGGER_LIST_STEP_AUTO	0
----------------------------------	---

**AXKDC\_ATTR\_TRIGGER\_SOURCE**

Data Type: ViInt32

Description: This channel-based attribute specifies the trigger source. After you call `axKDC_Initiate`, the power supply waits for a trigger event from the source you specify with this attribute. After a trigger event occurs, the power supply changes the voltage level to the value of the `AXKDC_ATTR_TRIGGERED_VOLTAGE_LEVEL` and the current limit to the value of the `AXKDC_ATTR_TRIGGERED_CURRENT_LIMIT` attributes.

Values:

AXKDC_VAL_TRIG_IMMEDIATE	0
--------------------------	---

The power supply does not wait for a trigger before changing the output signal.

AXKDC_VAL_TRIG_EXTERNAL	1
-------------------------	---

The power supply waits for a trigger on the external trigger input.

AXKDC_VAL_SOFTWARE_TRIG	2
-------------------------	---

The power supply waits for you to call the `axKDC_SendSoftwareTrigger` function.

**AXKDC\_ATTR\_TRIGGERED\_CURRENT\_LIMIT**

Data Type: ViReal64

Description: This channel-based attribute specifies the value to which the power supply sets the current limit after a trigger event occurs. The units are amps.

After you call `axKDC_Initiate`, the power supply waits for a trigger event from the source you specify with the `AXKDC_ATTR_TRIGGER_SOURCE` attribute. After a

trigger event occurs, the power supply sets the current limit to the value of this attribute. After a trigger occurs, the value of the `AXKDC_ATTR_CURRENT_LIMIT` attribute reflects the new value to which the current limit has been set.

#### `AXKDC_ATTR_TRIGGERED_SLEW_VOLTAGE_RATE`

Data Type: `ViReal64`

Description:

#### `AXKDC_ATTR_TRIGGERED_SLEW_VOLTAGE_RATE_MODE`

Data Type: `ViInt32`

Description:

Values:

<code>AXKDC_VAL_TRIGGER_MODE_FIX</code>	0
<code>AXKDC_VAL_TRIGGER_MODE_STEP</code>	0
<code>AXKDC_VAL_TRIGGER_MODE_PULSE</code>	0
<code>AXKDC_VAL_TRIGGER_MODE_LIST</code>	0

#### `AXKDC_ATTR_TRIGGERED_VOLTAGE_LEVEL`

Data Type: `ViReal64`

Description: This channel-based attribute specifies the value to which the power supply sets the voltage level after a trigger event occurs. The units are Volts.

After you call `axKDC_Initiate`, the power supply waits for a trigger event from the source you specify with the `AXKDC_ATTR_TRIGGER_SOURCE` attribute. After a trigger event occurs, the power supply sets the voltage level to the value of this attribute. After a trigger occurs, the value of the `AXKDC_ATTR_VOLTAGE_LEVEL` attribute reflects the new value to which the voltage level has been set.

#### `AXKDC_ATTR_TRIGGERED_VOLTAGE_MODE`

Data Type: `ViInt32`

Description:

Values:

<code>AXKDC_VAL_TRIGGER_MODE_FIX</code>	0
<code>AXKDC_VAL_TRIGGER_MODE_STEP</code>	0
<code>AXKDC_VAL_TRIGGER_MODE_PULSE</code>	0
<code>AXKDC_VAL_TRIGGER_MODE_LIST</code>	0

#### `AXKDC_ATTR_VOLTAGE_LEVEL`

Data Type: `ViReal64`

Description: This channel-based attribute specifies the voltage level the DC power supply attempts to generate. The units are Volts.

#### `AXKDC_ATTR_VOLTAGE_MAXIMUM`

Data Type: `ViReal64`