

Finger ActiveX Control

**Finger
ActiveX Control
for Microsoft® Windows™**

Version 5.2

**Copyright © 1995 - 2003 by Distinct Corporation
All rights reserved**

Table of Contents

Table of Contents	2
1 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 D_FINGER.TXT	4
2 Properties	7
2.1 Action	7
2.2 Host	8
2.3 Port	9
2.4 User	10
3 Events	11
3.1 OnClose	11
3.2 OnConnect	12
3.3 OnError	13
3.4 OnReceive	15
4 Methods	17
4.1 Abort	17
4.2 Connect	18
4.3 Disconnect	19

1 Overview

1.1 Introduction

The Distinct Finger ActiveX control allows you to finger machines directly from within your applications. This ActiveX control can be used to monitor users logged into a remote system by getting information about the users based on the username, and tracking the amount of time they have been logged in for.

1.2 Usage

See the section entitled "Using Distinct ActiveX controls in various environments" on how to add the control to your project.

After placing a Finger ActiveX control into a form, some properties can be set at design time. The Port property defaults to 79, which is the most commonly used value. The Host property is usually set at run time right before a session is established. This allows the application to request the host name or internet address of the remote server from the user. If an application will always connect to the same host, then the Host property can also be set at design time to minimize user interaction.

The User property is usually not set at design time (although the application can set it at design time). Instead, it is set at run time before a session is established. This allows the application to request information from the user. Please check the reference pages of these properties for more details.

The Action property can only be accessed at run time. To establish a remote session, the value of the Action property must be set to ACTION_CONNECT (or call the Connect method). The Host and Port properties must be set before the connection is established. If the connection can be established, then the OnConnect event will occur before the next line of code is reached. Otherwise, an OnError event will be fired. The result of the query will be returned to the application in one or more OnReceive events.

Once a connection is no longer needed, the Action property must be set to ACTION_DISCONNECT (or the Disconnect method must be called). After the session is disconnected, the OnClose event will occur before the next line of code is reached. Applications must disconnect all connected sessions before terminating.

1.3 Property Summary

Action

Connect, disconnect or abort a remote session

Host

Name of host or dotted decimal internet address

Port

Finger service port on server

User

User name query

1.4 Event Summary

OnClose

Connection has been closed

OnConnect

Connection has been established

OnError

Local error has occurred

OnReceive

More data has been received

1.5 Method Summary

Abort

Abort a remote session

Connect

Connect to remote server

Disconnect

Close connection to remote server

1.6 D_FINGER.TXT

The following provides a complete listing of the D_FINGER.TXT definition file. If your application uses more than one Distinct ActiveX control in the same form, then some definitions will conflict. For example, the FTP Client ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 3
```

in the D_FTP.TXT file and the Telnet ActiveX control includes the definition

```
Global Const ACTION_DISCONNECT = 2
```

in the D_TNET.TXT file. To avoid this conflict, you must rename at least one of the constants (for example, FTP_ACTION_DISCONNECT or TNET_ACTION_DISCONNECT).

```
' Finger ActiveX control  
' (C) Copyright 1995 - 1998 by Distinct Corporation  
' All rights reserved
```

```
' actions
Global Const ACTION_NONE = 0
Global Const ACTION_CONNECT = 1
Global Const ACTION_DISCONNECT = 2
Global Const ACTION_ABORT = 3

' error codes
Global Const ERR_CANNOT_CONNECT = 1
Global Const ERR_HOST_NOT_DEFINED = 2
Global Const ERR_PORT_NOT_DEFINED = 3
Global Const ERR_IN_ACTION = 4
Global Const ERR_CANNOT_CHANGE_HOST = 5
Global Const ERR_CANNOT_CHANGE_PORT = 6
Global Const ERR_UNABLE_TO_GET_SOCKET = 7
Global Const ERR_UNABLE_TO_BIND = 8
Global Const ERR_CANNOT_SEND = 9
```


2 Properties

2.1 Action

Summary

Connect, disconnect or abort a remote session.

Description

The Action property controls the connection state of the Finger ActiveX control. A session can be established, closed or aborted by assigning one of the following values to this property.

Value	Meaning
ACTION_CONNECT	Establish session.
ACTION_DISCONNECT	Close session.
ACTION_ABORT	Abort session.

This property can be changed at run time only.

Before setting the Action property to ACTION_CONNECT, the following properties must be initialized. The Port property must be set to the remote server port (defaults to 79). The Host property must be set to the name or internet address (in dotted decimal notation) of the remote server. The User property must be set to a user name if the application wishes to query a specific user. Otherwise, it can be left as an empty string.

If the connection can be established, then the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. These events will occur before the next statement (i.e. the statement following the assignment of ACTION_CONNECT to the Action property) is executed. The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection has not been established. Please check the reference page of the OnError event for a complete listing of error codes.

Once a connection is no longer needed, the session can be terminated by setting the Action property to ACTION_DISCONNECT. An application must close all connections it has created before it quits. A connection can also be closed by setting the Action property to ACTION_ABORT. This action resets and closes the connection without properly closing down and should not be called under normal circumstances.

The Connect, Disconnect and Abort methods accomplish the same as the above actions. Please check the reference pages of these methods for more detailed information on their usage.

There is no default value for this property.

Example

```
Finger.Host = "speedy.distinct.com"  
Finger.User = "santa"  
Finger.Action = ACTION_CONNECT
```

2.2 Host

Summary

Name of host or dotted decimal internet address.

Description

The Host property specifies the name or internet address of a remote server. This property must be set before a session can be established. There are three possible ways of specifying a remote server.

Machine Name

An application only needs to specify the name of the remote server if the server is located on the same network as the local PC or if the internet address of the server is defined in the local host table. If the remote server is not on the local network, then the underlying protocol will route the traffic through a gateway. If the remote server is not defined in the local host table, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Machine and Domain Name

An application needs to specify the machine name and the domain name if the remote server is not located on the same network as the local PC. Fully domain qualified machine names are written from left to right in ascending order (for example, *speedy.distinct.com*). If both machine and domain names are specified, then the underlying protocol will contact the domain server to resolve the internet address of the server.

Internet Address

Sometimes the user knows only the internet address of the remote server that he or she wants to use. In this case, the internet address can be entered in what is known as the dotted decimal notation (for example, *127.43.101.12*). If the remote server identified by this address is not on the local network, then the underlying protocol will route the traffic through a gateway.

This property can be changed at design time and at run time before a connection has been established. There is no default value for this property.

Example

```
Finger.Host = "127.43.101.12"  
Finger.User = "santa"  
Finger.Action = ACTION_CONNECT
```

2.3 Port

Summary

Finger service port on server.

Description

The Port property specifies the remote port on the server on which the Finger service resides. Most Finger services listen for connection requests on port 79. Sometimes, usually for security reasons, port numbers other than 79 are used for Finger connections.

If the application will be connecting to Finger services on a different port, then the Port property must be set before the connection attempt is made. An application may even want to query the user for the correct port before connecting. The ActiveX control does not verify the setting of the Port property and any value is therefore legal.

This property can be changed at design time and at run time before a connection has been established. The default value for this property is port 79.

Example

```
Finger.Host = "127.43.101.12"  
Finger.User = "santa"  
Finger.Port = 79  
Finger.Action = ACTION_CONNECT
```

2.4 User

Summary

User name used to find a specific person.

Description

The User property specifies the query sent to the server. This property must be set before the Action property is set to ACTION_CONNECT.

The application must set the User property to a user name to retrieve information about the user. If the server can process the query, information about the user is returned. If the application does not have a specific query, then this property can be set to an empty string. In this case, the server will return all the user names currently logged on to the server.

This property can be set at design time or at run time before a connection has been established. There is no default value for this property.

Example

```
Finger.Host = "127.43.101.12"  
Finger.User = "santa"  
Finger.Action = ACTION_CONNECT
```

3 Events

3.1 OnClose

Summary

Connection has been closed.

Description

The OnClose event occurs usually in response to setting the Action property to ACTION_DISCONNECT or ACTION_ABORT (or in response to the Disconnect or Abort method). In most cases, the remote server may close a connection. This will also trigger an OnClose event. In this case, the application must still set the Action property to ACTION_DISCONNECT (or use the Disconnect method) to free up all the resources allocated for the connection. However, this should be done with care during the OnClose event because it might result in an infinite loop.

If this event occurs in response to setting the Action property to ACTION_DISCONNECT (or in response to the Disconnect method), it will occur before the statement following the assignment of ACTION_DISCONNECT (or the Disconnect method) to the Action property is reached.

Normally, an application should simply set a flag in response to this event. Then, this flag can be checked directly after the ACTION_DISCONNECT action (or the Disconnect method) to make sure that the connection was actually terminated.

Example

```
Sub Finger_OnClose ()
    If Connected = True Then
        Connected = False
        Finger.Action = ACTION_DISCONNECT
    End If
End Sub
```

3.2 OnConnect

Summary

Connection has been established.

Description

The OnConnect event occurs in response to setting the Action property to ACTION_CONNECT (or in response to the Connect method). This event will occur before the statement following the assignment of ACTION_CONNECT to the Action property (or the Connect method) is reached.

Normally, an application should simply set a flag in response to this event. Then, this flag can be checked directly after the ACTION_CONNECT action (or the Connect method) to make sure that the connection was actually established.

If the connection could not be established, then the OnError event will be called instead of the OnConnect event.

While handling the OnConnect event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box.

Example

```
Sub Finger_OnConnect ()  
    Connected = TrueEnd Sub
```

3.3 OnError

Summary

Local error has occurred.

Description

The OnError event occurs when a property is accessed in an illegal way or when a connection with the remote server cannot be established. The table below lists all possible error codes delivered by this event.

Value	Meaning
ERR_CANNOT_CONNECT	Unable to connect to remote host.
ERR_HOST_NOT_DEFINED	Host name must be defined before connecting.
ERR_PORT_NOT_DEFINED	Port must be defined before connecting.
ERR_IN_ACTION	Another action is in progress.
ERR_CANNOT_CHANGE_HOST	Cannot change host while connected.
ERR_CANNOT_CHANGE_PORT	Cannot change port while connected.
ERR_UNABLE_TO_GET_SOCKET	Unable to allocate socket.
ERR_UNABLE_TO_BIND	Unable to bind socket.
ERR_CANNOT_SEND	Unable to send query to remote host.

The following describes each error in more detail.

ERR_CANNOT_CONNECT

The remote server is unreachable. Make sure the Host and Port properties are set correctly before setting the Action property to ACTION_CONNECT. Also, the remote host may be down.

ERR_HOST_NOT_DEFINED

The Host property must be set before attempting to establish a connection (by setting the Action property to ACTION_CONNECT).

ERR_PORT_NOT_DEFINED

The Port property must be set before attempting to establish a connection (by setting the Action property to ACTION_CONNECT or by calling the Connect method).

ERR_IN_ACTION

Another action is already in progress.

ERR_CANNOT_CHANGE_HOST

Host cannot be changed while connected. You cannot set the Host property while a session is established.

ERR_CANNOT_CHANGE_PORT

Remote port cannot be changed while connected. You cannot set the Port property while a session is established.

ERR_UNABLE_TO_GET_SOCKET

Unable to allocate socket.

ERR_UNABLE_TO_BIND

Unable to bind socket to local address.

ERR_CANNOT_SEND

Unable to send the query to the remote server.

Example

```
Sub Finger_OnError (ErrorCode As Integer)
  If ErrorCode = ERR_CANNOT_CONNECT Then
    MsgBox "Unable to connect to remote host", 64, "Sample Program"
  End If
End Sub
```

3.4 OnReceive

Summary

More data has been received.

Description

The OnReceive event occurs whenever the underlying protocol stack receives one or more bytes of additional data from the remote server. This event actually delivers the data to the application.

The OnReceive event occurs in response to the execution of the query on the remote server. The data delivered will contain the response from the remote server. One or more OnReceive events may occur to deliver the complete response.

While handling the OnReceive event, an application should not perform tasks which have the potential of requiring a lot of time to complete, such as generating a message box. A substantial delay could cause the application to not receive subsequent OnReceive events.

Example

```
Sub Finger_OnReceive (Buffer As String)
    Dim Message As String

    Message = Buffer
End Sub
```


4. Methods

4.1 Abort

Summary

Abort a remote session.

Syntax

Boolean Abort ()

Description

The Abort method aborts a session to the remote server. This method resets and closes the connection without properly closing down and should not be called under normal circumstances.

The Abort method takes no parameters and returns a boolean. If the connection is successfully reset and closed, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the Action property to ACTION_ABORT.

Example

```
Result = Finger.Abort ()  
If Result = False Then  
    MsgBox "Unable to abort", 64, "Sample Program"  
End If
```

4.2 Connect

Summary

Connect to the remote server.

Syntax

Boolean Connect (*Host*, *User*)

Host String

User String

Description

The Connect method establishes a connection to the remote server.

The Connect method takes a host name (*Host*) and a user name (*User*) as its parameters and returns a boolean. The *Host* parameter must be the name or internet address (in dotted decimal notation) of the remote server. The *User* parameter can be an empty string. If a connection is successfully established, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

If the connection can be established, then the OnConnect event will be fired. If the connection cannot be established, then the OnError event will be fired. These events will occur before the next statement (i.e. the statement following the call to the Connect method) is executed. The application should set a flag in the OnConnect and OnError events, so that it can determine if the session has been established or not. In addition, the application may want to display an error message in the OnError event to inform the user that the connection has not been established. Please check the reference page of the OnError event for a complete listing of error codes.

Calling this method is equivalent to setting the Action property to ACTION_CONNECT.

Example

```
Result = Finger.Connect ("speedy.distinct.com", "")
If Result = False Then
    MsgBox "Unable to connect to server", 64, "Sample Program"
End If
```

4.3 Disconnect

Summary

Close connection to the remote server.

Syntax

Boolean Disconnect ()

Description

Once a connection is no longer needed, the session can be terminated by calling the Disconnect method. An application must close all connections it has created before it quits.

The Disconnect method takes no parameters and returns a boolean. If a connection is successfully terminated, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Calling this method is equivalent to setting the Action property to ACTION_DISCONNECT.

Example

```
Result = Finger.Disconnect ()  
If Result = False Then  
    MsgBox "Unable to disconnect from server", 64, "Sample Program"  
End If
```

