

VI Reference

This section contains detailed descriptions of all the top-level VIs included with AppletVIEW.

Common Inputs and Outputs:



Connection RefNum In

the TCP/IP connection refnum representing the socket connection being used by the AppletVIEW VI's.



Error In

a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status:

TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.



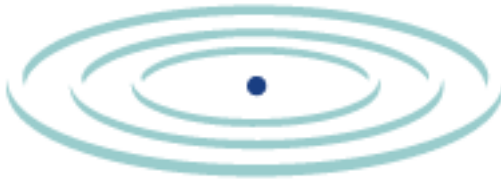
code:

the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.



source:

a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.



Connection RefNum Out

the TCP/IP connection refnum representing the socket connection being used by the AppletVIEW VI's.



Error Out

a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.



status:

TRUE if an error occurred, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.



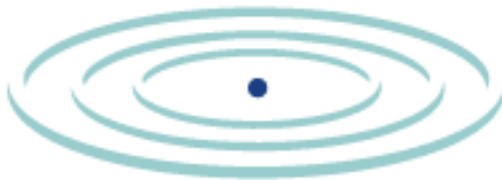
code:

the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.



source:

a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.



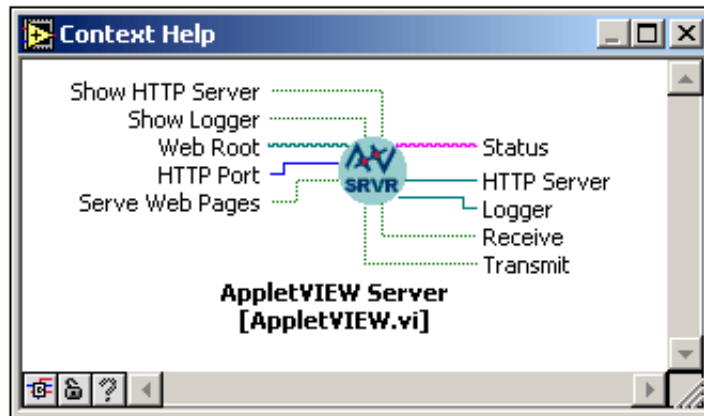
we connect all sorts of things to the web

13-3

AppletVIEW.IIb This is the main VI library containing the AppletVIEW Server and related VIs.

AppletVIEW.vi This is the main AppletVIEW VI, implementing the AppletVIEW Server.

Connector Pane



Controls

 Show HTTP Server

 Show Logger

 Web Root

 HTTP Port

 Serve Web Pages

Indicators

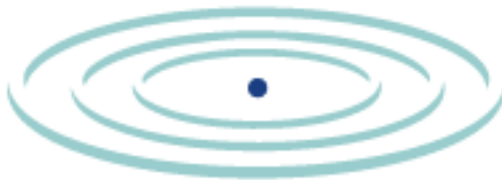
 Status

 HTTP Server

 Logger

 Receive

 Transmit

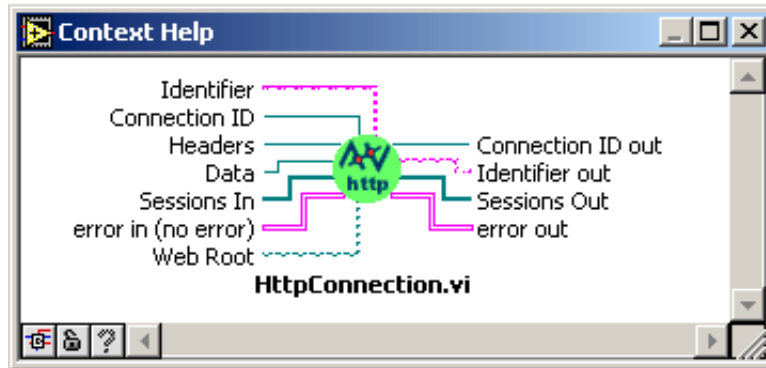


we connect all sorts of things to the web








13-4

HTTPConnection.vi An instance of this handles each HTTP request.





Connector Pane

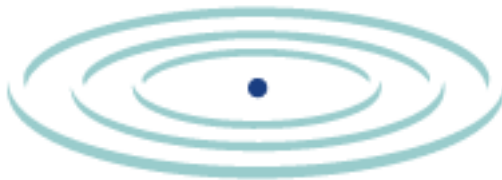


Controls

-  **Identifier**
-  **ConnectionID**
-  **Headers**
-  **Data**
-  **Sessions In**
-  **Error in (no error)**
-  **Web Root**

Indicators

-  **Connection ID out**
-  **Identifier Out**
-  **Sessions Out**
-  **Error out**

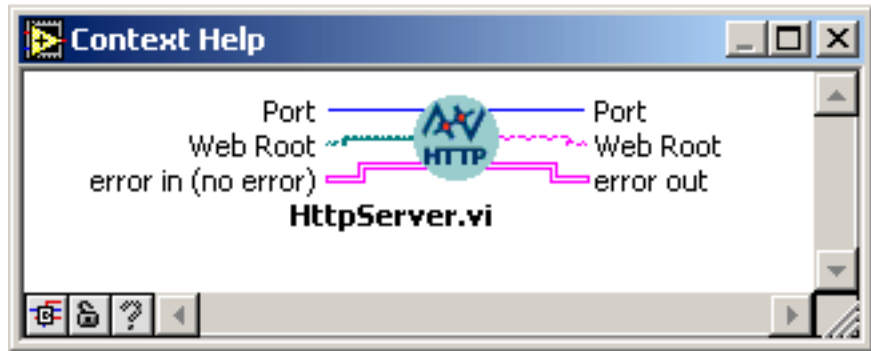


we connect all sorts of things to the web

13-5

HTTPServer.vi The AppletVIEW HTTP Server

Connector Pane



Controls

 Port

 Web Root

 Error in (no error)

Indicators

 Port

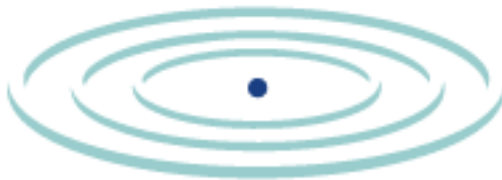
 Web Root

 Error Out

MimeTypes.vi Modify this VI to control how the HTTP Server returns different file types.

Connector Pane





we connect all sorts of things to the web

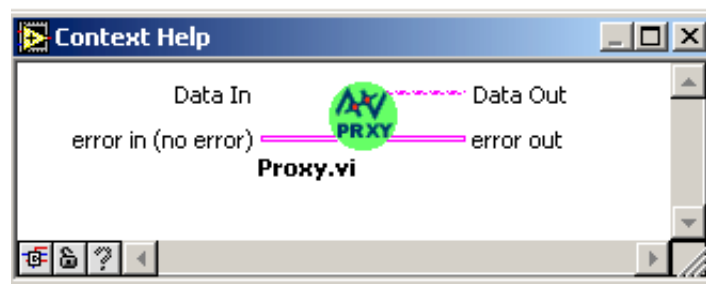
13-6

Controls  **Suffix**

Indicators  **MIME Type**

Proxy.vi Interfaces between a remote web client and an open VI.

Connector Pane

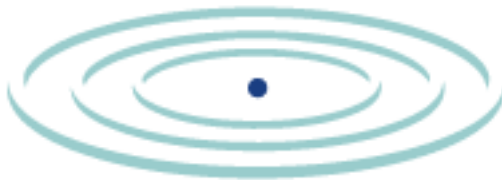


Controls  **Data In**

 **Error in (no error)**

 **Data Out**

 **Error out**

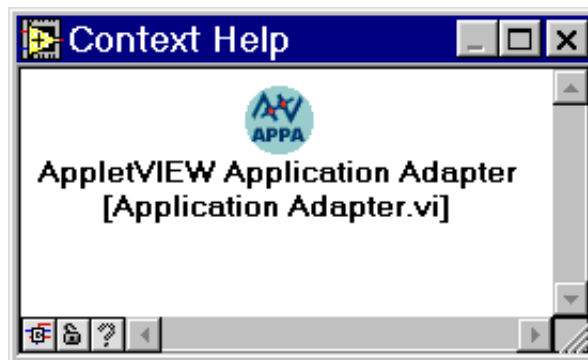


we connect all sorts of things to the web

13-7

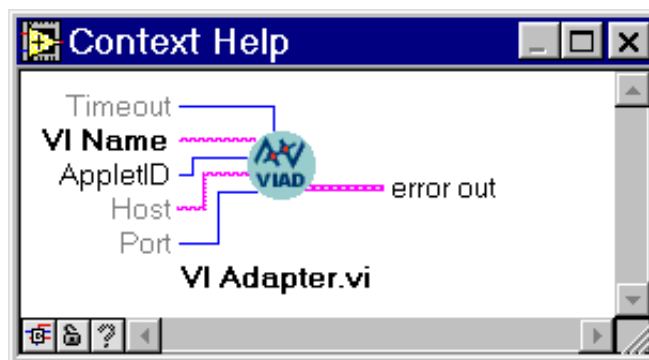
Application Adapter.vi

Connector Pane



VI Adapter.vi

Connector Pane



Controls

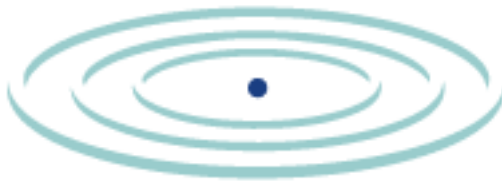
 **VI Name**

 **Applet ID**

 **Host** (optional)

 **Port** (optional)

 **Timeout** (optional)



we connect all sorts of things to the web

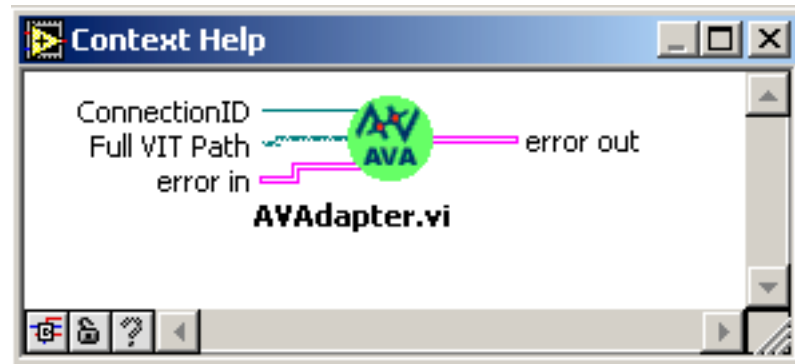
Indicators



error out

AVAdapter.vi

Connector Pane



Controls



ConnectionID



Full VIT Path

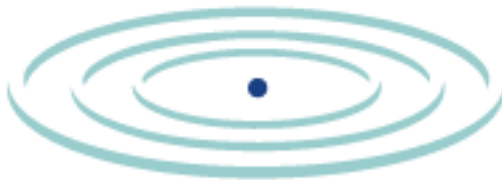


Error in

Indicators



Error out



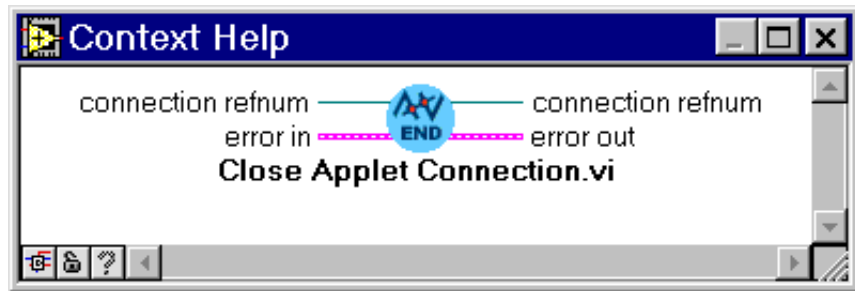
we connect all sorts of things to the web

13-9

AVCore.lib This is the main programming library for AppletVIEW.

Close Applet Connection.vi Closes the network connection with the applet. The client cannot re-connect after this VI is called, unless it reloads and a new "Wait for Applet Connection" is called.

Connector Pane



Controls

 **connection refnum in**

 **error in**

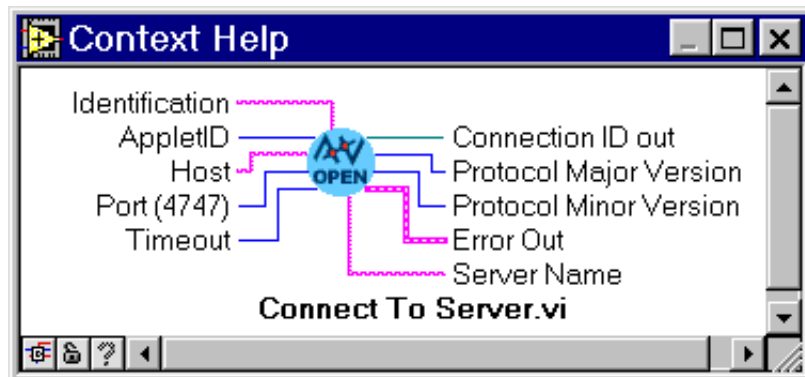
Indicators

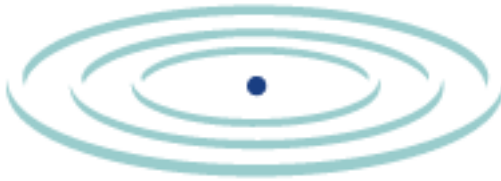
 **connection refnum out**

 **error out**

Connect To Server.vi This VI is used to open a communication session with the AppletVIEW Server. When used, your LabVIEW application initiates the communication session.

Connector Pane





we connect all sorts of things to the web

13-10

Controls



Identification



AppletID (1)



Host (localhost)

the name or IP address of the AppletVIEW Server



port (4747)

the port being listened on by the AppletVIEW Server



Timeout (60000)

amount of time to wait for a connection acknowledgement from the AppletVIEW Server before giving up

Indicators



ConnectionID out

the TCP/IP connection RefNum, suitable for feeding into another AppletVIEW VI if more than one connection is being used



Protocol Major Version

VITP major version being used by the AppletVIEW Server



Protocol Minor Version

VITP minor version being used by the AppletVIEW Server

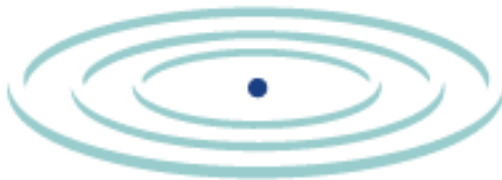


Error Out

any error that occurs during connection



Server Name



CreateAppletListener.vi

Creates a listener on the specified port. This VI is used together with Wait On Applet Connection to allow your LabVIEW application to wait for a client to initiate a communication session.

Connector Pane



Controls



port (4747)

the port to listen on for connections

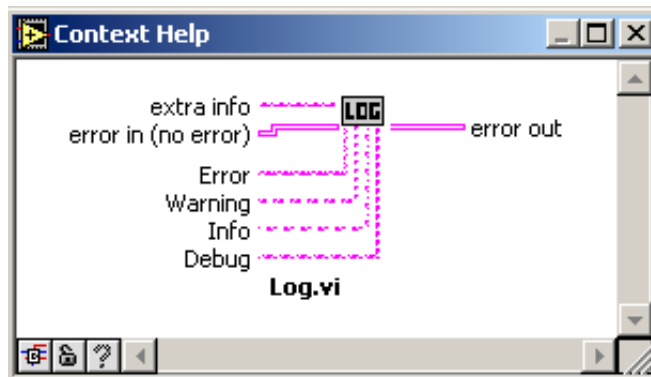
Indicators



error out

any error that occurs setting up the listener

Log.vi



Controls



Extra info



error in(no error)



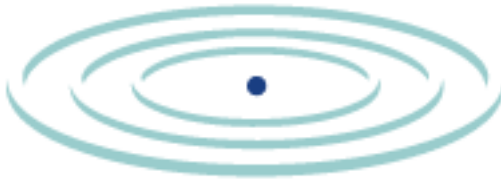
error



warning



info



we connect all sorts of things to the web

13-12

 **debug**

Indicators

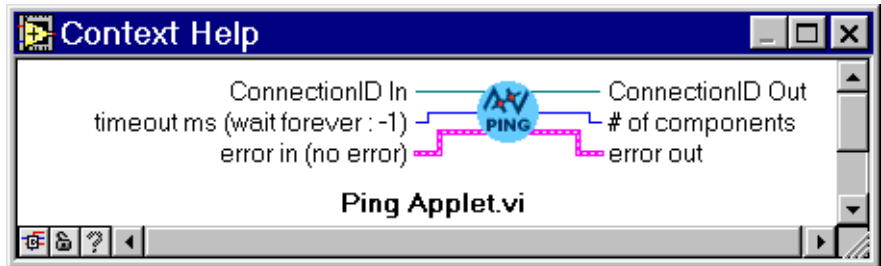
 **error out**

Ping Applet.vi

This VI checks to see if the applet is still connected.

- It returns the # of components on the applet.
- If no applet is connected, it returns error -1111

Connector Pane



Controls


 **ConnectionID In**

 **time-out ms (wait forever: -1)**

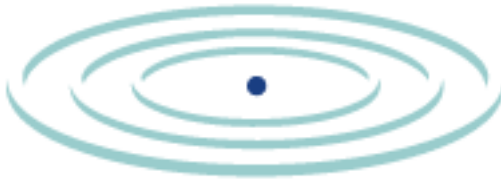
 **error in (no error)**

Indicators

 **ConnectionID Out**

 **# of components:**
the number of components on the applet

 **error out**



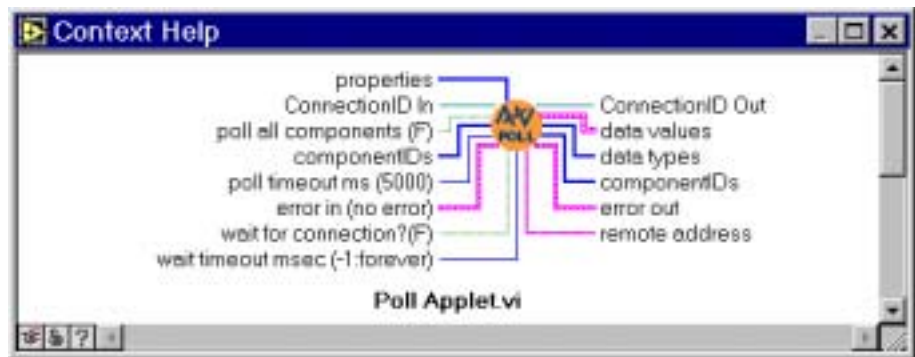
Poll Applet.vi Reads data from a Java applet by polling its component values. When the applet has been configured in poll-mode (not the default mode), use this VI to capture the changes in component values the client sends.

If "poll all components" is set to TRUE, all the componentIDs and their values are returned. If set to FALSE, you must specify an array of componentIDs to poll.

The data is returned as an array of clusters, each cluster containing the four datatypes. The data type that matches the componentID will contain a value.

If the VI was configured in event-mode, you should use Read Applet.vi instead to capture the changes in value the client sends.

Connector Pane



Controls



properties



ConnectionID In



poll all components (F):

(T) If TRUE (default), this VI will poll all the components on the applet. If you set this to TRUE, you do not need to wire the componentID array input.



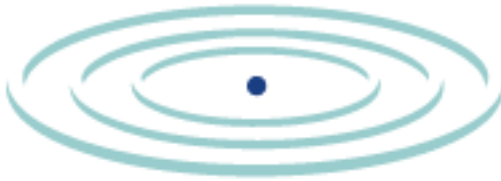
componentIDs:

an array of componentIDs that will be polled for values.



poll timeout ms (5000):

specifies the time-out for a poll operation in msec.



 **error in** (no error)

 **wait for connection?(F):**

(F) if TRUE, waits for a client to connect before initiating a poll operation. Set this to TRUE only if you did not already run Wait on Applet Connection.vi

 **wait timeout msec (-1:forever):**

this input is ignored if "wait for connection" is FALSE. Specifies the time in msec to wait for a client connection, or waits forever if set to -1.

Indicators

 **remote address:**

returns the remote IP address of the client, only if "wait for connection" was TRUE.

 **ConnectionID Out**

 **data values:**

returns an array of the data values from the polled components. The data values are in the same order as the array of componentIDs. Only one of the four data indicators in the cluster which comprises each array element contains data, according to the data type of the component polled.

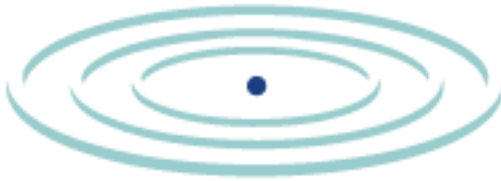
 **data types:**

returns an array of the data types of the polled components, in the same order as the array of componentIDs.

 **componentIDs:**

returns an array of the componentIDs whose values were successfully polled. Normally should be the same as the input array of componentIDs.

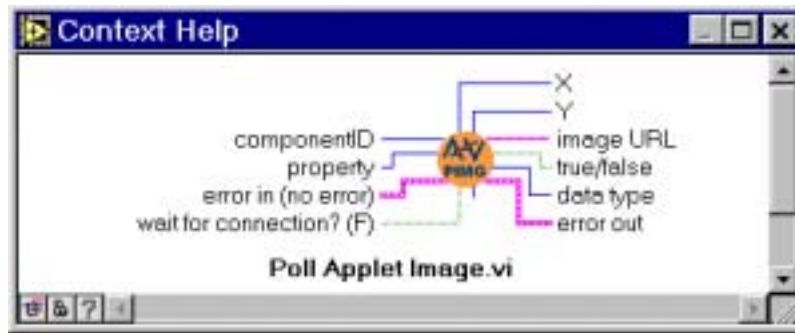
 **error out**



Poll Applet Image.vi

Poll Image allows you to poll an image component for specific properties.

Connector Pane



Controls



componentID

the componentID of the image you are polling.



property

list of properties that you can poll. Depending on the property you are polling, you will get the value to one of the 4 outputs (X [int32], Y [int32], imageURL [string], or true/false [Boolean]). The properties you can poll are:

- imageURL ("imageURL")
- minorImageURL ("imageURL")
- moveable ("true/false")
- location ("X" and "Y")
- limitMinX ("X")
- limitMinY ("Y")
- limitMaxX ("X")
- limitMaxY ("Y")
- sendDragEvents ("true/false")
- state ("true/false")

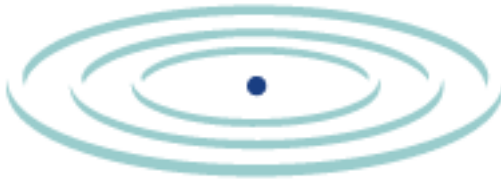


error in (no error)



wait for connection?(F)

(F) if TRUE, waits for a client to connect before initiating a read operation. Set this to TRUE only if you did not already run Wait on Applet Connection.vi.



Indicators



X returns

an X-coordinate in pixels if an appropriate property was polled.



Y returns

an Y-coordinate in pixels if an appropriate property was polled.



wait time-out ms (-1: forever)

this input is ignored if "wait for connection" is FALSE. Specifies the time in msec to wait for a client connection, or waits forever if set to -1.



image URL

returns the URL for either imageURL or minorImageURL, if one of these two was selected for polling.



true/false

returns a Boolean value when a Boolean property was polled.



data type

the data type of the component polled

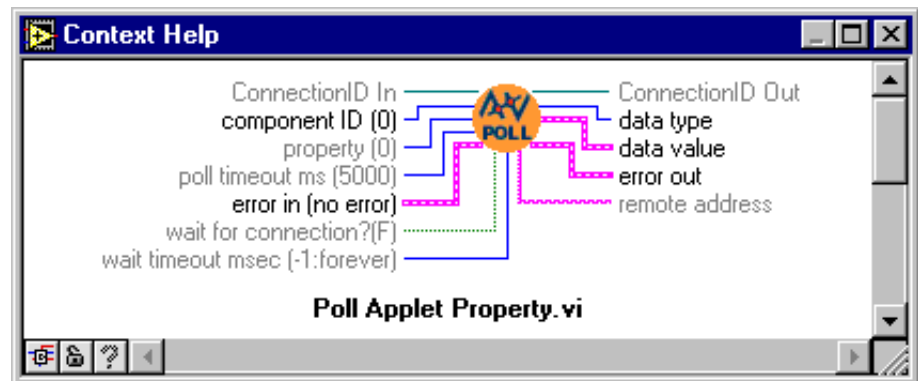


error out

Poll Applet Property.vi

Poll Applet Property allows you to query the applet for a specific component property.

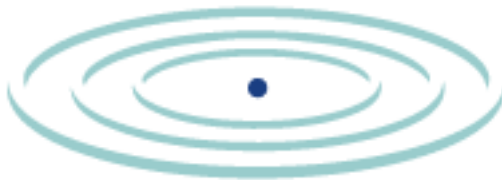
Connector Pane



Controls



ConnectionID In (optional)



we connect all sorts of things to the web

13-17

I16 component ID (0)

I8 property (0) (optional)

I32 poll timeout ms (5000) (optional)

E+ error in (no error)

TF wait for connection ? (F) (optional)

I32 wait timeout ms (-1: forever) (optional)

Indicators

D ConnectionID out (optional)

↔ data type

E+ data value

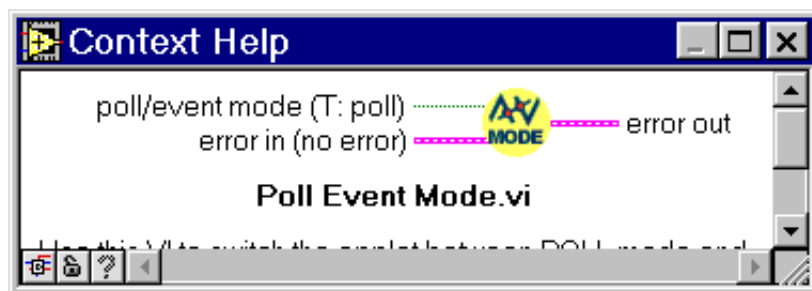
E+ error out

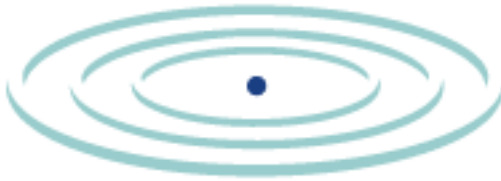
abc remote address (optional)

Poll/Event Mode.vi

Use this VI to switch the applet between POLL mode and EVENT mode. In POLL mode, the applet does not send any data when a client event occurs; rather, LabVIEW must poll the applet to find out the state of a components

Connector Pane





Controls

 **error in** (no error)

 **poll/event mode (T: poll):**

set to TRUE to put the applet in POLL mode. Set to FALSE to put it back in EVENT mode.

Indicators

 **error out**

Read Applet.vi

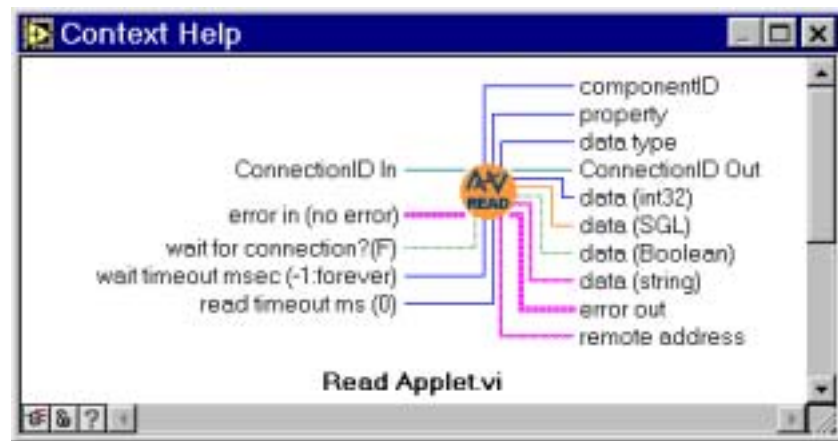
Reads data from a Java applet. When the applet has been configured in event-mode (default), use this VI to capture the changes in component values the client sends.

Use the "componentID" and/or the "data type" outputs of this VI to figure out which of the four data outputs (int32, SGL, string, Boolean) to read.

This VI normally exits immediately if there is no data available, so you often will want to place it in a While Loop to catch the data from the applet.

If the applet was configured in poll-mode, then you must use the Poll Applet.vi instead of Read Applet.vi

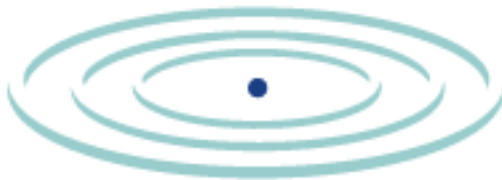
Connector Pane



Controls

 **ConnectionID In**

 **error in** (no error)



wait for connection?(F):

(F) if TRUE, waits for a client to connect before initiating a read operation. Set this to TRUE only if you did not already run Wait on Applet Connection.vi



read timeout ms (0)

specifies the time-out for a Read operation in msec. Normally it should be 0 so the VI will terminate immediately if there is no data available.



wait timeout msec (-1:forever):

this input is ignored if "wait for connection" is FALSE. Specifies the time in msec to wait for a client connection; or waits forever if set to -1.

Indicators



componentID:

returns the componentID of the component value changed sent by the client. Returns -1 if no data was sent



Property



data type:

an enumerated type that returns the data type from the component read. Possible values are:

- [no data]
- [int32]
- [SGL]
- [string]
- [Boolean]

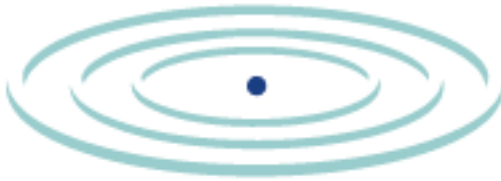


ConnectionID Out



data (int32):

if the component read was of type int32, the value will be returned here.



data (SGL):

if the component read was of type SGL, the value will be returned here.



data (Boolean):

if the component read was of type Boolean, the value will be returned here.



data (string):

if the component read was of type string, the value will be returned here.



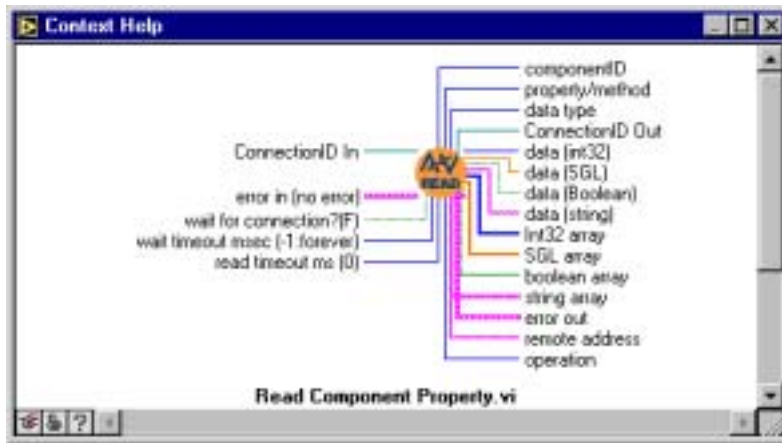
error out



remote address

Read Component Property.vi

Connector Pane



Controls



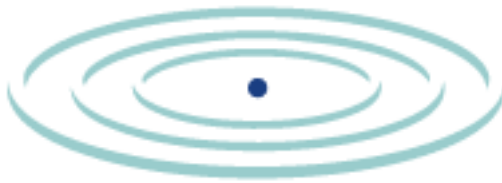
ConnectionID In



error in (no error)



wait for connection?(F):



we connect all sorts of things to the web

13-21

Indicators

[I32] wait timeout msec (-1:forever):

[I32] read timeout ms (0)

[I16] componentID:

[U8] property/method

[↔] data type:

[B] ConnectionID Out

[I32] data (int32):

[SGL] data (SGL):

[TF] data (Boolean):

[abc] data (string)

[I32] int32 array

[SGL] SGL array

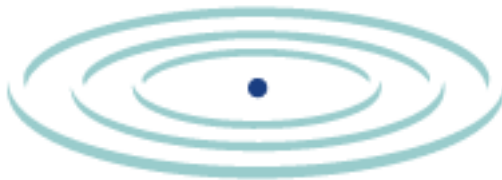
[TF] boolean array

[abc] string array

[Err] error out

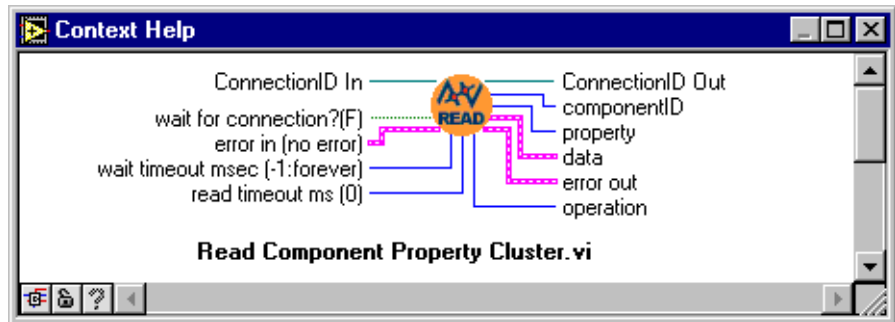
[abc] remote address

[↔] operation



Read Component Property Cluster.vi

Connector Pane



Controls



ConnectionID In



wait for connection?(F)

F) if TRUE, waits for a client to connect before initiating a read operation. Set this to TRUE only if you did not already run Wait on Applet Connection.vi



error in (no error)



wait time-out msec (-1:forever):



read timeout ms (0):

Indicators



connectionID Out



componentID:



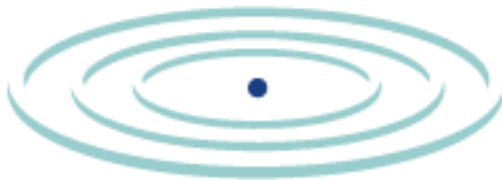
property



data



error out



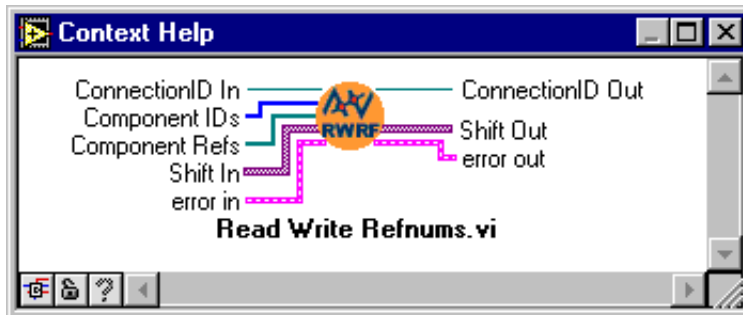
we connect all sorts of things to the web

13-23

 **Operation**

Read Write Refnums.vi

Connector Pane




Controls

 **ConnectionID In**

 **Component IDs**

 **Component Refs**

 **Shift In**

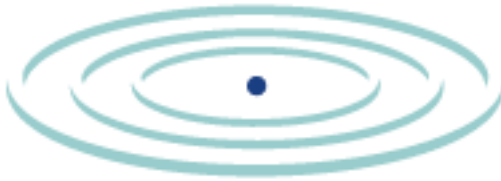
 **error in**

Indicators

 **ConnectionID Out**

 **Shift Out**

 **error out**

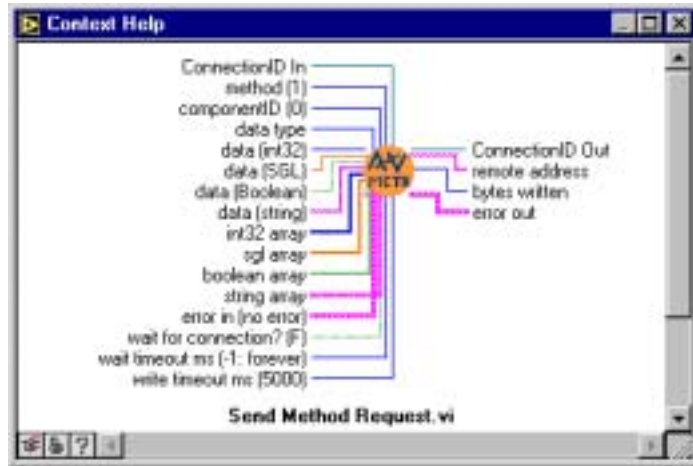


we connect all sorts of things to the web

13-24

Send Method Request.vi

Connector Pane



Controls



ConnectionID In



method (1)



Component ID (0)



data type:



data (int32):



data (SGL)



data (Boolean)



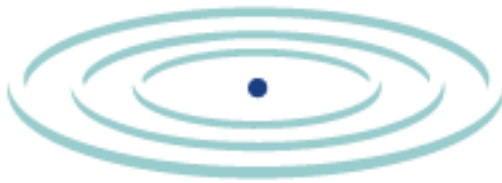
data (string):



int32 array



sgl array



[TF] boolean array

[abc] string array

[Err] error in (no error)

[TF] wait for connection ? (F)

[I32] wait timeout msec (-1:forever)

[I32] write timeout ms (5000)

Indicators

[D] connectionID out

[abc] remote address

[I32] bytes within

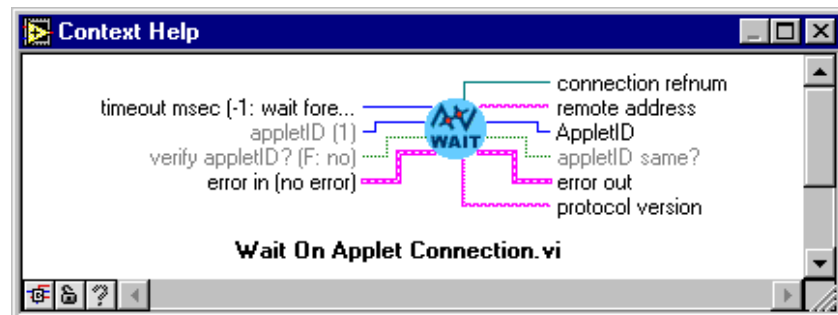
[Err] error out

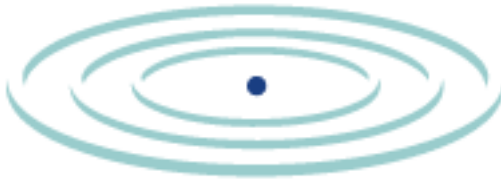
Wait On Applet Connection.vi

This VI is a companion to the Create Applet Listener VI. It waits for an client to connect and initiate a communication session. Specify the time-out in msec to wait, or -1 to wait forever.

You can either run this VI to wait every time a new client connection is made, or omit it and specify the "Read Applet" and "Write Applet" VIs to wait for the client connection.

Connector Pane





Controls



time-out msec (-1: wait forever):

the time-out in milliseconds before this VI gives up on waiting for a client connection. Set to -1 (default) to wait indefinitely.



appletID (1): (optional)

use this input only if you wish to verify the appletID of the incoming client. This value, combined with "verify appletID?" set to TRUE, will check the client's appletID.



verify appletID? (F: no): (optional)

set this to TRUE (default is FALSE) if you wish to verify the incoming client's appletID. If set to TRUE, the client's appletID must match the "appletID" value, otherwise an error is returned.



error in (no error)

Indicators



Connection refnum



protocol version:

returns the protocol version of AppletVIEW that this applet is using.



remote address:

returns the remote IP address of the applet.



AppletID

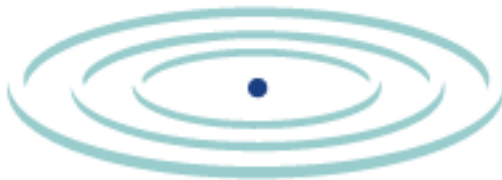


appletID same?: (optional)

is TRUE, only if: "verify appletID?" was set to TRUE, AND the appletID of the connected applet matches the appletID input of this VI



error out

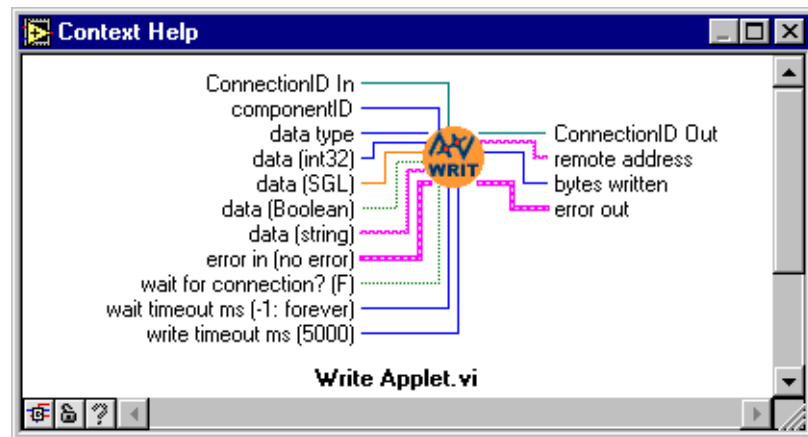


Write Applet.vi

Writes data to a component on a Java applet. The following inputs MUST be wired:

- "componentID" should match the componentID you are writing to.
- one and ONLY one of the "data" inputs. Be sure you are wiring to the data input whose types matches the data type of the applet component.

Connector Pane



Controls



ConnectionID In



componentID:

wire the componentID of the applet component you are sending data to.



data type



data (int32):

if the component you are writing to is of type int32, use this input.



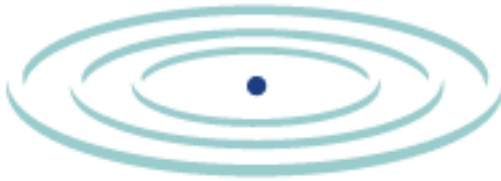
data (SGL):

if the component you are writing to is of type SGL, use this input.



data (Boolean):

if the component you are writing to is of type Boolean, use this input.



data (string):

if the component you are writing to is of type string, use this input.



error in (no error)



wait for connection? (F)

(F) if TRUE, waits for a client to connect before initiating a read operation. Set this to TRUE only if you did not already run Wait on Applet Connection.vi



wait timeout ms (-1: forever):

this input is ignored if "wait for connection" is FALSE. Specifies the time in msec to wait for a client connection, or waits forever if set to -1.



write timeout ms (5000):

the time in msec to continue to attempt a write to the applet before giving up.

Indicators



ConnectionID Out



remote address

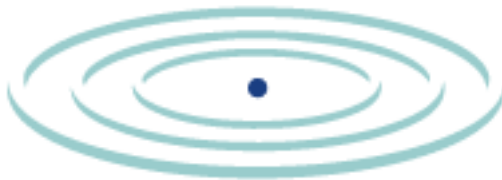
returns the remote IP address of the client, only if "wait for connection" was TRUE.



bytes written

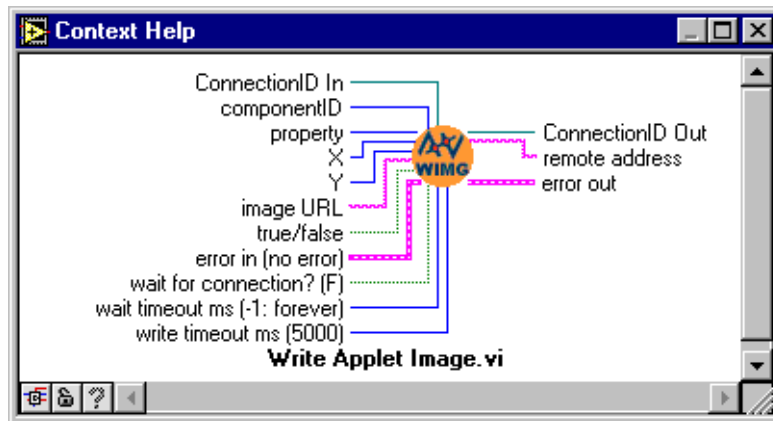


error out



Write Applet Image.vi

Connector Pane



Controls



ConnectionID In



property:

a list of properties that you can change. Depending on the property you are modifying, you will pass the value to one of the 4 inputs (X [int32], Y [int32], imageURL [string], or true/false [Boolean]). The properties you can modify and the value input in () are:

- imageURL ("imageURL")
- minorImageURL ("imageURL")
- moveable ("true/false")
- location ("X" and "Y")
- limitMinX ("X")
- limitMinY ("Y")
- limitMaxX ("X")
- limitMaxY ("Y")
- sendDragEvents ("true/false")
- state ("true/false")



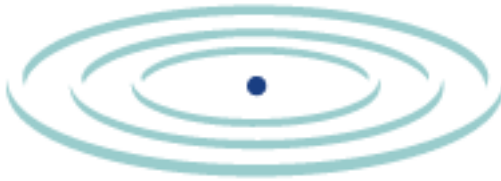
componentID:

the componentID of the image you are writing to.



X:

the X-value for the following properties:



- location
- LimitMinX
- LimitMaxX



Y:

the Y-value for the following properties:

- location
- LimitMinY
- LimitMaxY



image URL:

a string with the relative image URL (relative to the .jvi file), when writing the following properties:

- imageURL
- minorImageURL



true/false:

a Boolean input for setting Boolean image properties:

- state
- moveable
- endDragEvents



error in (no error)



wait for connection?:

(F) if TRUE, waits for a client to connect before initiating a read operation. Set this to TRUE only if you did not already run Wait on Applet Connection.vi.



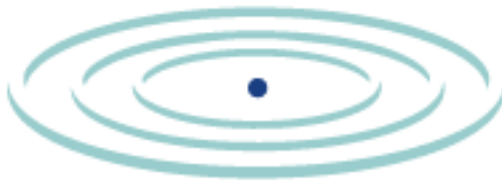
wait timeout ms (-1: forever):

this input is ignored if "wait for connection" is FALSE. Specifies the time in msec to wait for a client connection; or waits forever if set to -1.



write timeout ms (5000):

the time in msec to continue to attempt a write to the applet before giving up.



Indicators



ConnectionID Out



remote address:

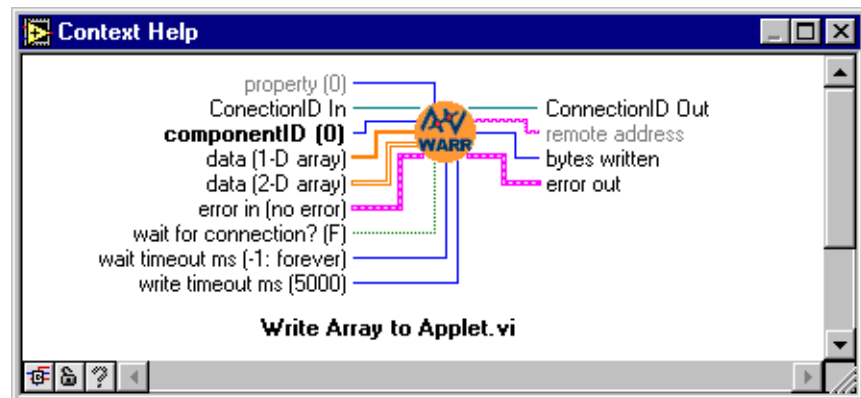
returns the remote IP address of the client, only if "wait for connection" was TRUE.



error out

Write Array to Applet.vi

Connector Pane



Controls



property (0) (optional)



ConnectionID In



componentID (0)



data (1-D array)



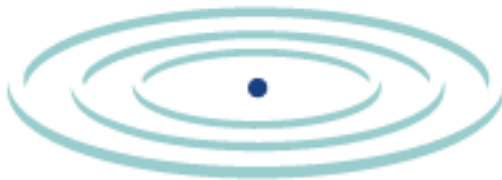
data (2-D array)



error in (no error)



wait for connection?(F)



we connect all sorts of things to the web

13-32

 wait timeout ms (-1: forever)

 write timeout ms (5000)

Indicators

 ConnectionID Out

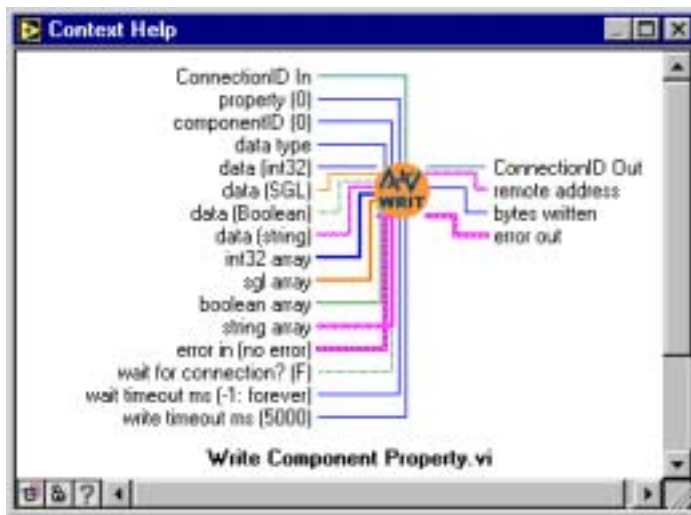
 remote address (optional)

 bytes written

 error out

Write Component Property.vi

Connector Pane



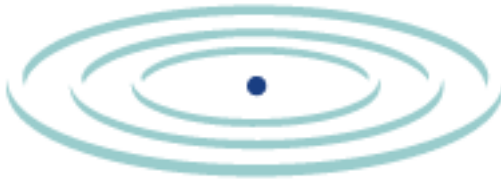
Controls

 ConnectionID In

 property (0)

 componentID (0)

 data type



we connect all sorts of things to the web

13-33

[I32] data (int32):

[SGL] data (SGL):

[TF] data (Boolean)

[abc] data (string):

[I32] int32 array

[SGL] sgl array

[TF] boolean array

[abc] string array

[Err] error in (no error)

[TF] wait for connection?(F)

[I32] wait timeout ms (-1: forever)

[I32] write timeout ms (5000)

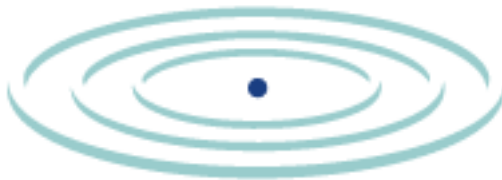
Indicators

[D] ConnectionID Out

[abc] remote address

[I32] bytes written

[Err] error out

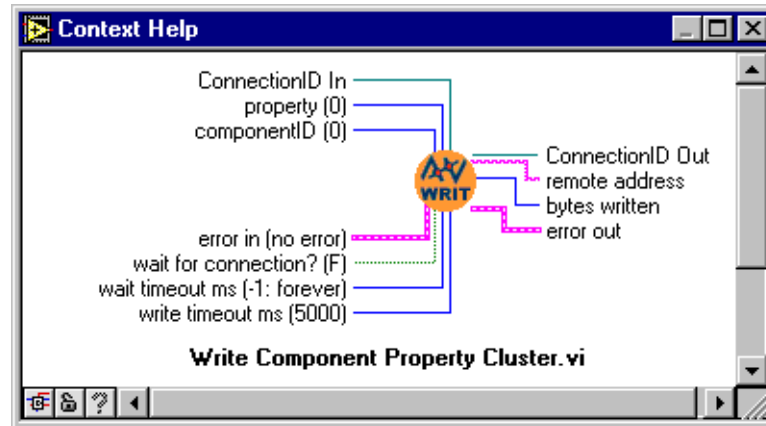


we connect all sorts of things to the web

13-34

Write Component Property Cluster.vi

Connector Pane



Controls



ConnectionID In



property (0)



componentID (0)



error in (no error)



wait for connection?(F)



wait timeout ms (-1: forever)



write timeout ms (5000)

Indicators



ConnectionID Out



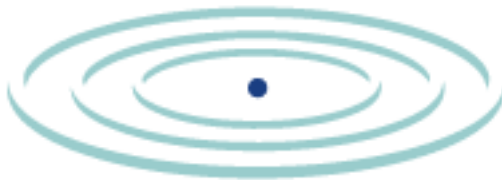
remote address



bytes written



error out

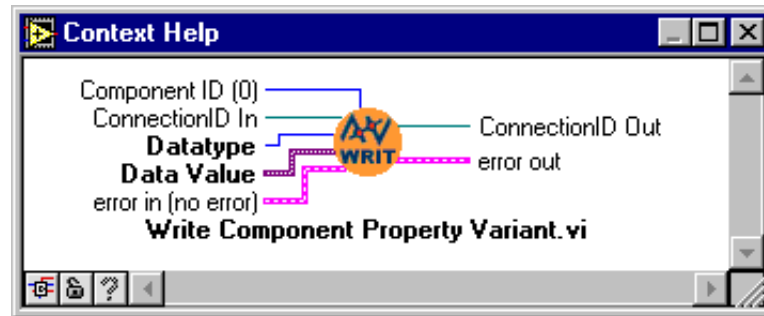


we connect all sorts of things to the web

13-35

Write Component Property Variant.vi

Connector Pane



Controls

 componentID (0)

 ConnectionID In

 datatype

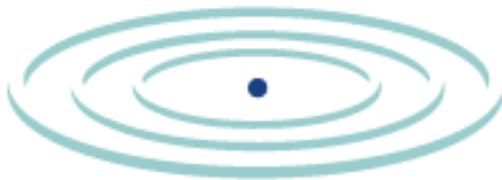
 data value

 error in (no error)

Indicators

 ConnectionID Out

 error out

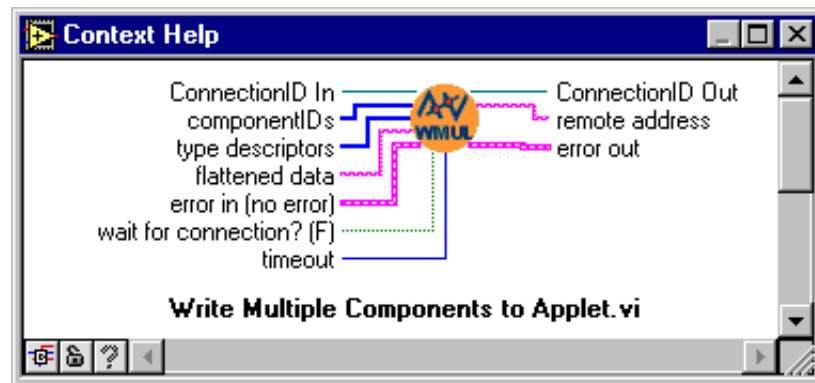


Write Multiple Components to Applet.vi

This function updates multiple components on an applet at once. To wire the inputs:

- Build a cluster of all the components in LabVIEW you are sending to the applet (using the Build Cluster function).
- Wire the cluster to a "Flatten To String" function (under Advanced>Data Manipulation palettes).
- Wire the outputs of "Flatten To String" to the type descriptors and string flattened data.
- Wire an array of component IDs-- MAKE SURE THE ORDER OF THE COMPONENT IDS MATCHES THE DATA CLUSTER ORDER!

Connector Pane



Controls



ConnectionID In



componentIDs:

an array of the componentIDs you are writing to.



type descriptors:

wire the output "type string" from the "Flatten To String" function here. This is an array of the type descriptors for the data.



flattened data:

Wire the output "Flattened Data" of the "Flatten to String" function here. This is all the data sent to the components.

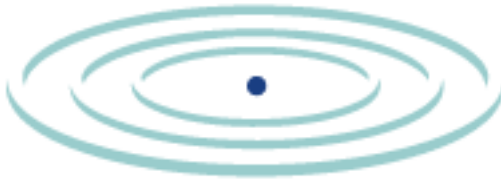


error in (no error)







wait for connection?:

(F) if TRUE, calls "Wait on Applet Connection" before proceeding.



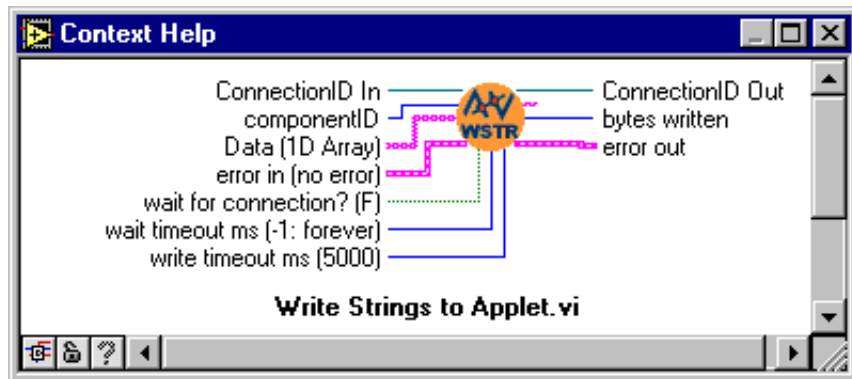
we connect all sorts of things to the web








13-37

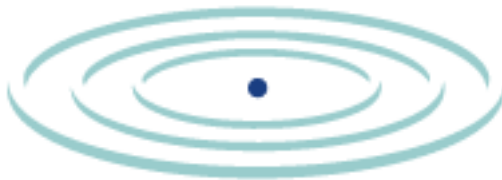
- Indicators**
-  **timeout**
the time-out for the "wait on connection"
 -  **ConnectionID Out**
 -  **remote address**
 -  **error out**

Write Strings to Applet.vi

Connector Pane



- Controls**
-  **ConnectionID In**
 -  **ComponentID**
 -  **Data (1D Array)**
 -  **error in (no error)**
 -  **wait for connection? (F)**
 -  **wait timeout ms (-1:forever)**
 -  **write timeout ms (5000)**



we connect all sorts of things to the web

13-38

Indicators



ConnectionID Out



remote address



bytes written



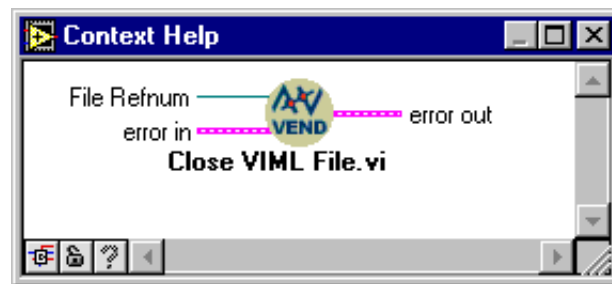
error out

VIML.IIb

These VIs are for generating VIML descriptions of front panels.

Close VIML File.vi

Connector Pane



Controls



file refnum

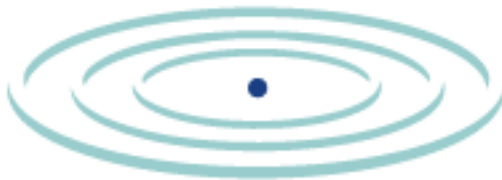


error in

Indicators



error out

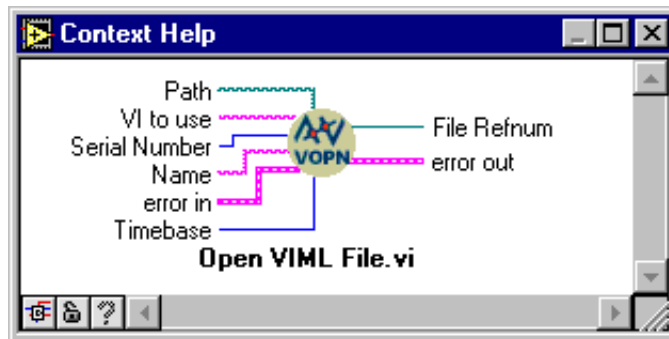


we connect all sorts of things to the web

13-39

Open VIML File.vi

Connector Pane



Controls



path



VI to use



Serial Number



name



error in



time base

Indicators



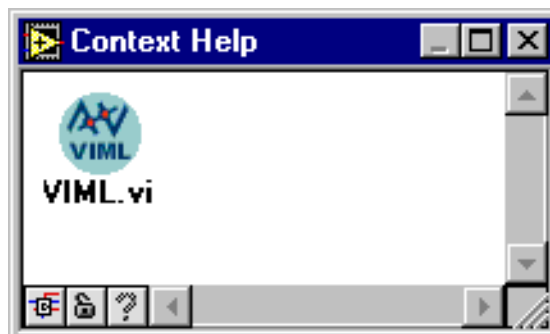
file refnum

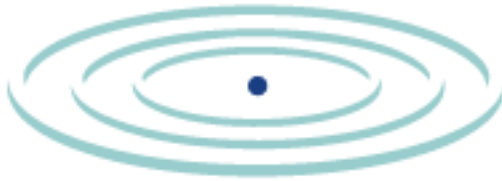


error out

VIML.vi

Connector Pane

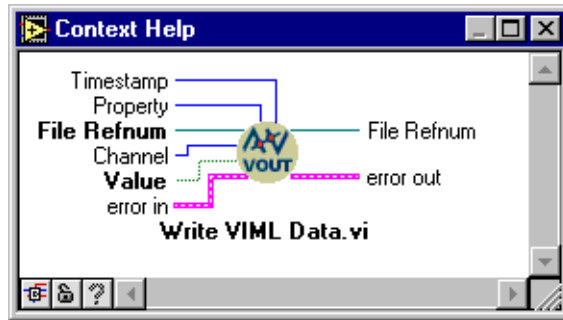




we connect all sorts of things to the web

Write VIML Data.vi

Connector Pane




Controls


 time stamp

 property

 file refnum

 channel

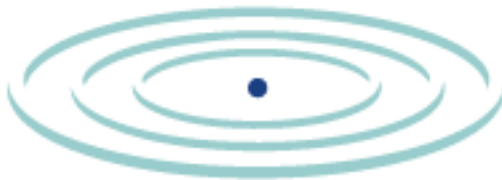
 value

 error in

Indicators

 file refnum

 error out

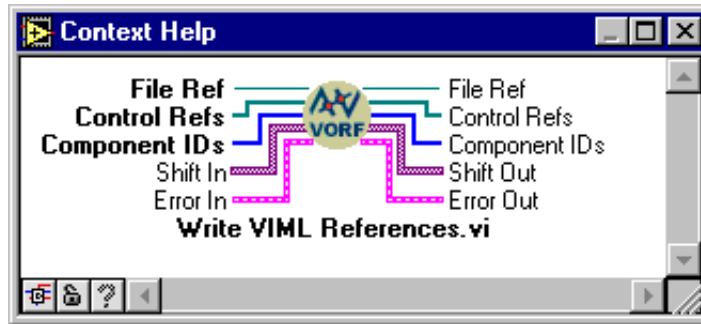


we connect all sorts of things to the web

13-41

Write VIML References.vi

Connector Pane



Controls



file ref



control refs



component ID's



Shift in



error in

Indicators



file ref



control refs



component ID's



Shift out



error out