Remote Copy
ActiveX Control

Remote Copy

ActiveX Control

for Microsoft® Windows™

Version 5.2

Distinct Corporation
3315 Almaden Expressway
San Jose, CA 95118 USA

Phone: +1 408-445-3270
Fax: +1 408-445-3274

Email: sales@distinct.com

WWW: http://www.distinct.com

Last updated June 5th, 2003

Published in the United States of America

Table of Contents

1 RCP ActiveX Overview

1.1 Introduction

The Distinct Remote Copy (RCP) ActiveX control allows you to integrate remote copy operations into your applications. An application can copy files between the local and remote machine, or from one remote machine to another. The Remote Copy ActiveX control supports recursive file copy and is able to preserve the original time and date attributes of the file.

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 Remote Copy ActiveX control into a form, some properties can be set at design time. The Options property can be set at design time to a specific value (for example, OPTION_PRESERVE), if the application will always use that value. This property can be changed at any time before setting the Action property to ACTION_COPY (or before calling the FileCopy method) or ACTION_EVENT (or the EventCopy method). The application can set this property at design time to minimize user interaction.

The SrcHost, SrcUser, SrcFile, DestHost, DestUser and DestFile properties are usually set at run time right before a copy operation. The Source and Destination methods can also be used to set the above properties. This allows the application to request the source and destination host names or internet addresses from the user. If the source is the local machine, then set the SrcHost and SrcUser properties to an empty string. If the destination is the local machine, then set the DestHost and DestUser properties to an empty string.

The SrcUser and DestUser properties are required to log into the source and/or destination servers. The SrcFile property is set to the source files or directories that need to be copied to the destination host. To specify multiple source files (or directories), separate the names with a space. Wildcards (such as *.* or *.exe) can also be used to specify a filter for the source files defined by the SrcFile property. The DestFile property should contain the destination file or directory name.

The Action property can only be accessed at run time. To enable the direct copy operation, set the value of the Action property to ACTION_COPY (or call the FileCopy method). The actual copying of files will be handled by the ActiveX control and the application will be notified after every transfer of a file or directory via OnCopy events. The application can abort the file transfer at any time by setting the Action property to ACTION_ABORT (or by calling the Abort method) inside the OnCopy event. However, the OnCopy events will not occur for file transfers between two remote machines.

If the copy process is successful, then the OnDone event will occur before the next statement (i.e. the statement following the assignment of ACTION_COPY to the Action property or the call to the FileCopy method) is executed.

For an event driven file copy, set the value of the Action property to ACTION_EVENT (or call the EventCopy method). In this case, the application must handle the copy operation. The application must also set the file date and time and take care of recursive copying. The ActiveX control will simply transfer data between the source and destination machines. The Remote Copy ActiveX control fires OnTransfer events whenever a section of the file is received or another section of the file can be sent. The application can abort the file transfer at any time by setting the Action property to ACTION_ABORT (or by calling the Abort method) inside the OnTransfer event.

The event driven method allows the application to transfer data to and from storage other than the local disk, such as through DDE or the clipboard. However, this is not possible for file transfers between two remote machines.

1.3 When you must be a Trusted Host

In order to make a connection using RSHELL, REXEC or RCopy you must register the PC running your application (that uses the Remote Execution or Remote Copy ActiveX control) as a trusted host on the Unix machine that you are trying to connect to.

Here is an example of how to make your PC a Trusted Host:

Assuming you are trying to connect to a UNIX system called unix1.mydomain.com (192.0.0.1) and your pc is called mypc.mydomain.com (192.0.0.3) you must update the following files to be able to use REXEC, RSHELL or RCopy successfully:

Add the following entries in /etc/hosts file on the Unix system called unix1.mydomain.com:

192.0.0.3 mypc mypc.mydomain.com

Add the following entries in the /etc/hosts.equiv file:

mypc

mypc.mydomain.com

Updating the /etc/hosts file allows a reference of the remote host by name. In some cases this file may NOT be read at all. If this is your case, you will have to update the appropriate hosts database file on that system. For example if the host uses a name server you will have to modify the name server's database to achieve the desired result.

If on the other hand you need to run RCopy between two UNIX systems, you will need to do all of the above plus: When running the RCopy command, both UNIX systems must have the other system defined in the hosts file and hosts.equiv files. So if your second system is called unix2.mydomain.com, add the following entry in /etc/hosts file

192.0.0.1 unix1 unix1.mydomain.com

192.0.0.3 mypc mypc.mydomain.com

Add the following entries in the /etc/hosts.equiv file:

unix1

unix1.mydomain.com

mypc

mypc.mydomain.com

In this case you will also need to add the information for unix2.mydomain.com to the hosts and hosts.equiv files on the unix1.mydomain.com system.

1.4 RCP ActiveX Property Summary

Action
File based copy, event based copy or abort copy

BinaryData
Contains the binary data received following a call to the RetrieveB method.

DestFile
Destination file or directory name

DestHost
Destination host name

DestUser
Destination user name

EndOfTransfer
End of file transfer

ErrorMessage
Error message returned by remote server

FileAttributes
Read, write and execute attributes of file or directory

FileDate
Last modification date of file or directory

FileName
Name of file or directory

FileSize
Size of file

FileTime
Last modification time of file or directory

FileType
Directory or file

Options
Preserve time and date of file and recursive copy options

SendData
Send data buffer

SrcFile
Source file or directory name

SrcHost
Source host name

SrcUser
Source user name

UseProperty
Send data by parameter or property

UseVariant
Send and receive binary or ascii data

1.5 RCP ActiveX Event Summary

OnCopy
Copy status

OnDone
Copy operation completed

OnError
Local error has occurred

OnTransfer
Receive or send data during event based copy

OnTransferB
Receive or send binary or ascii data during event based copy

1.6 RCP ActiveX Method Summary

Abort
Abort copy

Destination
Set the destination host, user and file names

EventCopy
Event based copy

FileCopy
File based copy

ReceiveB
Receive binary data

SendB
Send binary data

Source
Set the source host, user and file names

1.7 D_RCP.TXT

The following provides a complete listing of the D_RCP.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).

```
' Remote Copy ActiveX Control
' (C) Copyright 1995 - 1998 by Distinct Corporation
' All rights reserved

' actions
Global Const ACTION_NONE = 0
Global Const ACTION_COPY = 1
Global Const ACTION_EVENT = 2
Global Const ACTION_ABORT = 3

' options
Global Const OPTION_NONE = 0
Global Const OPTION_PRESERVE = 1
Global Const OPTION_RECURSIVE = 2
Global Const OPTION_BOTH = 3

' file type
Global Const FILETYPE_FILE = 0
Global Const FILETYPE_DIRECTORY = 1

' file attributes
Global Const ATTRIBUTE_EXECUTE = 1
Global Const ATTRIBUTE_WRITE = 2
Global Const ATTRIBUTE_READ = 4

' transfer type
Global Const XFERTYPE_RECV_INFO = 1
Global Const XFERTYPE_RECV_DATA = 2
Global Const XFERTYPE_SEND_INFO = 3
Global Const XFERTYPE_SEND_DATA = 4

' error codes
Global Const ERR_CANNOT_COPY = 1
Global Const ERR_COPY_ABORTED = 2
Global Const ERR_IN_ACTION = 3
Global Const ERR_INVALID_TIME = 4
Global Const ERR_INVALID_DATE = 5
Global Const ERR_UNABLE_TO_LOAD = 6
```

2 RCP ActiveX Properties

2.1 Action

Summary

File based copy, event based copy or abort copy.

Description

The Action property controls the state of the Remote Copy ActiveX control. The copy operation (file or event based) can be invoked or aborted by assigning one of the following values to this property.

Value	Meaning
ACTION_COPY	File based copy.
ACTION_EVENT	Event based copy.
ACTION_ABORT	Abort copy.

This property can be changed at run time only.

Files can be copied to and from a remote machine either directly or through events. In the direct copy (file based) mode, the Action property is set to ACTION_COPY. This allows an application to copy files of any size without doing any I/O itself. If an application requires events to copy files or needs more control over the copy operation, then the Action property can be set to ACTION_EVENT. The event based file copy allows an application to move data to and from storage other than the local disk, such as through DDE or the clipboard. However, for a remote to remote copy the event mode can **not** be used.

Before setting the Action property to ACTION_COPY or ACTION_EVENT, the following properties must be initialized. The SrcHost and DestHost properties must be set to the names or internet addresses (in dotted decimal notation) of the source and destination machines. The SrcUser and DestUser properties must contain a valid user name to complete the login process. If the source or the destination is the local machine, then the corresponding host and user properties must be set to an empty string. The SrcFile property can contain one or more file or directory names (separated by spaces) to be copied to the destination. The DestFile property must contain a file or directory name.

If the connection can be established, then the copy operation will proceed and an OnDone event will be fired to indicate the completion of the copy operation. 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_COPY or ACTION_EVENT to the Action property) is executed. The application may want to display an error message in the OnError event to inform the user that the copy operation was not successful. Please check the reference page of the OnError event for a complete listing of error codes.

If the Action property is set to ACTION_COPY, then the Remote Copy ActiveX control takes care of obtaining the file information, reading from or writing to files and recursively copying and creating subdirectories. An OnCopy event will be fired for each file or directory that is copied (unless files are copied between two remote machines). The application can use this event to display any status information, such as a progress bar. Please check the reference page of the OnCopy event for more information.

A copy operation in the direct (file based) mode can be aborted at any time by setting the Action property to ACTION_ABORT inside the OnCopy event.

If the Action property is set to ACTION_EVENT, then the Remote Copy ActiveX control will not obtain file information or file contents. Instead, it will rely on the application to provide the correct information. In addition, the application must take care of any recursive copy operations. The ActiveX control will fire OnTransfer events to obtain or deliver all the information. The OnTransfer event will contain a TransferType argument to indicate whether the application should provide file information or the next portion of the contents of a file or whether the application is being given file information or the next portion of the contents of a file. Please check the reference page of the OnTransfer event for complete details on event based transfer.

The properties FileName, FileType, FileSize, FileDate, FileTime and FileAttributes need to be set or read during the OnTransfer event for an event based copy. Please check the reference pages of these properties for more details.

To abort the event based copy operation set the Action property to ACTION_ABORT inside the OnTransfer event.

The FileCopy, EventCopy 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

```
Rcp.SrcHost = ""
Rcp.SrcUser = ""
Rcp.SrcFile = "c:\sample.exe"
Rcp.DestHost = "speedy"
Rcp.DestUser = "test"
Rcp.DestFile = "sample.exe"
Rcp.Action = ACTION_COPY
```

2.2 BinaryData

Summary

Contains the binary data received following a call to the ReceiveB method.

Description

After each call to the ReceiveB method in a C# application this property needs to be read to access binary data.

Example

```
Private void Rcp_OnReceive (object sender, System.EventArgs e)
{
    int len;
    int bytes;
    Object arrData = new byte[bytes];

    Len = axRcp1.ReceiveB (arrData);
    If  (len > 0)
    {
            object pBuf = new Byte[len];
            pBuf = axRcp1.BinaryData;
            byte []RcvByte = (byte [])pBuf;
            ……..
            ……..
    }
}
```

2.3 DestFile

Summary

Destination file or directory name.

Description

The DestFile property specifies the destination file name or directory name. The destination could be a remote host or the local machine. If the destination is the local machine, then both the DestHost and DestUser properties must be set to an empty string.

Usually, this property is set before a copy operation is invoked. The DestFile property can contain only one file or directory name.

The Destination method can also be used to set this property.

This property can be set at design time or at run time before a copy operation is invoked (by setting the Action property to either ACTION_COPY or ACTION_EVENT). There is no default value for this property.

Example

```
Rcp.SrcHost = ""
Rcp.SrcUser = ""
Rcp.SrcFile = "c:\sample.exe"
Rcp.DestHost = "speedy.distinct.com"
Rcp.DestUser = "test"
Rcp.DestFile = "sample.exe"
Rcp.Action = ACTION_COPY
```

2.4 DestHost

Summary

Destination host name.

Description

The DestHost property specifies the name or internet address of the destination machine. This property must be set before a copy operation is started. If the destination is the local machine, then this property must be set to an empty string. If the destination is a remote machine, then there are three possible ways of specifying the destination.

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.

The Destination method can also be used to set this property.

This property can be changed at design time and at run time before a copy operation is invoked (by setting the Action property to either ACTION_COPY or ACTION_EVENT). There is no default value for this property.

Example

```
Rcp.SrcHost = ""
Rcp.SrcUser = ""
Rcp.SrcFile = "c:\sample.exe"
Rcp.DestHost = "127.43.101.12"
Rcp.DestUser = "test"
Rcp.DestFile = "sample.exe"
Rcp.Action = ACTION_COPY
```

2.5 DestUser

Summary

Destination user name.

Description

The DestUser property specifies the destination user name. The destination can be a remote host or the local machine. If the destination is the local machine, then both the DestHost and DestUser properties must be set to an empty string.

If the destination is a remote machine, then the DestUser property must be set to the user login name on the remote machine to complete the login process.

For security reasons, the user name is usually obtained from the user just before a copy operation. If security is not an important issue and the application will always connect to the same remote server with the same user name, then this property can be set at design time. Usually, this property is set before a copy operation is invoked.

The Destination method can also be used to set this property.

This property can be set at design time or at run time before a copy operation is invoked (by setting the Action property to either ACTION_COPY or ACTION_EVENT). There is no default value for this property.

Example

```
Rcp.SrcHost = ""
Rcp.SrcUser = ""
Rcp.SrcFile = "c:\sample.exe"
Rcp.DestHost = "speedy"
Rcp.DestUser = "test"
Rcp.DestFile = "sample.exe"
Rcp.Action = ACTION_COPY
```

2.6 EndOfTransfer

Summary

End of file transfer.

Description

The EndOfTransfer property is used to signal that there are no more files or directories to be copied in the OnTransfer event. If UseProperty is set to True, the control checks this property to determine if more files or directories are to be copied, otherwise it will determine the end of transfer by when the *BufferLength* parameter is set to -1. The UseProperty property should usually be set to False, and the event parameter is normally used. See the UseProperty property for more information on when to use this property.

This property should be set in the OnTransfer event. This property can be read at any time. The default value for this property is False.

Example

```
Sub Rcp_OnTransfer (TransferType As Integer, Buffer As String, BufferLength As Integer)
    Dim Done As Integer                 ' initialize Done = False
    Dim Message As String               ' initialize Message = "This is a test"

    Select Case TransferType
        .
        .
        .
        Case XFERTYPE_SEND_INFO                         ' send information
            If (Done = False) Then
                Rcp.FileType = FILETYPE_FILE
                Rcp.FileName = Rcp.SrcFile
                Rcp.FileSize = Len(Message)
                Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
                Rcp.FileTime = Format(Now, "hh:mm:ss")
                Rcp.FileDate = Format(Now, "dd/mm/yy")
            Else
                Rcp.SendData = ""                       'end of transfer
                Rcp.EndOfTransfer = True
                Done = False
            End If

        Case XFERTYPE_SEND_DATA                         ' send data
            If (Done = False) Then
                If (Len(Message) < BufferLength) Then
                    Rcp.SendData = Message
                    Done = True
                End If
            Else
                Rcp.SendData = ""                       'end of file
            End If
    End Select
End Sub
```

2.7 ErrorMessage

Summary

Error message returned by remote server.

Description

The ErrorMessage property contains the response sent by the remote machine if the connection could not be established or if the copy operation could not be properly executed for some other reason. This property will contain the exact response from the server explaining the error that occurred.

The ErrorMessage property will also contain any error message that the underlying protocol stack might return during the process of copying files.

If setting the Action property to ACTION_COPY or ACTION_EVENT (or if calling the FileCopy or EventCopy method) generates an error, then the ErrorMessage property can be used to retrieve the exact error message sent by the remote machine. This message will describe the reason for the failure of ACTION_COPY or ACTION_EVENT (or FileCopy or EventCopy method).

This property can only be read at run time. There is no default value for this property.

Example

MsgBox Rcp.**ErrorMessage**, 64, "Sample Program"

2.8 FileAttributes

Summary

Read, write and execute attributes of file or directory.

Description

The FileAttributes property can be referenced during an OnCopy or OnTransfer event. This property is generally used during event based copy operations in the OnTransfer event.

If the value of the TransferType argument in the OnTransfer event is TYPE_RECV_INFO, then this property will contain the attributes of the file given by the FileName property. If the value of TransferType is TYPE_SEND_INFO, then the application must set this property to the permissions of the file or directory being copied.

The value of the FileAttributes property can be any combination of the values mentioned below.

Value	Meaning
ATTRIBUTE_EXECUTE	Execute permission.
ATTRIBUTE_WRITE	Write permission.
ATTRIBUTE_READ	Read permission.

The '+' operator can be used to combine the different values. For example,
FileAttributes = ATTRIBUTE_EXECUTE + ATTRIBUTE_WRITE + ATTRIBUTE_READ

In the OnCopy event, this property will contain the current file attributes (permissions).

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Dim Message As String

Message = "This is a test"
Rcp.FileType = FILETYPE_FILE
Rcp.FileName = Rcp.SrcFile
Rcp.FileSize = Len(Message)
Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
Rcp.FileTime = Format(Now, "hh:mm:ss")
Rcp.FileDate = Format(Now, "dd/mm/yy")
```

2.9 FileDate

Summary

Last modification date of file or directory.

Description

The FileDate property specifies the date of the last modification of the file or directory specified in the FileName property. This property is set to an empty string unless the Options property is set to OPTION_PRESERVE or OPTION_BOTH.

This property is referenced during an OnCopy or OnTransfer event. If the value of TransferType in the OnTransfer event is TYPE_RECV_INFO, then this property will contain the file date. If the value of TransferType is TYPE_SEND_INFO, then the application must set this property to the desired file date.

The syntax for this property is **"day/month/year"**, where **day** can be any number in the range of 1 to 31 as the day of the month, **month** can be any number between 1 and 12 specifying the month and **year** denotes the year minus 1900. The separator is a '/'. The application must strictly follow this syntax for correct results.

In the OnCopy event, this property will contain the file date.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Dim Message As String

Message = "This is a test"
Rcp.FileType = FILETYPE_FILE
Rcp.FileName = Rcp.SrcFile
Rcp.FileSize = Len(Message)
Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
Rcp.FileTime = Format(Now, "hh:mm:ss")
Rcp.FileDate = Format(Now, "dd/mm/yy")
```

2.10 FileName

Summary

Name of file or directory.

Description

The FileName property specifies the name of the file or directory that is currently being copied. This property is used during an OnCopy or OnTransfer event.

If the value of TransferType in the OnTransfer event is TYPE_RECV_INFO, then the file or directory name will be available in the FileName property. The FileType property tells the application whether FileName specifies a file or a directory.

If the value of TransferType is TYPE_SEND_INFO, then the application must set this property to the name of the current file or directory.

In the OnCopy event, this property will contain the file name.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Dim Message As String

Message = "This is a test"
Rcp.FileType = FILETYPE_FILE
Rcp.FileName = Rcp.SrcFile
Rcp.FileSize = Len(Message)
Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
Rcp.FileTime = Format(Now, "hh:mm:ss")
Rcp.FileDate = Format(Now, "dd/mm/yy")
```

2.11 FileSize

Summary

Size of file.

Description

The FileSize property specifies the size of the file or directory specified by the FileName property. This property is referenced during an OnCopy or OnTransfer event.

If the value of TransferType in the OnTransfer event is TYPE_RECV_INFO, then the size of the file or directory will be given in the FileSize property. The FileType property tells the application whether the FileName property specifies a file or a directory. If the FileType is FILETYPE_DIRECTORY, then the value of the FileSize property is 0.

If the value of TransferType is TYPE_SEND_INFO, then the application must set this property to the size of the file or directory. If the FileName property denotes a directory (FileType property is set to FILETYPE_DIRECTORY), then the value of the FileSize property should be set to 0.

In the OnCopy event, this property will contain the file size.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Dim Message As String

Message = "This is a test"
Rcp.FileType = FILETYPE_FILE
Rcp.FileName = Rcp.SrcFile
Rcp.FileSize = Len(Message)
Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
Rcp.FileTime = Format(Now, "hh:mm:ss")
Rcp.FileDate = Format(Now, "dd/mm/yy")
```

2.12 FileTime

Summary

Last modification time of file or directory.

Description

The FileTime property specifies the time of the last modification of the file or directory specified in the FileName property. This property is set to an empty string unless the Options property is set to OPTION_PRESERVE or OPTION_BOTH.

This property is referenced during an OnCopy or OnTransfer event. If the TransferType is TYPE_RECV_INFO in the OnTransfer event, then this property will contain the file modification time. If the TransferType is TYPE_SEND_INFO, then the application must set this property to the desired file time.

The syntax for this property is **"hour:minute:second"**, where **hour** is in the range of 0 to 23, **minute** is any number between 0 and 59 and **second** is in the range 0 to 59. The separator is a ':'. The application must strictly follow this syntax for correct results.

In the OnCopy event, this property will contain the file time.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Dim Message As String

Message = "This is a test"
Rcp.FileType = FILETYPE_FILE
Rcp.FileName = Rcp.SrcFile
Rcp.FileSize = Len(Message)
Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
Rcp.FileTime = Format(Now, "hh:mm:ss")
Rcp.FileDate = Format(Now, "dd/mm/yy")
```

2.13 FileType

Summary

Directory or file.

Description

The FileType property indicates whether the name specified in the FileName property refers to a file or a directory. The value of the FileType property can be one of the two values listed below.

Value	Meaning
FILETYPE_DIRECTORY	Directory.
FILETYPE_FILE	File.

This property is referenced during an OnCopy or OnTransfer event. If the TransferType is TYPE_RECV_INFO in the OnTransfer event, then this property will contain the file or directory type.

If the TransferType is TYPE_SEND_INFO, then the application must set this property to the file or directory type. If the FileName property contains the name of a file, then the FileType property must be set to FILETYPE_FILE. If the FileName property contains the name of a directory, then the FileType property must be set to FILETYPE_DIRECTORY.

In the OnCopy event, this property will contain the file or directory type.

This property can be changed at design time and at run time. There is no default value for this property.

Example

```
Dim Message As String

Message = "This is a test"
Rcp.FileType = FILETYPE_FILE
Rcp.FileName = Rcp.SrcFile
Rcp.FileSize = Len(Message)
Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
Rcp.FileTime = Format(Now, "hh:mm:ss")
Rcp.FileDate = Format(Now, "dd/mm/yy")
```

2.14 Options

Summary

Preserve time and date of file and recursive copy options.

Description

The Options property is used to specify the various options during the copy operation. The value of the Options property can be one of the values listed below.

Value	Meaning
OPTION_NONE	None.
OPTION_PRESERVE	Preserve file date and time.
OPTION_RECURSIVE	Recursive copy.
OPTION_BOTH	Preserve date and time and recursive copy.

If the Options property is set to OPTION_PRESERVE, then the date and time stamp of the source file will be copied to the destination file. Otherwise, the destination file will not have the same date and time stamp as the source file.

If the Options property is set to OPTION_RECURSIVE, then the Remote Copy ActiveX control will not only copy the files in the source directory, but also in all of its subdirectories. In the case of ACTION_COPY , the ActiveX control will take care of creating all the subdirectories.

To specify both options set the Options property to OPTION_BOTH.

This property can be changed at design time or at run time before a copy operation is invoked (by setting the Action property to either ACTION_COPY or ACTION_EVENT or by calling the FileCopy or EventCopy method). The default value for this property is OPTION_NONE.

Example

```
Rcp.SrcHost = ""
Rcp.SrcUser = ""
Rcp.SrcFile = "c:\sample.exe"
Rcp.DestHost = "speedy.distinct.com"
Rcp.DestUser = "test"
Rcp.DestFile = "sample.exe"
Rcp.Options = OPTION_PRESERVE
Rcp.Action = ACTION_COPY
```

2.15 SendData

Summary

Send data buffer.

Description

The SendData property is used to send data in the OnTransfer event. If UseProperty is set to True, the control will get the data to send from the SendData property, otherwise it will get the the data from the *Buffer* parameter. The UseProperty property should generally be set to False and the event parameter will be used. See the UseProperty for more information on when to use this property

This property should be set in the OnTransfer event. This property can be read at any time. There is no default value for this property.

Example

```
Sub Rcp_OnTransfer (TransferType As Integer, Buffer As String, BufferLength As Integer)
    Dim Done As Integer                 ' initialize Done = False
    Dim Message As String               ' initialize Message = "This is a test"

    Select Case TransferType
        .
        .
        .
        Case XFERTYPE_SEND_INFO                         ' send information
            If (Done = False) Then
                Rcp.FileType = FILETYPE_FILE
                Rcp.FileName = Rcp.SrcFile
                Rcp.FileSize = Len(Message)
                Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
                Rcp.FileTime = Format(Now, "hh:mm:ss")
                Rcp.FileDate = Format(Now, "dd/mm/yy")
            Else
                Rcp.SendData = ""                       'end of transfer
                Rcp.EndOfTransfer = True
                Done = False
            End If

        Case XFERTYPE_SEND_DATA                         ' send data
            If (Done = False) Then
                If (Len(Message) < BufferLength) Then
                    Rcp.SendData = Message
                    Done = True
                End If
            Else
                Rcp.SendData = ""                       'end of file
            End If
    End Select
End Sub
```

2.16 SrcFile

Summary

Source file or directory name.

Description

The SrcFile property specifies the source file or directory names. The source can be a remote host or the local machine. If the source is the local machine, then both the SrcHost and SrcUser properties must be set to an empty string.

Usually, this property is set before a copy operation is invoked. The SrcFile property may contain one or more file or directory names. Multiple file or directory names must be separated by spaces.

The Source method can also be used to set this property.

This property can be set at design time or at run time before a copy operation is invoked (by setting the Action property to either ACTION_COPY or ACTION_EVENT or by calling the FileCopy or EventCopy method). There is no default value for this property.

Example

```
Rcp.SrcHost = "127.43.101.12"
Rcp.SrcUser = "test"
Rcp.SrcFile = "/tmp/sample.exe"
Rcp.DestHost = ""
Rcp.DestUser = ""
Rcp.DestFile = "c:\sample.exe"
Rcp.Action = ACTION_COPY
```

2.17 SrcHost

Summary

Source host name.

Description

The SrcHost property specifies the name or internet address of the source machine. This property must be set before a copy operation. If the source is the local machine, then the SrcHost property must be set to an empty string. If the source is a remote machine, then there are three possible ways of specifying the source.

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.

The Source method can also be used to set this property.

This property can be changed at design time and at run time before a copy operation is invoked (by setting the Action property to either ACTION_COPY or ACTION_EVENT or by calling the FileCopy or EventCopy method). There is no default value for this property.

Example

```
Rcp.SrcHost = "speedy"
Rcp.SrcUser = "test"
Rcp.SrcFile = "/tmp/sample.exe"
Rcp.DestHost = ""
Rcp.DestUser = ""
Rcp.DestFile = "c:\sample.exe"
Rcp.Action = ACTION_COPY
```

2.18 SrcUser

Summary

Source user name.

Description

The SrcUser property specifies the source user name. The source could be a remote host or the local machine. If the source is the local machine, then both the SrcHost and SrcUser properties must be set to an empty string.

If the source is a remote machine, then the SrcUser property must be set to the user login name on the remote machine to complete the login process.

For security reasons, the user name is usually obtained from the user just before a copy operation. If security is not an important issue and the application will always connect to the same remote server with the same user name, then this property can be set at design time. Usually, this property is set before a copy operation is invoked.

The Source method can also be used to set this property.

This property can be set at design time or at run time before a copy operation is invoked (by setting the Action property to either ACTION_COPY or ACTION_EVENT or by calling the FileCopy or EventCopy method). There is no default value for this property.

Example

```
Rcp.SrcHost = "speedy.distinct.com"
Rcp.SrcUser = "test"
Rcp.SrcFile = "/tmp/sample.exe"
Rcp.DestHost = ""
Rcp.DestUser = ""
Rcp.DestFile = "c:\sample.exe"
Rcp.Action = ACTION_COPY
```

2.19 UseProperty

Summary

Send data by parameter or property.

Description

The UseProperty property is used to specify whether to use the *Buffer* and *BufferLength* parameters or the SendData and EndOfTransfer properties in the OnTransfer event. If UseProperty is set to True, the control will get the data to send from the SendData property, otherwise it will get the the data from the *Buffer* parameter. To indicate the that there are no more files or directories to be copied set the EndOfTransfer property to True if UseProperty is True, otherwise the *BufferLength* parameter should be set to -1. See the SendData and EndOfTransfer properties and the OnTransfer event for more information.

The UseProperty property is generally set to False, and the event parameters should be used in most cases. It should only be set to True in environments like Visual J++ where the control is unable to get the new value of the event parameter. This is because Visual J++ has to use VBScript to catch and pass the ActiveX events, and VBScript is unable to pass back parameters to the control.

This property should be set before setting the Action property to ACTION_COPY (or calling the EventCopy method) . This property can be read at any time. The default value for this property is False.

Example

Rcp.**UseProperty** = False

2.20 UseVariant

Summary

Send and receive binary or ascii data.

Description

The UseVariant property is used to specify whether data is to be received or sent in binary or ascii form. If this property is set to True, the OnTransferB event will be fired when data arrives or when there is data to be sent; otherwise, the OnReceive event is fired.

This property should be set before any data arrives or before sending any data. This property can be read at any time. The default value for this property is False.

Example

Rcp.**UseVariant** = True

3 RCP ActiveX Events

3.1 OnCopy

Summary

Copy status.

Description

The OnCopy event occurs in response to setting the Action property to ACTION_COPY (or in response to the FileCopy method). Except in the case of copy operations between two remote machines, the OnCopy event will occur each time a file has been successfully copied.

This event will occur one or more times before the statement following the assignment of ACTION_COPY to the Action property (or the call to the FileCopy method) is reached.

Normally, an application should simply use this event to show the status of the copy operation. The application can obtain the current file name from the FileName property. The FileType property indicates whether FileName specifies a file or a directory. The FileAttributes property gives the file permissions, the date and time of the current file is given by the FileDate and FileTime properties and the FileSize property indicates the size of the current file.

If the application needs to abort the copy operation, then the Action property can be set to ACTION_ABORT (or the Abort method can be called) inside this event. This will abort any further file copy operations.

Example

```
Sub Rcp_OnCopy ()
    Dim Message As String

    If (Rcp.FileType = FILETYPE_DIRECTORY) Then
        Message = "Copied Directory " + Rcp.FileName        ' transferred directory
    Else
        Message = "Copied File " + Rcp.FileName + ": "      ' transferred file
        Message = Message & Rcp.FileSize & " bytes long"
    End If
    MsgBox Message, 64, "Sample Program"
End Sub
```

3.2 OnDone

Summary

Copy operation completed.

Description

The OnDone event occurs in response to setting the Action property to ACTION_COPY or ACTION_EVENT (or in response to the FileCopy or EventCopy method). This event will occur before the statement following the assignment of ACTION_COPY or ACTION_EVENT to the Action property (or before the call to the FileCopy method or EventCopy method) is reached.

If the copy operation is successfully completed, then the OnDone event is fired. If the copy operation could not be completed, then the OnError event will be called instead of the OnDone event.

An application may want to display a message in this event to indicate successful completion of the copy operation.

Example

```
Sub Rcp_OnDone ()
    MsgBox "File(s) transferred successfully", 64, "Sample Program"
End 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 machine cannot be established. The table below lists all possible error codes delivered by this event.

Value	Meaning
ERR_CANNOT_COPY	Unable to copy files.
ERR_IN_ACTION	Another action is in progress.
ERR_COPY_ABORTED	File copy aborted.
ERR_INVALID_TIME	Invalid file time specified.
ERR_INVALID_DATE	Invalid file date specified.

The following describes each error in more detail.

ERR_CANNOT_COPY
The remote machine is unreachable or an error occurred during the copy operation. The host and user names may be incorrect or the remote machine may be down.

ERR_IN_ACTION
Another action is already in progress.

ERR_COPY_ABORTED
The file copy operation was aborted by setting the Action property to ACTION_ABORT (or by using the Abort method).

ERR_INVALID_TIME
The file modification time specified in the FileTime property was not valid. Please check the reference page of the FileTime property for the correct syntax.

ERR_INVALID_DATE
The file modification date specified in the FileDate property was not valid. Please check the reference page of the FileDate property for the correct syntax.

An application should also check the ErrorMessage property to obtain any error messages returned by the underlying protocol or the remote machine.

Example

```
Sub Rcp_OnError (ErrorCode As Integer)
    If ErrorCode = ERR_CANNOT_COPY Then
        MsgBox "Unable to copy file(s)", 64, "Sample Program"
    End If
End Sub
```

3.4 OnTransfer

Summary

Receive or send data during event based copy.

Description

The OnTransfer event occurs in response to setting the Action property to ACTION_EVENT (or in response to the EventCopy method). Except in the case of copy operations between two remote machines, one or more OnTransfer events will occur to request portions of the file contents from the application or to deliver portions of the file contents to the application.

These events will occur before the statement following the assignment of ACTION_COPY to the Action property (or the call to the EventCopy method) is reached.

This event contains three arguments: *TransferType*, *Buffer* and *BufferLength.*

Depending on the value of the *TransferType* argument, the application must either send information about the file or portions of the file data to the ActiveX control or is being given information about the file or portions of the file data by the ActiveX control. The value of *TransferType* will be one of the values given below.

Value	Meaning
TYPE_RECV_INFO	Receive file or directory information.
TYPE_RECV_DATA	Receive file contents.
TYPE_SEND_INFO	Send file or directory information.
TYPE_SEND_DATA	Send file contents.

If the *TransferType* argument is set to TYPE_RECV_INFO, then the name of the current file is available in the FileName property. The FileType property indicates whether FileName specifies a file or a directory. The FileAttributes property specifies the file permissions, the date and time of the file are given in the FileDate and FileTime properties and the FileSize property indicates the size of the file.

If the *TransferType* argument is set to TYPE_RECV_DATA, then the argument *Buffer* is a character string containing a portion or all of the file data received from the remote machine. The argument *BufferLength* specifies the number of bytes in the buffer. If *BufferLength* is set to 0, then the end of the file has been reached. For directories no TYPE_RECV_DATA event will occur.

If the *TransferType* argument is set to TYPE_SEND_INFO, then the application must set the FileName, FileType, FileSize, FileAttributes, FileTime and FileDate properties. To indicate that there are no more files or directories to be transferred, the application must set *BufferLength* to -1. In some environments, such as Visual J++, the new value assigned to the *BufferLength* parameter can not be successfully retrieved by the control. In these cases the UseProperty property should be set to True, and the EndOfTransfer property should be set to True to indicate that there are no more files or directories to be transferred. The UseProperty property must be set to True before the Action property is set to ACTION_COPY (or before calling the EventCopy method).

If the *TransferType* argument is set to TYPE_SEND_DATA, then the *BufferLength* argument will indicate the maximum number of bytes that can be assigned to *Buffer*. The application must assign all or a portion of the file data (up to the maximum number of bytes specified by *BufferLength*) to *Buffer*. The *BufferLength* argument must then be set to the number of bytes actually assigned to *Buffer*. To indicate that the end of file has been reached, *BufferLength* must be set to 0. In some environments, such as Visual J++, the new value assigned to the *Buffer* parameter can not be successfully retrieved by the control. In these cases the UseProperty property should be set to True, and the data should be assigned to the SendData property instead of the *Buffer* parameter. The UseProperty property must be set to True before the Action property is set to ACTION_COPY (or

before calling the EventCopy method) In this case treat the SendData property exactly as you would the *Buffer* parameter. To indicate that the end of file has been reached, SendData should be set to an empty string. If UseProperty is True, the *BufferLength* parameter will be ignored.

Normally, an application should use a case statement to handle all four possibilities of this event. If the application needs to abort the copy operation, then the Action property can be set to ACTION_ABORT inside this event. This will abort any further file copy operations.

Example

```
Sub Rcp_OnTransfer (TransferType As Integer, Buffer As String, BufferLength As Integer)
    Dim Done As Integer                 ' initialize Done = False
    Dim Message As String               ' initialize Message = "This is a test"

    Select Case TransferType
        Case XFERTYPE_RECV_INFO                         ' get information
            Message = "Name: " + Rcp.FileName
            If (Rcp.FileType = FILETYPE_DIRECTORY) Then
                Message = Message + "Type: Directory"   ' directory
            Else
                Message = Message + "Type: File"        ' file
            End If
            Message = Message & "Size:" & Rcp.FileSize
            MsgBox Message, 64, "Sample Program"

        Case XFERTYPE_RECV_DATA                         ' get data
            If (BufferLength <> 0) Then
                MsgBox Buffer, 64, "Sample Program"
            End If

        Case XFERTYPE_SEND_INFO                         ' send information
            If (Done = False) Then
                Rcp.FileType = FILETYPE_FILE
                Rcp.FileName = Rcp.SrcFile
                Rcp.FileSize = Len(Message)
                Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
                Rcp.FileTime = Format(Now, "hh:mm:ss")
                Rcp.FileDate = Format(Now, "dd/mm/yy")
            Else
                Buffer = ""                             'end of transfer
                BufferLength = -1
                Done = False
            End If

        Case XFERTYPE_SEND_DATA                         ' send data
            If (Done = False) Then
                If (Len(Message) < BufferLength) Then
                    Buffer = Message
                    BufferLength = Len(Message)
                    Done = True
                End If
            Else
                Buffer = ""                             'end of file
                BufferLength = 0
            End If
    End Select
End Sub
```

3.5 OnTransferB

Summary

Receive and send binary or ascii data during event based copy.

Description

The OnTransferB event occurs in response to setting the Action property to ACTION_EVENT (or in response to the EventCopy method). Except in the case of copy operations between two remote machines, one or more OnTransferB events will occur to request portions of the file contents from the application or to deliver portions of the file contents to the application.

These events will occur before the statement following the assignment of ACTION_COPY to the Action property (or the call to the EventCopy method) is reached.

This event contains two arguments: *TransferType* and *Bytes.*

Depending on the value of the *TransferType* argument, the application must either send information about the file or portions of the file data to the ActiveX control or is being given information about the file or portions of the file data by the ActiveX control. The value of *TransferType* will be one of the values given below.

Value	Meaning
TYPE_RECV_INFO	Receive file or directory information.
TYPE_RECV_DATA	Receive file contents.
TYPE_SEND_INFO	Send file or directory information.
TYPE_SEND_DATA	Send file contents.

If the *TransferType* argument is set to TYPE_RECV_INFO, then the name of the current file is available in the FileName property. The FileType property indicates whether FileName specifies a file or a directory. The FileAttributes property specifies the file permissions, the date and time of the file are given in the FileDate and FileTime properties and the FileSize property indicates the size of the file.

If the *TransferType* argument is set to TYPE_RECV_DATA, then the ReceiveB method must be called to retrieve a portion or all of the file data received from the remote machine. The argument *Bytes* specifies the number of bytes available to be retrieved. If *Bytes* is set to 0, then the end of the file has been reached. For directories no TYPE_RECV_DATA event will occur.

If the *TransferType* argument is set to TYPE_SEND_INFO, then the application must set the FileName, FileType, FileSize, FileAttributes, FileTime and FileDate properties. To indicate that there are no more files or directories to be transferred, the application must set *Bytes* to -1.

If the *TransferType* argument is set to TYPE_SEND_DATA, then the *Bytes* argument will indicate the maximum number of bytes that can be sent at this moment. The application must pass all or a portion of the file data (up to the maximum number of bytes specified by *Bytes*) to the SendB method. To indicate that the end of file has been reached, the Bytes parameter to the SendB method must be set to 0.

Normally, an application should use a case statement to handle all four possibilities of this event. If the application needs to abort the copy operation, then the Action property can be set to ACTION_ABORT inside this event. This will abort any further file copy operations.

Example

```
Sub Rcp_OnTransferB (TransferType As Integer, Bytes As Long)
    Dim Done As Integer                    ' initialize Done = False
    Dim Message As String                  ' initialize Message = "This is a test"
```

```
Select Case TransferType
    Case XFERTYPE_RECV_INFO                          ' get information
        Message = "Name: " + Rcp.FileName
        If (Rcp.FileType = FILETYPE_DIRECTORY) Then
            Message = Message + "Type: Directory"    ' directory
        Else
            Message = Message + "Type: File"         ' file
        End If
        Message = Message & "Size:" & Rcp.FileSize
        MsgBox Message, 64, "Sample Program"

    Case XFERTYPE_RECV_DATA                          ' get data
        Dim Buffer() As Byte
        Dim Siz As Long
        Dim i As Integer

        Siz = Rcp.ReceiveB (Buffer, Bytes)
        If Siz > 0 Then
            Open "c:\abc.exe" For Binary Access Write As #1
            For i = 1 To Siz
                Put #1, i, Buffer (x)
            Next i
            Close #1
        Else
            MsgBox "Cannot receive binary data", 64, "Sample Program"
        End If
    Case XFERTYPE_SEND_INFO                          ' send information
        If (Done = False) Then
            Rcp.FileType = FILETYPE_FILE
            Rcp.FileName = Rcp.SrcFile
            Rcp.FileSize = Len(Message)
            Rcp.FileAttributes = ATTRIBUTE_READ + ATTRIBUTE_WRITE
            Rcp.FileTime = Format(Now, "hh:mm:ss")
            Rcp.FileDate = Format(Now, "dd/mm/yy")
        Else
            Bytes = -1                               'end of transfer
            Done = False
        End If

    Case XFERTYPE_SEND_DATA                          ' send data
        Dim Buffer(1 to Bytes) As Byte
        Dim i As Integer

        If (Siz > Bytes) Then
            For i = 1 to Bytes
                Get #1, , Buffer(i)
            Next i
            Result = Rcp.SendB(Buffer, Bytes)        ' send data
            Siz = Siz - Bytes
        ElseIf (Siz > 0) Then
            For i = 1 to Siz
                Get #1, , Buffer(i)
            Next i
            Result = Rcp.SendB(Buffer, Siz)          ' send data
```

```
            Siz = 0
        Else
            Result = Rcp.SendB (Buffer, 0)                    ' end of file
        End If
    End Select
End Sub
```

4 RCP ActiveX Methods

4.1 Abort

Summary

Abort copy.

Syntax

Boolean Abort ()

Description

The Abort method aborts a copy operation in the direct (file based) mode at any time. To abort a direct copy operation, call this method in the OnCopy event. To abort an event based copy operation, call this method in the OnTransfer event.

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

4.2 Destination

Summary

Set the destination host, user, and file names.

Syntax

Boolean Destination (*Host, User, File*)

Host	String
User	String
File	String

Description

The Destination method sets the DestHost, DestUser and DestFile properties. This method should be called before calling the FileCopy or the EventCopy method (or before setting the Action property to ACTION_COPY or ACTION_EVENT).

The Destination method takes a destination host name (*Host*), a destination user name (*User*) and a destination file name (*File*) as its parameters and returns a boolean. The destination host name must be set to the name or internet address (in dotted decimal notation) of the destination machine. The *User* must be a valid user name to complete the login process. If the destination is the local machine, then the *Host* and the *User* parameters must be set to an empty string. The destination file name must contain a file or directory name.

If the three destination properties are successfully set, 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 DestHost property, the DestUser property, and the DestFile property.

Example

```
Result = Rcp.Destination ("speedy.distinct.com", "santa", "abc")
If Result = False Then
    MsgBox "Cannot set destination information", 64, "Sample Program"
End If
```

4.3 EventCopy

Summary

Event based copy.

Syntax

Boolean EventCopy (*Option*)
Option Integer

Description

Files can be copied to and from a remote machine either directly or through events. Direct copy (file based) allows an application to copy files of any size without doing any I/O itself. Event based copy fires events to copy files and provides the application with more control over the copy operation. The event based file copy allows an application to move data to and from storage other than the local disk, such as through DDE or the clipboard. However, for a remote to remote copy the event mode can **not** be used. To initiate an event based copy, call this method.

Before calling the EventCopy method, the Destination and the Source methods must be called to set the necessary properties. The EventCopy method takes an option parameter (*Option*) and returns a boolean. Please check the reference page of the Options property for a complete listing of the available options. If the file(s) can be successfully copied, 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 copy operation will proceed and an OnDone event will be fired to indicate the completion of the copy operation. 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 EventCopy method) is executed. The application may want to display an error message in the OnError event to inform the user that the copy operation was not successful. Please check the reference page of the OnError event for a complete listing of error codes.

If the EventCopy method is called, then the Remote Copy ActiveX control will not obtain file information or file contents. Instead, it will rely on the application to provide the correct information. In addition, the application must take care of any recursive copy operations. The ActiveX control will fire OnTransfer events to obtain or deliver all the information. The OnTransfer event will contain a TransferType argument to indicate whether the application should provide file information or the next portion of the contents of a file or whether the application is being given file information or the next portion of the contents of a file. Please check the reference page of the OnTransfer event for complete details on event based transfer.

The properties FileName, FileType, FileSize, FileDate, FileTime and FileAttributes need to be set or read during the OnTransfer event for an event based copy. Please check the reference pages of these properties for more details.

To abort the event based copy operation, set the Action property to ACTION_ABORT inside the OnTransfer event or call the Abort method.

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

Example

```
Result = Rcp.EventCopy (OPTIONS_BOTH)
If Result = False Then
    MsgBox "Cannot copy file", 64, "Sample Program"
End If
```

4.4 FileCopy

Summary

File based copy.

Syntax

Boolean FileCopy (*Option*)
Option Integer

Description

Files can be copied to and from a remote machine either directly or through events. Direct copy (file based) allows an application to copy files of any size without doing any I/O itself. Event based copy fires events to copy files and provides the application with more control over the copy operation. The FileCopy method initiates a direct file copy.

Before calling the FileCopy method, the Destination and the Source methods must be called to set the necessary properties. The FileCopy method takes an option parameter (*Option*)and returns a boolean. Please check the reference page of the Options property for a complete listing of the available options. If the file(s) can be successfully copied, 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 copy operation will proceed and an OnDone event will be fired to indicate the completion of the copy operation. 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 FileCopy method) is executed. The application may want to display an error message in the OnError event to inform the user that the copy operation was not successful. Please check the reference page of the OnError event for a complete listing of error codes.

If the FileCopy method is called, then the Remote Copy ActiveX control takes care of obtaining the file information, reading from or writing to files and recursively copying and creating subdirectories. An OnCopy event will be fired for each file or directory that is copied (unless files are copied between two remote machines). The application can use this event to display any status information, such as a progress bar. Please check the reference page of the OnCopy event for more information.

A copy operation in the direct mode can be aborted at any time by setting the Action property to ACTION_ABORT or by calling the Abort method inside the OnCopy event.

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

Example

```
Result = Rcp.FileCopy (OPTIONS_PRESERVE)
If Result = False Then
    MsgBox "Cannot copy file", 64, "Sample Program"
End If
```

4.5 ReceiveB

Summary

Retrieve binary data.

Syntax

Long ReceiveB (*Buffer, Bytes*)

Buffer	Variant
Bytes	Long

Description

The ReceiveB method retrieves the binary data and passes it to the application.

The ReceiveB method takes a buffer (*Buffer*) and the number of bytes to read (*Bytes*) as its parameters and returns the actual number of bytes retrieved. The *Buffer* parameter should be an array of bytes data type. The *Bytes* parameter specifies how many bytes should be retrieved.

If there are less data to be read than the requested *Bytes*, only the available data are read. This is reflected in the return value. If the returned value is less than the requested *Bytes*, that means there are less data available than requested. The application should also check to make sure what is the actual number of bytes read.

Example

Length = Rcp.**ReceiveB** (Buffer, Bytes)

4.6 SendB

Summary

Send binary data.

Syntax

Boolean SendB (*Buffer, Bytes*)
Buffer Variant
Bytes Long

Description

The SendB method sends the binary data over an established connection. The application should not pass more data than the maximum specified by the Bytes argument of the OnTransferB event to this method.

The SendB method takes a buffer (*Buffer*) and the number of bytes to be sent (*Bytes*) as its parameters and returns a boolean. The *Buffer* parameter should be an array of bytes data type. The *Bytes* parameter must contain the number of bytes of data to send. If the data is successfully sent, then the method returns True; otherwise, it returns False. The application should ensure that the method was successfully executed by checking the return value.

Example

```
Result = Rcp.SendB (Buffer, Bytes)
If Result = False Then
    MsgBox "Cannot send binary data", 64, "Sample Program"
End If
```

4.7 Source

Summary

Set the source host ,user, and file names.

Syntax

Boolean Source (*Host, User, File*)

Host	String
User	String
File	String

Description

The Source method sets the SrcHost, SrcUser and SrcFile properties. This method should be called before calling the FileCopy or the EventCopy method (or before setting the Action property to ACTION_COPY or ACTION_EVENT).

The Source method takes a source host name (*Host*), a source user name (*User*) and a source file name (*File*) as its parameters and returns a boolean. The source host name must be set to the name or internet address (in dotted decimal notation) of the source machine. The *User* must be a valid user name to complete the login process. If the source is the local machine, then the *Host* and the *User* parameters must be set to an empty string. The source file name must contain a file or directory name(s).

If the three source properties are successfully set, 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 SrcHost property, the SrcUser property, and the SrcFile property.

Example

```
Result = Rcp.Source ("speedy.distinct.com", "santa", "abc")
If Result = False Then
    MsgBox "Cannot set source information", 64, "Sample Program"
End If
```

Registry Entries

RCOPY registry entries

RcpTimeout REG_DWORD 1-999

Specifies the default timeout in seconds used while connecting to an rcp server. The default value is 20 seconds.

RcpErrorBuffer REG_DWORD 128-32000

Specifies the size of the error buffer. The default value is 2048 bytes.

RcpMaxErrors REG_DWORD 1-999

Specifies the maximum number of errors that are allowed. If the number of errors is greater than this value, then the function will terminate. The default value is 50.