



Getting Started with the CIPOP and CISMTP Controls

Related Topics

The CIMAIL.OCX contains the CIPOP and CISMTP controls which let you add Internet-compliant send and retrieve mail functionality to VB4 applications. The following table identifies the protocol that each control supports and the RFC it complies with:

Icon	Control	Protocol	RFC
	<u>CIPOP</u>	POP3	1725
	<u>CISMTP</u>	SMTP	821

This topic introduces the SMTP and POP3 mail protocols and the Crescent Internet ToolPak mail controls. You can access the following information:

- [What POP3 is](#)
- [What SMTP is](#)
- [What the CIPOP control is](#)
- [What the CISMTP control is](#)
- [What CIPOP programming tasks are](#)
- [What CISMTP programming tasks are](#)

The programming tasks topics identify the major tasks that you can perform with CIPOP and CISMTP, it lists the properties to set and the methods to call, then it illustrates the task with a code sample.

You can use the Mail controls to create applications such as:

- **Mail Filter**
Create a VB4 application that sorts mail based on the sender address (or any other property). When the application receives mail from a specified email address, the application can notify the user. In addition, the application can make mail from this sender available upon arrival, while depositing mail from other senders in a file for later retrieval. It can also designate mail with certain attributes as junk, the application can then delete it on receipt.
- **Web Form Mail**
Create an application that automatically parses mail received from a sendmail CGI script that contains information about users that fill out web site forms, and use the parsed information to populate a database.
- **FTP Event Driven Mail**
Create an application that integrates the Mail control and the FTP control that automatically sends a mail message when a file on the FTP site is updated.



What is POP3?

Related Topics

POP3 stands for Post Office Protocol. POP3 is a standard that defines how remote users, or users that are not always connected to the Internet, can receive electronic mail messages. The POP3 standard is defined in RFC 1725.



What is the CIPOP Control?

Related Topics

The Crescent Internet ToolPak Post Office Protocol (CIPOP) control can retrieve mail from a POP server from within any OLE-compliant container.

CIPOP has a simple programming interface in which you set property values and call methods to perform the desired tasks. The methods (which generally do not take arguments) take the necessary data directly from the property sheet, and submit the requests to the POP3 server. When the server returns its response, the control parses the data and updates the appropriate properties.

The CIPOP control executes the methods as blocking methods. A *blocking method* prevents (or blocks) your application from executing any other code until the function completes. Programming that uses blocking functions is known as polled, or synchronous, programming. Programming that does not use blocking functions is known as event-driven, or asynchronous, programming.

■ CIPOP Programming Tasks

Related Topics

The CIPOP control lets you build POP3 client applications. Each client application must contain code that connects to a POP3 server before it can issue any requests to the server. The connection requirements include:

1. Setting the HostName or HostAddress property and the MailPort property.
2. Calling the ConnectToPOPServer method.

The code looks like this:

```
Form_Load
CIPOP1.HostName = Mypopserver.com
CIPOP1.MailPort = 110
CIPOP1.UserName = MyUserName
CIPOP1.Password = MyPassword
nResult% = CIPOP1.ConnectToPOPServer
```

When the connection attempt succeeds, *nResult%* is an integer that represents the socket number opened by the ConnectToPOPServer method. When the connection attempt fails, *nResult%* is 0 and the WSAError event fires.

Once connected, you can retrieve or delete electronic mail messages.

■ Retrieving a Mail Message Using Low-level Methods

Related Topics

Before you can retrieve messages using the low-level methods, you must first connect to the Mailbox.

1. Set these properties:

HostName (or HostAddress)

MailPort (change from default only if host uses non-standard port)

UserName

Password

2. Call the ConnectToPOPServer method followed by the USER and PASS methods. The USER and PASS methods define the mailbox that you want to retrieve messages.
3. To find out the total number of messages waiting, call the STAT method. You can view the number of messages in the TotalMessages property.

4. Set the MessageNumber property to the message that you want to read, and call the RETR method to retrieve the message. View the message in these properties:

Message

MessageBody

MessageDate

MessageHeader

MessageSubject

5. Call the QUIT method to terminate the connection to the mailbox.

The code looks like this:

Form_Load

```
CIPOP1.HostName = Mypopserver.com
CIPOP1.MailPort = 110
CIPOP1.UserName = MyUserName
CIPOP1.Password = MyPassword
nResult% = CIPOP1.ConnectToPOPServer
```

```
If nResult% > 0 Then
```

```
    CIPOP1.USER
```

```
    CIPOP1.PASS
```

```
    CIPOP1.STAT
```

```
    CIPOP1.MessageNumber = xx
```

```
    CIPOP1.RETR
```

```
    CIPOP1.QUIT
```

```
End If
```

■ Deleting a Mail Message

Related Topics

1. Set these properties:
HostName (or HostAddress)
MailPort (change from default only if host uses non-standard port)
UserName
Password
2. Call the ConnectToPOPServer method followed by the USER and PASS methods. The USER and PASS methods define the mailbox from which you want to delete messages.
3. Set the MessageNumber property to the message that you want to delete, and call the DELE method.
4. Call the QUIT method to quit and update the mailbox to reflect current deletions

The code looks like this:

```
Form_Load
    CIPOP1.HostName = Mypopserver.com
    CIPOP1.MailPort = 110
    CIPOP1.UserName = MyUserName
    CIPOP1.Password = MyPassword
    nResult% = CIPOP1.ConnectToPOPServer

    If nResult% > 0 Then
        CIPOP1.USER
        CIPOP1.PASS
        CIPOP1.MessageNumber = xx
        CIPOP1.DELE
        CIPOP1.QUIT
    End If
```

■ What is SMTP?

Related Topics

SMTP stands for Simple Mail Transfer Protocol. SMTP is a standard that defines how to send electronic mail messages from one Internet user to another. The SMTP standard is defined by RFC 821.

■ What is the CISMTP Control?

Related Topics

The Crescent Internet ToolPak Simple Mail Transfer Protocol (CISMTP) control can send mail to an SMTP server from within any OLE-compliant container. It can also query an SMTP server for the contents of a mail list.

CISMTP has a simple programming interface in which you set property values and call methods to perform the desired tasks. The methods (which generally do not take arguments) instead take the necessary data directly from the property sheet, and submits the request to the server. When the server returns its response, the control parses the data and updates the appropriate properties or ListBox object.

The CISMTP control executes the methods as blocking methods. A *blocking method* prevents (or blocks) your application from executing any other code until the function completes. Programming that uses blocking functions is known as polled, or synchronous, programming.

CISMTP supports two types of methods: low-level and high-level. A *low-level method* performs a single function like providing a single piece of information to the SMTP server, such as a host name. A complete task, therefore, is comprised of multiple low-level method calls. A *high-level method* performs a complete task: It encapsulates all of the low-level methods necessary to perform the complete task into a single method call, thus significantly reducing the amount of code that you must write. CISMTP supports two high-level methods. One high-level method, SendMail, encapsulates the functionality required to send a mail message, and the other, ExpandMailList, encapsulates the functionality required to retrieve the contents of a mail list.

■ CISMTP Programming Tasks

Related Topics

The CISMTP control lets you build SMTP client applications. Each client application must contain code that connects to an SMTP server before it can issue any requests to the SMTP server. The connection requirements include:

1. Setting the HostName **or** HostAddress property and the MailPort property.
2. Calling the ConnectToSMTPServer method.

The code looks like this:

```
CISMTP1.HostName = SMTPServerName  
CISMTP1.MailPort = 25  
nResult% = CISMTP1.ConnectToSMTPServer
```

When the connection attempt succeeds, *nResult%* is an integer that represents the socket number opened by the ConnectToSMTPServer method. When the connection attempt fails, *nResult%* is 0 and the WSAError event fires.

Once connected, you can retrieve the contents of a mail list, or send mail.

■ Retrieving the Contents of a Mail List Using Low-level Methods

Related Topics

This topic describes how to use CISMTP to retrieve the contents of a mail list.

1. Define a connection and return address by setting these properties:

HostName (or HostAddress)

MailPort

Sender

2. Define the Mail List whose contents you want to retrieve by setting the MailList property, and what you want to do with the contents once they are retrieved by setting the ListBoxName property to a listbox object that exists on the form.
3. Call the ConnectToSMTPServer method and the HELO method.

The ConnectToSMTPServer method connects to the SMTP server performs based upon the address you supplied in Step 1, and the HELO method identifies the requesting SMTP server to the receiving SMTP server.

4. Call the EXPN method to request that the SMTP server return the contents of the mail list.
5. Call the QUIT method to close the connection to the SMTP server.

The code to perform the Mail List extraction looks like this:

```
Form_Load
CISMTP1.HostName = MySMTPHost
CISMTP1.MailPort = 25
CISMTP1.Sender = Me
CISMTP1.MailList = MyMailList
Set CISMTP1.ListBoxName = MailListBox
nResult = CISMTP1.ConnectToSMTPServer

If nResult% > 0 Then
    CISMTP1.HELO
    MailListBox.Clear
    CISMTP1.EXP_N
    CISMTP1.QUIT
End If
```

■ Retrieving the Contents of a Mail List Using High-level Methods

Related Topics

This topic describes how to use CISMTP to retrieve the contents of a mail list using the high-level method ExpandMailList.

1. Define a connection and return address by setting these properties:

HostName (or HostAddress)

MailPort

Sender

2. Set these properties to define what you want to retrieve and what you want to do with it:

MailList

ListBoxName

The code looks like this:

```
Form_Load
CISMTP1.HostName = MySMTPHost
CISMTP1.MailPort = 25
CISMTP1.Sender = Me
CISMTP1.MailList = MyMailList
Set CISMTP1.ListBoxName = MailListBox
nResult = CISMTP1.ExpandMailList
```

■ Adding Send Mail Capabilities to an Application Using Low-level Methods

Related Topics

This topic describes how you can use CISMTPs low-level methods to create a Visual Basic application with which you can send mail to any valid email address on the Internet.

1. Define a connection and return address by setting these properties:

DomainName

HostName (or HostAddress)

MailPort

Sender

Recipient

Create the mail message to send by setting these properties:

MessageSubject

MessageBody

2. Call the ConnectToSMTPServer method to connect to the SMTP server identified in the address you supplied in Step 1.
3. Call the HELO method to connect the sending and receiving SMTP servers.
4. Call the MAIL method to identify the mail sender to the SMTP server .
5. Call the RCPT method to identify the recipient to the SMTP server.
6. Call the DATA method to send the message.
7. Call the QUIT method to close the connection to the SMTP server.

The code looks like this:

```
CISMTP1.DomainName = MyDomain
CISMTP1.HostName = MySMTPHost
CISMTP1.MailPort = 25
CISMTP1.Sender = Me
CISMTP1.Recipient = You
CISMTP1.MessageSubject = Schedules
CISMTP1.MessageBody = Lots of Text
nResult% = CISMTP1.ConnectToSMTPServer

If nResult > 0 Then
    CISMTP1.HELO
    CISMTP1.MAIL
    CISMTP1.RCPT
    CISMTP1.DATA
    CISMTP1.QUIT
End If
```

■ Adding Send Mail Capabilities to an Application Using High-level Methods

Related Topics

This topic describes how you can use CISMTPs high-level method, SendMail, to create a Visual Basic application with which you can send mail to any valid email address on the Internet.

1. Define a connection and return address by setting these properties:

DomainName

HostName (or HostAddress)

MailPort

Sender

Recipient

Create the mail message to send by setting these properties:

MessageSubject

MessageBody

7. Call the SendMail method to send the message.

The code looks like this:

```
CISMTP1.DomainName = MyDomain
CISMTP1.HostName = MySMTPHost
CISMTP1.MailPort = 25
CISMTP1.Sender = Me
CISMTP1.Recipient = You
CISMTP1.MessageSubject = Schedules
CISMTP1.MessageBody = Lots of Text
nResult% = CISMTP1.SendMail
```

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

Getting Started with the CIPOP and CISMTP Controls

What is POP3?

What is the CIPOP Control?

CIPOP Programming Tasks

Retrieving a Mail Message Using Low-level Methods

Deleting a Mail Message

What is SMTP?

CISMTP Programming Tasks

Retrieving the Contents of a Mail List Using Low-level Methods

Retrieving the Contents of a Mail List Using High-level Methods

Adding Send Mail Capabilities to an Application Using Low-level Methods

Adding Send Mail Capabilities to an Application Using High-level Methods

Crescent Internet ToolPak Simple Mail Transfer Protocol Control Crescent Internet ToolPak Post Office Protocol Control Reference

[Getting Started with the CIPOP and CSMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CSMTP Control?](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using Low-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

[Getting Started with the CIPOP and CISMTP Controls](#)

[What is POP3?](#)

[What is the CIPOP Control?](#)

[CIPOP Programming Tasks](#)

[Retrieving a Mail Message Using Low-level Methods](#)

[Deleting a Mail Message](#)

[What is SMTP?](#)

[What is the CISMTP Control?](#)

[CISMTP Programming Tasks](#)

[Retrieving the Contents of a Mail List Using Low-level Methods](#)

[Retrieving the Contents of a Mail List Using High-level Methods](#)

[Adding Send Mail Capabilities to an Application Using High-level Methods](#)

[Crescent Internet ToolPak Simple Mail Transfer Protocol Control Reference](#)

[Crescent Internet ToolPak Post Office Protocol Control Reference](#)

The Crescent Mail Controls

Related Topics

The Crescent Mail controls (CIMAIL.OCX) let you add Internet-compliant send and retrieve mail functionality to VB4 applications:

Icon	Objec Name	Description
■	<u>CISMT</u> P	The Crescent Internet Toolpak Simple Mail Control supports the ability to send mail and obtain mail lists from within an OLE container. It complies with the Simple Mail Transfer Protocol (SMTP) described in RFC 821.
■	<u>CIPOP</u>	The Crescent Internet ToolPak Retreive Mail Control control supports the ability to retrieve mail from within an OLE container. It complies with the Post Office Protocol 3 (POP3) described in RFC 959.

You can use the Mail controls to create applications such as:

- Mail Filter (CIPOP)
Create a VB4 application that sorts mail based on the sender address (or any other property). When the application receives mail from a specified user, the application can notify the user. In addition, the application can make mail from this sender available upon arrival, while depositing mail from other senders in a file for later retrieval. It can also designate mail with certain attributes as junk, the application can then delete it on receipt.
- Web Form Mail (CIPOP)
Create an application that automatically parses mail received from users that fill out web site forms, and use the parsed information to populate a database.
- FTP Event Driven Mail (CISMT)P
Create an application that integrates the Mail control and an FTP control that automatically sends a mail message when the FTP site is updated.

■ Crescent Internet ToolPak Simple Mail Transfer Protocol Control

Related Topics

Control File

CIMAIL.OCX

Object Type

CISMTP

Purpose

The Crescent Internet ToolPak Simple Mail Transfer Protocol (CISMTP) control can send mail from within any OLE-compliant container, and can query an SMTP (Simple Mail Transfer Protocol) server for the contents of a mail list.

Properties

About	<u>BCc</u>	<u>CC</u> c
Container	<u>DomainName</u> c	DragIcon
DragMode	Height	HelpWhatsThisID
<u>HostAddress</u> c	<u>HostName</u> c	Index
Left	<u>ListBoxName</u> c	<u>MailList</u> c
<u>MailPort</u> c	<u>MessageBody</u> c	<u>MessageSubject</u> c
Name	Object	Parent
<u>Recipient</u> c	<u>RecvTimeout</u> c	<u>Sender</u> c
<u>SendTimeout</u> c	Tag	Top
Visible	Width	

Events

DragDrop	DragOver	<u>ListBoxPopulated</u> c
<u>PacketReceived</u> c	<u>PacketSent</u> c	<u>SocketClosed</u> c
<u>WSAError</u> c		

Methods

<u>CleanupConnection</u> c	<u>ConnectToSMTPServer</u> c	<u>DATA</u> c
Drag	<u>EXPN</u> c	<u>ExpandMailList</u> c
<u>HELO</u> c	<u>MAIL</u> c	Move
<u>NOOP</u> c	<u>QUIT</u> c	<u>RCPT</u> c
<u>RSET</u> c	<u>SendMail</u> c	ShowWhatsThis
<u>VRFY</u> c	ZOrder	

c A custom or modified property, method, or event.

■ Crescent Internet ToolPak Post Office Protocol Control

Related Topics

Control File

CIMAIL.OCX

Object Type

CIPOP

Purpose

The Crescent Internet ToolPak Post Office Protocol Control (CIPOP) control can be used to retrieve mail from a POP3 (Post Office Protocol Version 3) mail server from any OLE container.

Properties

About	Container	DragIcon
DragMode	Height	<u>HostAddress</u> c
<u>HostName</u> c	Index	Left
<u>ListBoxName</u> c	<u>MailPort</u> c	<u>MessageBody</u> c
<u>Message</u> c	<u>MessageDate</u> c	<u>MessageHeader</u> c
<u>MessageNumber</u> c	<u>MessageSubject</u> c	Name
Object	Parent	<u>Password</u> c
<u>RecvTimeout</u> c	<u>Sender</u> c	<u>SendTimeout</u> c
Tag	Top	<u>TotalMessages</u> c
<u>UserName</u> c	Visible	WhatsThisHelpID
Width	ZOrder	

Events

DragDrop	DragOver	<u>MessageReceived</u> c
<u>PacketReceived</u> c	<u>SocketClosed</u> c	<u>STATReceived</u> c
<u>WSAError</u> c		

Methods

<u>CleanupConnection</u> c	<u>ConnectToPOPServer</u> c	<u>DELE</u> c
Drag	<u>LIST</u> c	Move
<u>NOOP</u> c	<u>PASS</u> c	<u>QUIT</u> c
<u>RETR</u> c	<u>RSET</u> c	ShowWhatsThis
<u>STAT</u> c	<u>USER</u> c	

c A custom or modified property, method, or event.

BC Property

Applies To

CISMTTP

Purpose

The BC (Blind Copy) property sets the address(es) to which you want to send a blind carbon copy of the message.

Syntax

```
[Form.] CISMTTP.BC [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

A blind carbon copy is a carbon copy of a message that the other recipients of the message do not know was sent. To send multiple blind copies use a comma-delimited string of addresses (with no spaces after the commas). For example:

```
anne@abc.com, tom@abc.com, rob@abc.com
```

Set BC before calling the RCPT method

See Also

CC, RCPT

CC Property

Applies To

CISMTTP

Purpose

The CC (carbon copy) property sets the address(es) to which you want to send a carbon copy of a message.

Syntax

```
[Form.] CISMTTP.CC [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

To send multiple carbon copies of the message, use a comma-delimited string of addresses (with no spaces after the commas).

For example:

```
anne@abc.com, tom@abc.com, rob@abc.com
```

Set the CC property before calling the RCPT method.

See Also

BC, RCPT

DomainName Property

Applies To

CISMTTP

Purpose

The DomainName property sets or returns the domain name of the SMTP server to which you want to send a message.

Syntax

```
[ Form. ] CISMTTP.DomainName [ = String$ ]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

This is required for any use of CISMTTP. A computers domain name identifies the SMTP server to the Internet. This address:

username@address.com

has these parts:

Address element	Description
username	Identifies the specific person (or maillist) to whom the mail is sent.
address.com	Identifies the domain name.

Set HostAddress before calling the HELO method.

See Also

HostAddress, HostName, HELO

HostAddress Property

Applies To

CISMTTP, CIPOP

Purpose

The HostAddress property sets or returns the IP address of the SMTP or POP server.

Syntax

```
[Form.] ControlName.HostAddress [ = String$]
```

Data Type

String

Usage

Read/Write at design time.

Comments

The HostAddress identifies the host to the Internet. HostAddress consists of the network number and the local host number. You must use either the HostName or HostAddress to send or retrieve a message.

An example of an IP address is:

198.137.64.1

NOTE You must provide either the HostName or HostAddress property to connect to an SMTP or POP server. When you provide both, the controls use HostName.

Set the HostAddress property before calling the ConnectToSMTPServer method.

See Also

ConnectToSMTPServer, DomainName, HostName

HostName Property

Applies To

CISMTTP, CIPOP

Purpose

The HostName property sets or returns the name of the SMTP or POP server.

Syntax

```
[Form.] ControlName.HostName [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

The HostName is the name that identifies an SMTP or POP server to the Internet. Depending on your connection to the Internet, the HostName might contain simply a machine name like `crescentserver`, or it can be a fully qualified Internet name like that follows this format: `machine.organizationname.com`. Contact your site administrator to determine the correct format to supply.

Set the HostName property before calling the `ConnectToSMTPServer` or `ConnectToPOPServer` method.

NOTE You must use either the HostName or HostAddress property to connect to a mail server. When you provide both, the control uses HostName.

See Also

[ConnectToPOPServer](#), [ConnectToSMTPServer](#), [DomainName](#), [HostAddress](#)

ListBoxName Property

Applies To

CISMTP

Purpose

The ListBoxName property sets the name of a listbox object that can be populated with the contents of a mail list.

Syntax

```
Set [Form.]CISMTP. ListBoxName = ObjectName
```

Data Type

Object

Usage

Read/Write at runtime.

Comments

Set the ListBoxName property before calling the EXPN method. After the ListBox is filled, the ListBoxPopulated event fires.

See Also

EXPN, ListBoxPopulated

MailList Property

Applies To

CISMTTP

Purpose

The MailList property sets the name of the mail list whose members you want to view.

Syntax

```
[Form.] CISMTTP.MailList [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

Set the MailList property before calling the EXPN method. PacketReceived fires when the maillist is received from the SMTP server. You can parse the contents PacketReceived to view the MailList, or if you defined a ListBox object (ListBoxName property), EXPN fills the ListBox object with the expanded contents of the mail list. After the ListBox is filled, the ListBoxesPopulated event fires.

The mail list must reside on the SMTP server to be accessible to the control.

See Also

EXPN, ListBoxesPopulated, ListBoxName, PacketReceived

MailPort Property

Applies To

CISMTTP, CIPOP

Purpose

The MailPort property sets or returns the port used for POP or SMTP services.

Syntax

```
[Form.] ControlName.MailPort [ = Integer% ]
```

Data Type

Integer

Usage

Read/Write at design time and runtime.

Comments

CIPOP and CISMTTP require that you set the MailPort and the HostAddress or HostName properties before initiating a connection to the respective server. The following table identifies the default ports for these services.

Control	Service	Default Port
CIPOP	POP	110
CISMTTP	SMTP	25

See Also

ConnectToPOPServer, ConnectToSMTPServer, HostAddress, HostName

Message Property

Applies To

CIPOP

Purpose

The Message property returns the text of the message retrieved from the POP server.

Syntax

```
[Form.] CIPOP.Message [ = String$]
```

Data Type

String

Usage

Read/Write at runtime and design time.

Comments

Use the RETR method to retrieve the message from the POP server. RETR updates the Message property with the message text it retrieved from the server. The Message property contains the entire message including body text, date, header information and subject. The RETR method also parses the retrieved message and updates these properties: MessageBody, MessageDate, MessageHeader, MessageNumber, and MessageSubject.

See Also

MessageBody, MessageDate, MessageHeader, MessageSubject, RETR

MessageBody Property

Applies To

CIPOP, CISMTTP

Purpose

The MessageBody property returns the text of the message to retrieve (CIPOP) or sets the text of the message to send (CISMTTP).

Syntax

```
[Form.] ControlName.MessageBody [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

This table shows how MessageBody is used by CIPOP and CISMTTP:

Control Name	Description
CIPOP	Use the RETR method to retrieve the message text from the POP server. RETR updates the MessageBody property with the message text. In addition, RETR also parses the message and updates these properties: MessageDate, MessageHeader, and MessageSubject. The unparsed message resides in Message.
CISMTTP	Use the DATA method to send the contents of MessageBody and MessageSubject to the SMTP server for delivery to the address specified by the Recipient property.

See Also

DATA, Message, MessageDate, MessageHeader, MessageSubject, Recipient, RETR

MessageDate Property

Applies To

CIPOP

Purpose

The MessageDate property returns the date of the message retrieved from the POP server.

Syntax

```
[Form.] CIPOP.MessageDate [ = String$]
```

Data Type

String

Usage

Read/Write at design time and at runtime.

Comments

The RETR method retrieves messages from the POP server. It updates the MessageDate property date and time information as included in the retrieved message. The format may vary depending on the tool used to send the message.

RETR also parses the retrieved message and updates these other properties: MessageBody, MessageHeader, and MessageSubject. The unparsed message resides in Message.

See Also

MessageBody, Message, MessageDate, MessageHeader, MessageSubject, Recipient, RETR

MessageHeader Property

Applies To

CIPOP

Purpose

The MessageHeader property returns the header of the message retrieved from the POP server.

Syntax

```
[Form.] CIPOP.MessageHeader [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

The RETR method updates the MessageHeader property with the header of the message retrieved from the POP server. RETR also parses the retrieved message and updates these properties: MessageBody, MessageDate, and MessageSubject. The unparsed message resides in Message.

See Also

MessageBody, Message, MessageDate, MessageSubject, Recipient, RETR

MessageNumber Property

Applies To

CIPOP

Purpose

The MessageNumber property identifies a specific message to the POP server.

Syntax

```
[Form.] CIPOP.MessageNumber [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

MessageNumber can be used to identify a message to the POP server so that you can retrieve it (RETR method), delete it (DELE method), or request its size (LIST method). You must set the MessageNumber property before calling RETR, DELE, and LIST.

The following table lists

This method...	uses the MessageNumber to...
DELE	If the server has a message with that number, it deletes it.
LIST	If the server has a message with that number, it returns the MessageSize to the PacketReceived event.
RETR	If the server has a message with that number, the Message property is updated with the unparsed message. In addition, the message is parsed and the following properties updated: MessageBody, MessageDate, MessageHeader, and MessageSubject properties.

See Also

DELE, LIST, RETR

MessageSubject Property

Applies To

CIPOP, CISMTTP

Purpose

The MessageSubject returns the subject of the message retrieved from the POP server (CIPOP) or sets the subject of the message to send (CISMTTP)

Syntax

```
[Form.] ControlName.MessageSubject [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

This table how MessageSubject is used by CIPOP and CISMTTP:

Control Name	Description
CIPOP	The RETR method updates the MessageSubject. RETR also updates these properties: MessageBody, MessageDate, and MessageHeader. The unparsed message resides in Message.
CISMTTP	Set MessageSubject (and MessageBody) before calling the DATA method. The DATA method delivers the contents of MessageSubject and MessageBody to the SMTP server who then delivers the message to the address specified by the Recipient property.

See Also

DATA, MessageBody, Message, MessageDate, MessageHeader, Recipient, RETR

Password Property

Applies To

CIPOP

Purpose

The Password property sets the password used to access the mailbox from which you are retrieving messages.

Syntax

```
[Form.] CIPOP.Password[ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

This property requires a value.

Recipient Property

Applies To

CISMTTP

Purpose

The Recipient property sets the address where you want to send a message.

Syntax

```
[Form.] CISMTTP.Recipient [ = String$ ]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

Set Recipient before calling the RCPT method. Recipient is a required property. To send the message to multiple recipients, type the addresses in a comma-delimited string (with no spaces after the commas) as shown below:

```
anne@abc.com,tom@abc.com,rob@abc.com
```


RecvTimeout Property

Applies To

CIPOP, CISMTP

Purpose

The RecvTimeout property represents the amount of time (in milliseconds) that the control waits for a response from the Mail host.

Syntax

```
[Form.] CISMTP.RecvTimeout [ = integer%]
```

Data Type

Integer

Usage

Read/Write at design time and runtime.

Comments

The default is -1. Leave the property at the default unless a method is failing or timing out or if the SMTP or POP server response time is particularly slow. Setting this value replaces the controls internal timeout level. Setting this value higher than is necessary can degrade performance.

Sender Property

Applies To

CIPOP, CISMTTP

Purpose

The Sender property sets (CIPOP) or returns (CISMTTP) the message senders address.

Syntax

```
[Form.] ControlName.Sender [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

This table describes how CIPOP and CISMTTP use the Sender property:

Control Name	Description
CIPOP	The RETR method updates Sender with the senders address as retrieved from the POP server. RETR also parses the retrieved message and updates these properties: MessageDate, MessageHeader, and MessageSubject. The unparsed message resides in Message.
CISMTTP	Set the Sender property before calling the MAIL method. This is a required property for CISMTTP when you are sending a message.

See Also

DATA, MAIL, Message, MessageDate, MessageHeader, MessageSubject, Recipient, RETR

SendTimeout Property

Applies To

CIPOP, CISMTP

Purpose

The SendTimeout property represents the amount of time (in milliseconds) that the control waits for a response from the Mail host.

Syntax

```
[Form.] ControlName.SendTimeout [ = integer% ]
```

Data Type

Integer

Usage

Read/Write at design time and runtime.

Comments

The default is -1. Leave the property at the default unless a method is failing or timing out or if the SMTP or POP server response time is particularly slow. Setting this value replaces the controls internal timeout level. Setting this value higher than is necessary can degrade performance.

TotalMessages Property

Applies To

CIPOP

Purpose

The TotalMessages property returns the number of messages in the connected mailbox on the POP server.

Syntax

```
[Form.] CIPOP.TotalMessages [ = Integer%]
```

Data Type

Integer

Usage

Read/Write at design time and runtime.

Comments

The STAT method updates the TotalMessages property. TotalMessages includes only the messages that have not been read.

See Also

STAT

UserName Property

Applies To

CIPOP

Purpose

The UserName property sets the username associated with the mailbox on the POP server.

Syntax

```
[Form.] CIPOP.UserName [ = String$]
```

Data Type

String

Usage

Read/Write at design time and runtime.

Comments

Set the UserName property before calling the USER method. It is a required property.

See Also

PASS, Password, USER

ListBoxPopulated Event

Applies To

CISMTP

Purpose

The ListBoxPopulated event fires when the ListBox object has been filled.

Syntax

```
Sub CISMTP_ListBoxPopulated()
```

Comments

The EXPN method returns a list of users associated with a specified MailList. When the ListBoxName property is set before calling EXPN, the returned list is automatically placed in the ListBox object and the ListBoxPopulated event fires.

See Also

EXPN, ListBoxName

PacketReceived Event

Applies To

CIPOP, CISMTTP

Purpose

The PacketReceived event fires when the control receives a packet from the server.

Syntax

```
Sub ControlName_PacketReceived(ByVal Packet As String)
```

Comments

The *Packet* contains data received from the server; its contents vary based on the method used to obtain the packet.

The CIPOP control fires PacketReceived when the POP server sends a response to the container application indicating that a message has been received.

The CISMTTP control fires PacketReceived when the SMTP server sends a response to the container application indicating that a message has been sent.

See Also

PacketSent

PacketSent Event

Applies To

CISMTTP

Purpose

The PacketSent event fires when CISMTTP sends a packet to the SMTP server.

Syntax

```
Sub ControlName_PacketSent (ByVal bytes_out As Integer)
```

Comments

The value of *bytes_out* is the number of bytes in the current packet.

See Also

PacketReceived

MessageReceived Event

Applies To

CIPOP

Purpose

The MessageReceived event fires when CIPOP has received and parsed a message.

Syntax

```
Sub CIPOP_MessageReceived()
```

Comments

Messages are retrieved from the POP server in response to the RETR method. The retrieved message is parsed to these properties: MessageBody, MessageDate, MessageHeader, and MessageSubject. The unparsed message resides in Message.

See Also

MessageBody, Message, MessageDate, MessageHeader, MessageSubject, RETR

SocketClosed Event

Applies To

CIPOP, CISMTTP

Purpose

The SocketClosed event fires when the socket closes.

Syntax

```
Sub ControlName_SocketClosed()
```

Comments

The socket should remain open during program execution. If the SocketClosed event fires while your application is running, an error condition has occurred.

See Also

CleanupConnection

STATReceived Event

Applies To

CIPOP

Purpose

The STATReceived event fires when theTotalMessages property is updated.

Syntax

```
Sub CIPOP_STATReceived()
```

Comments

The STAT method updates the TotalMessages property

See Also

STAT, TotalMessages

WSAError Event

Applies To

CIPOP, CISMTP

Purpose

The WSAError event fires when the control receives a Winsock error.

Syntax

```
Sub ControlName_WSAError(ByVal error_number As Integer)
```

Comments

Use the WSAError event to monitor the Winsock activity. The *error_number* identifies the Winsock error that occurred. The errors are listed in the CITPAK.BAS:

See Also

CITPAK.BAS

CleanupConnection Method

Applies To

CISMTTP, CIPOP

Purpose

The CleanupConnection method closes the socket and cleans up Winsock.

Syntax

ControlName.CleanupConnection

Data Type

None

Comments

Use CleanupConnection to clear errors that might occur when a send or receive message fails or in situations when the socket does not close properly, and you need to re connect to the server. Use this method when all else fails.

See Also

ConnectToPOPServer, ConnectToSMTPServer, WSAError

ConnectToPOPServer Method

Applies To

CIPOP

Purpose

The ConnectToPOPServer method establishes a connection to a POP server.

Syntax

```
nResult% = [Form.] CIPOP.ConnectToPOPServer
```

Data Type

Integer

Comments

You must establish a connection to the POP server before the CIPOP control can retrieve or delete mail messages. To establish a connection, you must set the HostName **or** HostAddress and the MailPort properties.

When the connection attempt succeeds, *nResult%* is an integer that represents the socket number opened by the ConnectToPOPServer method. When the connection attempt fails, *nResult%* is 0 and the WSAError event fires.

After a successful connection, the POP server expects the USER and PASS methods.

See Also

HostAddress, HostName, MailPort, PASS, USER, WSAError

ConnectToSMTPServer Method

Applies To

CISMTTP

Purpose

The ConnectToSMTPServer method establishes a connection to the mail senders SMTP server.

Syntax

```
nResult% = [Form.] CISMTTP.ConnectToSMTPServer
```

Data Type

Integer

Comments

You must establish a connection to the SMTP server before CISMTTP can send mail or retrieve the contents of MailLists. To establish a connection, set the HostName **or** HostAddress and the MailPort properties.

When the connection attempt succeeds, *nResult%* is an integer that represents the socket number opened by the ConnectToSMTPServer method. When the connection attempt fails, *nResult%* is 0 and the WSAError event fires.

Once you are connected, the SMTP server expects the HELO method followed by the MAIL method (to perform send mail functions), or the EXPN method (to obtain the contents mail list).

See Also

HostAddress, HostName, MailPort, WSAError

DATA Method

Applies To

CISMTTP

Purpose

The DATA Method sends the message to the SMTP server.

Syntax

nResult% = [*Form.*] *CISMTTP*.DATA

Data Type

Integer

Comments

Set the MessageSubject and MessageBody properties before calling the DATA method.

When the DATA method is successful *nResult%* is 0. Invoke the DATA method after the RCPT method.

See Also

MessageBody, MessageSubject, RCPT

DELE Method

Applies To

CIPOP

Purpose

The DELE method deletes a specific message from the mailbox.

Syntax

nResult% = [*Form.*] *CIPOP.DELE*

Data Type

Integer

Comments

Set the MessageNumber property before calling DELE. The MessageNumber identifies the number to delete. When DELE is successful, *nResult%* is 0. The DELE method can fail for the following reasons:

- The MessageNumber does not exist
- The MessageNumber is already be marked for deletion
- The user does not have write privileges to the mailbox

See Also

MessageNumber

EXPN Method

Applies To

CISMTTP

Purpose

The EXPN method returns a list of subscribers to a mail list.

Syntax

nResult% = [*Form.*] *CISMTTP*.EXPN

Data Type

Integer

Comments

Set the MailList (required) and the ListBoxName (optional) property before calling EXPN. Call EXPN after calling the ConnectToSMTPServer and HELO methods.

When EXPN is successful, the *nResult%* is 0.

See Also

ConnectToSMTPServer, HELO, ListBoxName

ExpandMailList Method

Applies To

CISMTTP

Purpose

The ExpandMailList method retrieves a mail list from an SMTP server.

Syntax

```
nResult% = CISMTTP.ExpandMailList
```

Data Type

Integer

Comments

If *nResult%* is 0, ExpandMailList succeeded; any other value indicates failure. When ExpandMailList succeeds, the mail list is returned to the object named by the ListBoxName property.

ExpandMailList encapsulates these four CISMTTP methods required to send a mail message (in this order): ConnectToSMTPServer, HELO, EXPN, QUIT.

NOTE If any of the encapsulated methods fails, then ExpandMailList also fails.

Set these properties before calling ExpandMailList: HostName (or HostAddress), MailList, MailPort, ListBoxName.

See Also

ConnectToSMTPServer, EXPN, HELO, HostAddress, HostName, ListBoxName, MailList, MailPort, QUIT

HELO Method

Applies To

CISMTTP

Purpose

The HELO method identifies the senders SMTP server to the recipients SMTP server.

Syntax

```
nResult% = [Form.] CISMTTP.HELO
```

Data Type

Integer

Comments

Set *HostAddress* or *HostName* before calling the HELO method. When the HELO method succeeds, *nResult%* is 0. Use HELO after the *ConnectToSMTPServer* method.

See Also

ConnectToSMTPServer, *HostAddress*, *HostName*

LIST Method

Applies To

CIPOP

Purpose

The LIST method requests that the POP server return the size of a specific message or the size of all messages in the mailbox. The specific message to return is the value identified by the MessageNumber property.

Syntax

```
nResult = [Form.] CIPOP.LIST
```

Data Type

Integer

Comments

Set the MessageNumber (optional) before calling the LIST method.

When the LIST method succeeds, *nResult%* is 0.

LIST returns the size of the associated message. If the requested message is marked for deletion or does not exist, the POP server is unable to return the size. If no MessageNumber is supplied, LIST returns the size of all messages in the mailbox. The size is returned to the PacketReceived event.

See Also

MessageNumber

MAIL Method

Applies To

CISMTTP

Purpose

The MAIL method forwards the value of the Sender property to the SMTP server.

Syntax

```
nResult = [Form.]CISMTTP.MAIL
```

Data Type

Integer

Comments

When the MAIL method succeeds, the nResult% is 0. Use MAIL after successfully invoking the ConnectToSMTPServer and HELO methods.

See Also

ConnectToSMTPServer, HELO, Sender

NOOP Method

Applies To

CIPOP, CISMTP

Purpose

The NOOP method indicates whether the connection to the POP or SMTP server exists.

Syntax

```
nResult = [Form.] ControlName.NOOP
```

Data Type

Integer

Comments

When the connection to a server is active, the return code (*nReturn*) is 0. Use NOOP in applications where the connection to the server remains for an extended period (for example, longer than it takes to send or retrieve a single message).

See Also

ConnectToSMTPServer, HELO

PASS Method

Applies To

CIPOP

Purpose

The PASS method sends the contents of the Password property to the POP server for verification that the password is valid for the mailbox (identified by the USER method) from which you want to retrieve messages.

Syntax

```
nResult = [Form.] CIPOP.PASS
```

Data Type

Integer

Comments

When the PASS method succeeds, *nResult%* is 0. Call PASS after successfully calling the ConnectToPOPServer and USER methods.

Once the PASS method succeeds, the POP server locks the mailbox so that other processes cannot change or delete messages in the mailbox. If the mailbox is locked by another process, the PASS method fails. The QUIT method unlocks the mailbox.

See Also

ConnectToPOPServer, Password, USER, UserName

QUIT Method

Applies To

CIPOP, CISMTP

Purpose

The QUIT method terminates the connection to the POP server or SMTP server.

Syntax

```
nResult = [Form.] ControlName.QUIT
```

Data Type

Integer

Comments

Invoke the QUIT method after youve completed the mail transactions that you wanted to perform. QUIT notifies the server to close the connection. When the connection closes, the SocketClosed event fires.

With CIPOP, if you invoke the QUIT method after marking messages for deletion, the POP server deletes the marked messages and updates the mailbox. Once the messages are deleted, the POP server unlocks the mailbox and closes the TCP/IP connection.

If you invoke QUIT after successfully invoking USER and PASS, but before any other methods, the POP server does not update the mailbox. If the connection terminates for any reason other than the QUIT method, the POP server does not delete messages or update the mailbox.

See Also

ConnectToPOPServer, ConnectToSMTPServer, PASS, USER

RCPT Method

Applies To

CISMTP

Purpose

The RCPT method sends the contents of the Recipient property to the SMTP host server.

Syntax

nResult = [*Form.*] *CISMT*P.RCPT

Data Type

Integer

Comments

When the RCPT method succeeds *nResult*% is 0. Call RCPT after calling the ConnectToSMTPServer, HELO and MAIL methods.

See Also

ConnectToSMTPServer, HELO, MAIL, Recipient

RETR Method

Applies To

CIPOP

Purpose

The RETR method retrieves a message from the POP server.

Syntax

nResult = [*Form.*] *CIPOP*.RETR

Data Type

Integer

Comments

Set the MessageNumber property before calling the RETR method.

The RETR method updates the Message property. It contains the entire message retrieved from the POP server including body text, date, header information and subject. RETR also parses the retrieved message and updates these properties: MessageBody, MessageDate, MessageHeader, MessageNumber, and MessageSubject. If the message contains zero characters, the POP server closes the connection. When *nResult%* is 0, the RETR method succeeded.

Call RETR after the USER, PASS, and LIST methods.

See Also

LIST, MessageBody, Message, MessageDate, MessageHeader, MessageNumber, MessageSubject, PASS, RETR, USER

RSET Method

Applies To

CIPOP

Purpose

RSET returns the mailbox to the state that existed when you connected to it.

Syntax

```
nResult = [Form.] CIPOP.RSET
```

Data Type

Integer

Comments

When you call RSET, any messages marked for deletion are unmarked. When *nResult%* is 0, the RSET method succeeded.

RSET Method

Applies To

CISMTF

Purpose

The RSET method terminates the current mail transaction.

Syntax

nResult = [*Form.*] *CISMTF*.RSET

Data Type

Integer

Comments

When *nResult%* is 0, RSET succeeded.

SendMail Method

Applies To

CISMTTP

Purpose

The SendMail method sends a mail message.

Syntax

```
nResult% = CISMTTP.SendMail
```

Data Type

Integer

Comments

If *nResult%* is 0, SendMail succeeded; any other value indicates failure.

SendMail encapsulates these six CISMTTP methods required to send a mail message (in this order): ConnectToSMTPServer, HELO, RCPT, MAIL, DATA, QUIT.

NOTE If any of the encapsulated methods fails, then SendMail also fails.

Set these properties before calling SendMail: HostName (or HostAddress), MailPort, Recipient, Sender. The following properties can also optionally be set: BC, CC, DomainName, MessageBody, MessageSubject.

See Also

BC, CC, ConnectToSMTPServer, DATA, DomainName, HELO, HostAddress, HostName, MAIL, MailPort, MessageBody, MessageSubject, QUIT, RCPT, Recipient, Sender

STAT Method

Applies To

CIPOP

Purpose

The STAT method updates the TotalMessages property with the number of unread messages in the mailbox.

Syntax

nResult% = [*Form.*] *CISMTP*.STAT

Data Type

Integer

Comments

Call the STAT method after successfully connecting to and locking the mailbox on the POP server with the USER and PASS methods. Messages in the mailbox that are marked for deletion are not included in the value returned to TotalMessages. When *nResult%* is 0, STAT succeeded.

See Also

PASS, TotalMessages, USER

USER Method

Applies To

CIPOP

Purpose

The USER method passes identifies the mailbox from which you want to retrieve messages.

Syntax

```
nResult% = [Form.]CIPOP.USER
```

Data Type

Integer

Comments

Set the UserName property before calling the USER method. When the USER method succeeds, *nResult%* is 0.

Call USER after calling the ConnectToPOPServer method. Once USER and PASS are successfully invoked, the POP server locks the mailbox to access by other processes.

See Also

ConnectToPOPServer, PASS, Password, Username

VRFY Method

Applies To

CISMTP

Purpose

The VRFY method lets you verify whether a specified name exists on the system.

Syntax

nResult% = [*Form.*] *CISMTP.VRFY*

Data Type

Integer

Comments

Set the Recipient property before calling the VRFY method. When the user does exist, the data is available in the PacketReceived event. The VRFY method succeeded when *nResult%* is 0.

See Also

Recipient

■ CISMTP General Tab

Applies To

CISMTP.OCX

This tab lets you set values for these properties:

Configuration-related properties:

DomainName

HostName

HostAddress

MailPort

Sender

Message/Mail List related properties:

Recipient

Subject

Message

CC

BC

MailList

■ CISMTP SoundEvents Tab

Applies To

CISMTP.OCX

This custom tab lets you assign sounds to custom events. The following lists the hidden properties that define the sound object and the event to which they are applied:

To add sound to this event

Set this hidden property...

ListBoxPopulated

ListBoxPopulatedWAV

PacketReceived

PacketReceivedWAV

PacketSent

PacketSentWAV

SocketClosed

SocketClosedWAV

WSAError

WSAErrorWAV

You can set the sound associations at design time or runtime. You can test the sounds from within the property page. At runtime, an event is fired and a sound is played asynchronously where an association is made.

To associate a sound, you can type in the full qualified path for any events you want to associate or click the Browse... button and select it from the dialog box. After you enter a sound, you can click the test button to hear it. If the sound file is invalid or missing you will hear nothing when you click Test.

■ CIPOP General Tab

Applies To

CIPOP.OCX

This custom tab sets values for these properties:

HostName

HostAddress

MailPort

MessageNumber

■ CIPOP SoundEvents Tab

Applies To

CIPOP.OCX

This custom tab lets you assign sounds to custom events. The following lists the hidden properties that define the sound object and the event to which they are applied:

To add sound to this event

Set this hidden property...

MessageReceived

MessageReceivedWAV

PacketReceived

PacketReceivedWAV

PacketSent

PacketSentWAV

SocketClosed

SocketClosedWAV

STATReceived

STATReceivedWAV

WSAError

WSAErrorWAV

You can set the sound associations at design time or runtime. You can test the sounds from within the property page. At runtime, an event is fired and a sound is played asynchronously where an association is made.

To associate a sound, you can type in the full qualified path for any events you want to associate or click the Browse... button and select it from the dialog box. After you enter a sound, you can click the test button to hear it. If the sound file is invalid or missing you will hear nothing when you click Test.

