

**SKICMP**

# **ActiveX Control**

**ICMP ActiveX Control for Microsoft® Windows™  
Copyright ©2001-2002 by Magneto Software, Inc.  
All rights reserved**

1	ICMP Overview .....	3
1.1	Introduction .....	3
1.2	Usage .....	3
1.3	Property Summary .....	4
1.4	Event Summary .....	4
1.5	Method Summary .....	4
1.6	Error codes .....	5
2	Properties .....	6
2.1	optCount .....	6
2.2	optDoNotFragment .....	7
2.3	optInterval .....	8
2.4	optRecordRoute .....	9
2.5	optResolve .....	10
2.6	optSize .....	11
2.7	optTimeout .....	12
2.8	optTimestamp .....	13
2.9	optTOS .....	14
2.10	optTraceRoute .....	15
2.11	optTraceRouteMaxHopsNumber .....	16
2.12	optTTL .....	17
3	Events .....	18
3.1	IcmpCompleted .....	18
3.2	IcmpReplyReceived .....	19
3.3	PingCompleted .....	20
3.4	PingReplyReceived .....	21
3.4	TraceRouteCompleted .....	22
3.5	TraceRouteReplyReceived .....	23
4	Methods .....	24
4.1	AboutBox .....	24
4.2	AddDestinationAddressEntry .....	25
4.3	CountDestinationAddressEntries .....	26
4.4	DeleteDestinationAddressEntry .....	27
4.5	GetAnyActiveDestinationAddress .....	28
4.6	GetDestinationAddressByIndex .....	29
4.7	GetNextActiveDestinationAddress .....	30
4.8	ICMPReset .....	31
4.9	ICMPSEndEchoRequest .....	32
4.10	ImportDestinationAddressList .....	33
4.11	PingGetReply .....	34
4.12	PingReset .....	35
4.13	ResetICMPSettings .....	36
4.14	TracerouteGetReply .....	37
4.15	TraceRouteReset .....	38

# 1 ICMP Overview

## 1.1 Introduction

The Magneto Software ICMP (Internet Control Message Protocol) ActiveX control (SKICMP.OCX) allows developers to integrate the ICMP protocol message sending capability into their applications without making calls directly to a Dynamic Link Library (DLL).

ICMP supports packets containing error, control, and informational messages.

The Magneto Software ICMP custom control takes care of creating an ICMP handle and sending the ICMP message to the desired host. Reply sent from the host is decoded and necessary information is passed to the application.

All functionality normally associated with an ICMP library is taken care of by setting properties and by handling events.

Properties can be changed at run time. Events occur to inform the application of errors and changes in the connection status and to deliver the incoming data.

Applications, which call a DLL directly are either notified of errors, connection changes and incoming data by an upcall posted by the library to the applications window or by making polling calls to the library at frequent intervals.

Some environments, such as Visual Basic, are not designed for either of these two methods. Using a custom control in an application frees the developer from handling upcalls or polling. Instead, events fired by the control deliver incoming data, errors and status changes directly to the application.

## 1.2 Usage

To change SKICMP.OCX from ping mode to traceroute mode the only property needs to change – **optTraceRoute** right before using **ICMPSEndEchoRequest**.

It is set to 0 for the ping requests, and set to non-zero for traceroute requests.

Ping mode is used to send ICMP echo requests to an IP address, and wait for ICMP echo responses. Ping reports on the number of responses received and the time interval between sending the request and receiving the response.

Traceroute works by sending ICMP echo requests to an

IP address, while incrementing the time-to-live (TTL) field in the IP header by one starting at 1, and analyzing ICMP errors that get returned. Each succeeding echo request should get one hop further into the network before the TTL field reaches 0 and an ICMP Time exceeded error is returned by the router attempting to forward it. Traceroute simply prints out an ordered list of the routers in the path that returned these error messages.

### ***1.3 Property Summary***

#### **optCount**

Number of echo requests to send.

Ping the specified host until stopped if set to 0

#### **optDoNotFragment**

Set don't fragment flag in packet

#### **optInterval**

Interval in seconds between two packets in Ping mode

#### **optRecordRoute**

Record route for count hops.

#### **optResolve**

Resolve addresses to hostnames

#### **optSize**

Send buffer size

#### **optTimeout**

Timeout in milliseconds to wait for each reply

#### **optTimestamp**

Timestamp for count hops

#### **optTOS**

Type Of Service

#### **optTraceRoute**

Mode (Ping or Traceroute)

#### **optTraceRouteMaxHopsNumber**

Maximum number of hops to search for target

#### **optTTL**

Time To Live

### ***1.4 Event Summary***

#### **IcmpCompleted**

Indicates that SKICMP.OCX has stopped processing ICMP replies.

#### **IcmpReplyReceived**

ICMP echo reply is received (Ping or Traceroute).

#### **PingCompleted**

Indicates that SKICMP.OCX has stopped processing Ping replies.

#### **PingReplyReceived**

Ping echo reply is received.

#### **TraceRouteCompleted**

Indicates that SKICMP.OCX has stopped processing Traceroute replies.

#### **TraceRouteReplyReceived**

Traceroute echo reply is received.

### ***1.5 Method Summary***

#### **AboutBox**

Display a dialog box with SKICMP activeX control license and version information.

#### **AddDestinationAddressEntry**

Adds entry to the destination address list (DAL).

#### **CountDestinationAddressEntries**

Counts entries in DAL

#### **DeleteDestinationAddressByIndex**

Deletes entry from DAL by index

#### **GetAnyActiveDestinationEntries**

Enumerates through the DAL entries and trying to send an ICMP packets to the remote hosts specified in the DAL.

**[GetDestinationAddressByIndex](#)**

Returns the DAL entry information given the entry index.

**[GetNextActiveDestinationAddress](#)**

Returns the next DAL entry information given the previous entry index.

**[ICMPReset](#)**

Stop all ICMP messages (Ping and Traceroute).

**[ICMPSendEchoRequest](#)**

Send an ICMP Request

**[ImportDestinationAddressList](#)**

Imports DAL table from specified remote host. If the currently logged user

Is not logged in to the remote machine, this method will fail with error ERROR\_ACCESS\_VIOLATION

**[PingGetReply](#)**

Retrieves Ping echo reply.

**[PingReset](#)**

Stop Ping messages (Traceroute messages will not be affected).

**[ResetICMPSettings](#)**

Reset all ICMP settings back to default values.

**[TracerouteGetReply](#)**

Retrieves Traceroute echo reply.

**[TraceRouteReset](#)**

Stop Traceroute messages (Ping messages will not be affected).

## ***1.6 Error codes***

The following provides a complete listing of error codes returned by SKICMP ActiveX control.

IP_SUCCESS (0)	IP Success
IP_BUF_TOO_SMALL (11001)	IP Buffer Too Small
IP_DEST_NET_UNREACHABLE (11002)	IP Destination Net Unreachable
IP_DEST_HOST_UNREACHABLE (11003)	IP Destination Host Unreachable
IP_DEST_PROT_UNREACHABLE (11004)	IP Destination Protocol Unreachable
IP_DEST_PORT_UNREACHABLE (11005)	IP Destination Port Unreachable
IP_NO_RESOURCES (11006)	IP No Resources
IP_BAD_OPTION (11007)	IP Bad Option
IP_HW_ERROR (11008)	IP Hardware Error
IP_PACKET_TOO_BIG (11009)	IP Packet Too Big
IP_REQ_TIMED_OUT (11010)	IP Request Timed Out
IP_BAD_REQ (11011)	IP Bad Request
IP_BAD_ROUTE (11012)	IP Bad Route
IP_TTL_EXPIRED_TRANSIT (11013)	IP TimeToLive Expired Transit
IP_TTL_EXPIRED_REASSEM (11014)	IP TimeToLive Expired Reassembly
IP_TTL_EXPIRED_REASSEM (11015)	IP Parameter Problem
IP_SOURCE_QUENCH (11016)	IP Source Quench
IP_OPTION_TOO_BIG (11017)	IP Option Too Big
IP_BAD_DESTINATION (11018)	IP Bad Destination

## 2 Properties

### 2.1 *optCount*

#### **Summary**

Number of echo requests to send.

#### **Description**

This property specifies the number of times that you want to resend your requests if it times out. By default this value will be set to 4. If this value is 0, SKICMP.OCX will ping the specified host until stopped.

This property is of type short.

#### **VB Example**

```
Dim Count As Integer
```

```
Count = 4
```

```
SKICMP.optCount = Count
```

## ***2.2 optDoNotFragment***

### **Summary**

Set Don't Fragment flag in the IP header options to be sent.

### **Description**

This property specifies the flags field contained in the IP header of the packet to be sent.

This property is of type short.

0=MayFragment, 1=Don't Fragment

The default value for this property is 1, which means, don't fragment flag is set.

### **VB Example**

```
ICMP.optDoNotFragment = 1
```

## ***2.3 optInterval***

### **Summary**

Interval in seconds between two packets in Ping mode.

### **Description**

By default this value will be set to 1.

Is applicable only in Ping mode

This property is of type long.

### **VB Example**

```
Dim Interval As Integer
```

```
Interval = 3600 'send a packet every hour (3600 seconds)
```

```
ICMP.optInterval =Interval
```

## ***2.4 optRecordRoute***

### **Summary**

Record route for count hops.

### **Description**

By default this value will be set to 0.

This option has to be set to value between 0 and 9.

This property is of type short.

### **VB Example**

```
Dim RecordRoute As Integer
```

```
RecordRoute = 9
```

```
ICMP.optRecordRoute = RecordRoute
```

## ***2.5 optResolve***

### **Summary**

Resolve addresses to hostnames

### **Description**

This property will enable an automatic lookup of the host name by IP address. This is to be able to supply the user with the host name. If the user does not need the name of the replying host, and wishes to decrease the network traffic this option has to be turned off. By default this value will be set to 0 (turned off).

This property is of type short.

### **VB Example**

```
Dim Timestamp As Integer  
Resolve = 1  
ICMP.optResolve = Resolve
```

## ***2.6 optSize***

### **Summary**

Send buffer size.

### **Description**

This property specifies the send buffer size. By default this value will be set to 64.

This property is of type short.

### **VB Example**

```
Dim Size As Integer
```

```
Count = 128
```

```
SKICMP.optSize = Size
```

## ***2.7 optTimeout***

### **Summary**

Timeout value in milliseconds to wait for replies.

### **Description**

This property specifies the timeout value in milliseconds that is used to wait for a reply when a request packet is sent. The application must set this value before the request is sent. By default this value is set to 2 second. (2000 milliseconds)

This property is of type long.

### **VB Example**

```
Dim Time As Long
```

```
Time = 2000
```

```
SKICMP.optTimeout = Time
```

## ***2.8 optTimestamp***

### **Summary**

Timestamp for count hops.

### **Description**

By default this value will be set to 0.

This option has to be set to value between 0 and 4.

This property is of type short.

### **VB Example**

```
Dim Timestamp As Integer
```

```
Timestamp = 2
```

```
ICMP.optTimestamp = Timestamp
```

## 2.9 *optTOS*

### Summary

Type of service field in the IP header of the packet to be sent.

### Description

This property specifies the TOS value to be placed in the IP header of the packet to be sent. Type of service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence, which somehow treats high precedence traffic as more important than other traffic. The major choice is a three-way tradeoff between low-delay, high-reliability, and high throughput. This value is 8bit value and its syntax is given below.

Bits 0-2: Precedence.

Bit 3: 0 = Normal Delay, 1 = Low Delay.

Bit 4: 0 = Normal Throughput, 1 = High Throughput

Bit 5: 0 = Normal Reliability, 1 = High Reliability.

Bit 6-7: Reserved for future Use.

Precedence

111 – Network Control

110 – Internetwork Control

101 – CRITIC/ECP

100 – Flash Override

011 – Flash

010 – Immediate

001 – Priority

000 – Routine

0 is the default value for this property. If this property is used then it should be set in unsigned short as only LSB will be extracted and used. If more information is required please refer to RFC 791 on IP protocol header format and its usage. Possible range of values for this property is between 0 and 63.

### VB Example

SKICMP.*optTOS* = 0

With 0 as the default value every packet is set for Routine treatment with Normal Delay, Normal throughput, and Normal Reliability.

## ***2.10 optTraceRoute***

### **Summary**

Mode (Ping or Traceroute)

### **Description**

This property specifies the mode of the ICMP requests that will be issued after setting this option by calling `ICMPSEndEchoRequest()`.

This property value for Ping mode is 0, for Traceroute mode is 1.

Default value for this property is 0 (Ping mode).

### **VB Example**

```
SKICMP.optTraceRoute = 0      ' Ping mode
```

## ***2.11 optTraceRouteMaxHopsNumber***

### **Summary**

Maximum number of hops to search for target.

### **Description**

By default this value will be set to 30.

To use SKICMP.OCX in traceroute mode this option has to be set to value between 0 and 255.

This property should be initialized before calling the ICMPSendEchoRequest method.

This property is of type short.

### **VB Example**

```
Dim Timestamp As Integer
```

```
MaxHops = 60
```

```
ICMP.optTraceRouteMaxHopsNumber = MaxHops
```

## ***2.12 optTTL***

### **Summary**

Time to live field in the IP header to be sent.

### **Description**

This property specifies the TTL value to be put in the IP header options. TTL is decremented at every hop and is used to have an upper bound on the time that a datagram can spend in the Internet before getting to the destination. If TTL becomes 0, the datagram is discarded. The time is measured in units of seconds, but since every module that processes a datagram must decrease the

TTL by at least one even if it processes the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. Values used for this property are between 0 and 255 and it is of type short.

Default value for this property is 255.

### **VB Example**

SKICMP.optTTL = 255

## **3 Events**

### ***3.1 IcmpCompleted***

#### **Summary**

Indicates that SKICMP.OCX has stopped processing ICMP replies.

#### **Syntax**

IcmpCompleted()

#### **Description**

SKICMP.OCX allows for processing of multiple Ping and multiple Traceroute requests by the same program simultaneously. This event can have a high importance for the application presentation logic, like Start/Stop buttons and other.

#### **Parameters**

None.

## ***3.2 IcmpReplyReceived***

### **Summary**

ICMP echo reply is received (Ping or Traceroute).

### **Syntax**

`IcmpReplyReceived(BSTR strRemoteAddress, BSTR strMessage, short nStatus)`

### **Description**

The `IcmpReplyReceived` event occurs whenever a reply to an echo request is received.

### **Parameters**

*strRemoteAddress* is the name of the remote host that was queried,

*strMessage* is the string message representing the result of the operation and

*nStatus* is the return status of each individual reply. See section [1.6 Error Codes](#) the complete list of supported error codes.

### ***3.3 PingCompleted***

#### **Summary**

Indicates that SKICMP.OCX has stopped processing Ping replies.

#### **Syntax**

PingCompleted()

#### **Description**

SKICMP.OCX allows for processing of multiple Ping and multiple Traceroute requests by the same program simultaneously. This event can have a high importance for the application presentation logic, like Start/Stop buttons and other.

#### **Parameters**

None.

### 3.4 PingReplyReceived

#### Summary

Ping echo reply is received.

#### Syntax

```
PingReplyReceived(BSTR strRemoteAddress, long lStatus, BSTR strReplyingAddress,  
                  BSTR strResolvedReplyingAddress, short nDataSize, long lRoundTripTime,  
                  short nTTL, BSTR strCountHopsInfo)
```

#### Description

The PingReplyReceived event occurs whenever a Ping reply to a Ping request is received.

#### Parameters

*strRemoteAddress* is the name of the remote host that was queried,

*lStatus* is the return status of each individual reply. See section [1.6 Error Codes](#) the complete list of supported error codes,

*strReplyingAddress* is the name of the replying remote host,

*strResolvedReplyingAddress* is the resolved name of the replying remote host; this argument may have a *value* if *optResolve* is set to 1.

*nDataSize* is the Ping reply packet size,

*lRoundTripTime* is the round trip time in milliseconds,

*nTTL* is the Ping reply packet TTL value,

*strCountHopsInfo* is the route info, this argument may have route info if *optRecordRoute* or *optTimestamp* is greater than 0.

### ***3.4 TraceRouteCompleted***

#### **Summary**

Indicates that SKICMP.OCX has stopped processing Traceroute replies.

#### **Syntax**

TraceRouteCompleted()

#### **Description**

SKICMP.OCX allows for processing of multiple Ping and multiple Traceroute requests by the same program simultaneously. This event can have a high importance for the application presentation logic, like Start/Stop buttons and other.

#### **Parameters**

None.

### 3.5 TraceRouteReplyReceived

#### Summary

Traceroute echo reply is received.

#### Syntax

TracerouteReplyReceived(BSTR *strRemoteAddress*, long *lStatus*, BSTR *strReplyingAddress*,  
BSTR *strResolvedReplyingAddress*, short *nDataSize*, long *lRoundTripTime*,  
short *nTtl*, short *nHop*, short *nPacket*)

#### Description

The TracerouteReplyReceived event occurs whenever a Traceroute reply to a Traceroute request is received.

#### Parameters

*strRemoteAddress* is the name of the remote host that was queried,

*lStatus* is the return status of each individual reply. See section [1.6 Error Codes](#) the complete list of supported error codes,

*strReplyingAddress* is the name of the replying remote host,

*strResolvedReplyingAddress* is the resolved name of the replying remote host; this argument may have a *value* if optResolve is set to 1.

*nDataSize* is the Ping reply packet size,

*lRoundTripTime* is the round trip time in milliseconds,

*nTTL* is the Ping reply packet TTL value,

*nHop* is the hop index, starting from 0,

*nPacket* is the packet index, starting from 1.

## **4 Methods**

### ***4.1 AboutBox***

#### **Summary**

Display a dialog box with SKICMP activeX control license and version information.

#### **Syntax**

```
void AboutBox();
```

#### **Description**

This method could be used to display version license information or to register SKICMP.OCX control.

#### **Parameters**

None.

## ***4.2 AddDestinationAddressEntry***

### **Summary**

Adds an entry to the destination address list (DAL).

### **Syntax**

long AddDestinationAddressEntry(BSTR *bstrDestinationAddress*)

### **Description**

It returns a long, which is set to 0 (ERROR\_SUCCESS) if the method is successfully executed, otherwise it will be set to the error code.

### **Parameters**

*bstrDestinationAddress* is the name of the remote host to be added to the DAL.

### ***4.3 CountDestinationAddressEntries***

#### **Summary**

Counts all entries in the destination address list (DAL).

#### **Syntax**

short CountDestinationAddressEntries (BSTR *bstrDestinationAddress*)

#### **Description**

It returns a number of entries in the DAL.

If calls fails it returns 0.

#### **Parameters**

None.

#### ***4.4 DeleteDestinationAddressEntry***

##### **Summary**

Deletes an entry from the destination address list (DAL) given the entry name.

##### **Syntax**

```
void DeleteDestinationAddressEntry(BSTR bstrDestinationAddress)
```

##### **Parameters**

*bstrDestinationAddress* is the name of the remote host to be deleted from the DAL.

## ***4.5 GetAnyActiveDestinationAddress***

### **Summary**

Enumerates through the destination address list (DAL) entries while querying them with ICMP echo requests.

### **Syntax**

long GetAnyActiveDestinationAddress(BSTR\* *pBstrDestinationAddress*, short\* *p nIndex*)

### **Description**

It returns a long, which is set to 0 (ERROR\_SUCCESS) if the method is successfully executed and an active host found, otherwise it will be set to the error code.

If successful, *pBstrDestinationAddress* contains an active host name; *p nIndex* contains an index of this host name in the DAL.

### **Parameters**

*pBstrDestinationAddress* is a first active host name from the DAL,  
*p nIndex* is an index of this host (starting from 0).

## ***4.6 GetDestinationAddressByIndex***

### **Summary**

Returns the DAL entry information given the entry index.

### **Syntax**

long GetDestinationAddressByIndex(short *nIndex*, BSTR\* *pbstrDestinationAddress*)

### **Description**

It returns a long, which is set to 0 (ERROR\_SUCCESS) if the method is successfully executed and an active host found, otherwise it will be set to the error code.

If successful, *pbstrDestinationAddress* contains host name from DAL that corresponds to *nIndex*.

### **Parameters**

*nIndex* is an ordinal index from the destination address list

*bstrDestinationAddress* is a host name from destination address list located at *nIndex*.

## ***4.7 GetNextActiveDestinationAddress***

### **Summary**

Returns the next DAL entry information given the previous entry index.

### **Syntax**

long GetNextActiveDestinationAddress(short *nPrevIndex*, BSTR\* *pbstrDestinationAddress*, short\* *pnIndex*)

### **Description**

If *nPrevIndex* set to  $-1$ , *pbstrDestinationAddress* will be assigned to the first active entry found in DAL (similar to method GetAnyActiveDestinationAddress).

It returns a long, which is set to 0 (ERROR\_SUCCESS) if the method is successfully executed and an active host found, otherwise it will be set to the error code.

If successful, *pbstrDestinationAddress* contains the first active host name from DAL, *pnIndex* contains an index of this host in the DAL. If *pnIndex* is set to  $-1$  there are no active entries available.

## ***4.8 ICMPReset***

### **Summary**

Terminate pinging

### **Syntax**

void ICMPReset (void)

### **Description**

The ICMPReset method terminates the ping process.

### **Parameters**

None.

## ***4.9 ICMPSendEchoRequest***

### **Summary**

Send an ICMP request.

### **Syntax**

long ICMPSendEchoRequest (BSTR *bstrDestinationAddress*)

### **Description**

The ICMPSendEchoRequest method sends an ICMP request to the host.

Before calling this method all other properties should be properly initialized.

It returns a long, which is set to 0 (ERROR\_SUCCESS) if the method is successfully executed, otherwise it will be set to the error code from section [1.6 Error Codes](#).

It takes one parameter, *bstrDestinationAddress*, the name of the host you want to send request to.

### **Parameters**

*bstrDestinationAddress* is the name of the remote host to query.

## ***4.10 ImportDestinationAddressList***

### **Summary**

Imports DAL table from specified remote host. If the currently logged user is not logged in to the remote machine, this method will fail with error.

### **Syntax**

long ImportDestinationAddressList(BSTR *bstrMachineName*, short *nOverwrite*)

### **Description**

Imports DAL table from specified remote host. If the currently logged user is not logged in to the remote machine, this method will fail with error ERROR\_ACCESS\_VIOLATION. If *nOverwrite* set to 1 local DAL will be overwritten, otherwise imported entries will be added to it. Using this method local DAL list can be synchronized if necessary with some remote central DAL. The DAL list is stored in registry under *HKEY\_LOCAL\_MACHINE\SOFTWARE\MagnetoSoft\SKICMP\Destination Address List*.

### **Parameters**

*bstrMachineName* is a remote machine name to import from.

*nOverwrite* is a flag to overwrite local DAL. If *nOverwrite* set to 1 local DAL will be overwritten, otherwise imported entries will be added to it.

## 4.11 PingGetReply

### Summary

Retrieves Ping echo reply.

### Syntax

```
PingReplyReceived(VARIANT* pvarRemoteAddress, VARIANT* pvarStatus, VARIANT*  
                  pvarReplyingAddress, VARIANT* pvarResolvedReplyingAddress, VARIANT*  
                  pvarDataSize, VARIANT* pvarRoundTripTime, VARIANT* pvarTTL, VARIANT*  
                  pvarCountHopsInfo)
```

### Description

Retrieves Ping reply when ICMPSendRequest is called.

Note that this method should be used only when SkICMP control is used as a COM server, not an ActiveX control, for instance, when SkICMP is instantiated from ASP page or Windows Scripting Host.

When SkICMP is used as regular ActiveX control, notification event [PingReplyReceived](#) should be used instead.

### Parameters

*pvarRemoteAddress* is the name of the remote host that was queried,

*pvarStatus* is the return status of each individual reply. See section [1.6 Error Codes](#) the complete list of supported error codes,

*pvarReplyingAddress* is the name of the replying remote host,

*pvarResolvedReplyingAddress* is the resolved name of the replying remote host; this argument may have a *value* if optResolve is set to 1.

*pvarDataSize* is the Ping reply packet size,

*pvarRoundTripTime* is the round trip time in milliseconds,

*pvarTTL* is the Ping reply packet TTL value,

*pvarCountHopsInfo* is the route info, this argument may have route info if optRecordRoute or optTimestamp is greater than 0.

### Return value

Return value indicates current state.

Possible values:

997 (ERROR\_IO\_PENDING), control is still processing Ping request.

234 (ERROR\_MORE\_DATA), Ping reply is retrieved.

259 (ERROR\_NO\_MORE\_ITEMS), there is no more data to retrieve.

## ***4.12 PingReset***

### **Summary**

Stop Ping messages (Traceroute messages will not be affected).

### **Syntax**

void PingReset (void)

### **Description**

The PingReset method terminates any pending Ping requests.

### **Parameters**

None.

### ***4.13 ResetICMPSettings***

#### **Summary**

Reset all ICMP settings back to default values.

#### **Syntax**

```
void ResetICMPSettings(void)
```

#### **Description**

All ICMP related settings will be reset to defaults.

The destination address list will remain intact.

#### **Parameters**

None.

## 4.14 TracerouteGetReply

### Summary

Retrieves Traceroute echo reply.

### Syntax

TracerouteReplyReceived(VARIANT\* *pvarRemoteAddress*, VARIANT\* *pvarStatus*, VARIANT\* *pvarReplyingAddress*, VARIANT\* *pvarResolvedReplyingAddress*, VARIANT\* *pvarDataSize*, VARIANT\* *pvarRoundTripTime*, VARIANT\* *pvarTTL*, VARIANT\* *pvarHop*, VARIANT\* *pvarPacket*)

### Description

Retrieves Traceroute reply when ICMPSendRequest is called.

Note that this method should be used only when SkICMP control is used as a COM server, not an ActiveX control, for instance, when SkICMP is instantiated from ASP page or Windows Scripting Host.

When SkICMP is used as regular ActiveX control, notification event [TracerouteReplyReceived](#) should be used instead.

### Parameters

*pvarRemoteAddress* is the name of the remote host that was queried,

*pvarStatus* is the return status of each individual reply. See section [1.6 Error Codes](#) the complete list of supported error codes,

*pvarReplyingAddress* is the name of the replying remote host,

*pvarResolvedReplyingAddress* is the resolved name of the replying remote host; this argument may have a *value* if optResolve is set to 1.

*pvarDataSize* is the Ping reply packet size,

*pvarRoundTripTime* is the round trip time in milliseconds,

*pvarTTL* is the Ping reply packet TTL value,

*pvarHop* is the hop index, starting from 0,

*pvarPacket* is the packet index, starting from 1.

### Return value

Return value indicates current state.

Possible values:

997 (ERROR\_IO\_PENDING), control is still processing Traceroute request.

234 (ERROR\_MORE\_DATA), Traceroute reply is retrieved.

259 (ERROR\_NO\_MORE\_ITEMS), there is no more data to retrieve.

## ***4.15 TraceRouteReset***

### **Summary**

Stop Traceroute messages (Ping messages will not be affected).

### **Syntax**

void TraceRouteReset (void)

### **Description**

The TraceRouteReset method terminates any pending Traceroute requests.

### **Parameters**

None.