WSH VBScript WMI FSO
ADSI CDO HTA CGI Perl

300165
Systems Administration Programming

```
set objWMI = GetObject("winmgmts:\\.\root\cimv2")
set fso = CreateObject("Scripting.FileSystemObject")
```

Lecture 8
E-Mail Automation

Print page

In the previous lectures, we have learnt the basic language components of VBScript and the key scripting technologies for Windows. In this and the next lectures we apply the scripting technologies to extended Windows system administration.

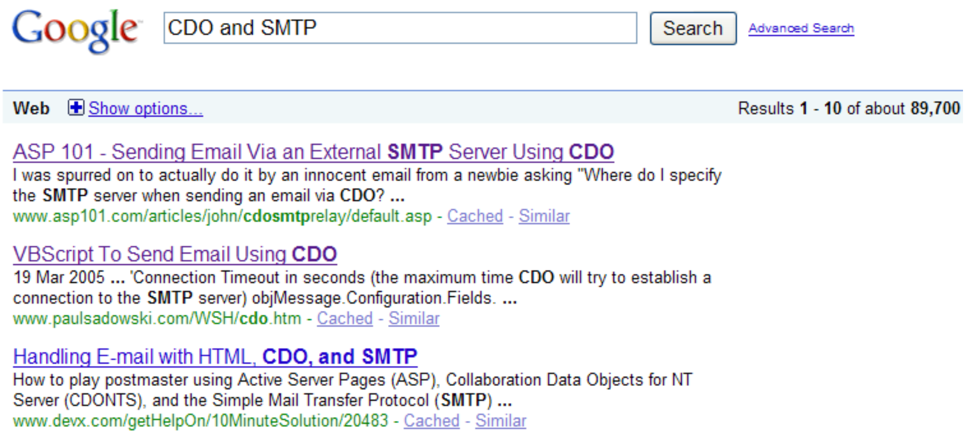
Nowadays almost all of us send emails using email softwares, such as Office Outlook, Thunderbird, Pine, Webmail and etc. These popular email softwares have to be used manually. Assume that we are going to send monthly bills to thousands of our customers via emails. Most email softwares cannot do the job efficiently. In this lecture, we introduce the techniques for generating and delivering emails automatically.

Key words

CDO (Collaboration Data Objects), SMTP (Simple Mail Transfer Protocol), Local SMTP Server, Remote SMTP Server, TCP/IP, Port Number

Reference to textbook chapters

The topic is not covered by the textbook. Part of the materials of this lecture were taken from Jeff Fellinge's book "IT Administrator's Top 10 Introductory Scripts for Windows" and Paul Sadowski's website "http://www.paulsadowski.com/". You can find plenty of related scripts from the Internet.



Results **1** - **10** of about **89,700**

ASP 101 - Sending Email Via an External **SMTP** Server Using **CDO**
I was spurred on to actually do it by an innocent email from a newbie asking "Where do I specify the **SMTP** server when sending an email via **CDO**? ...
www.asp101.com/articles/john/**cdosmtp**relay/default.asp - Cached - Similar

VBScript To Send Email Using **CDO**
19 Mar 2005 ... 'Connection Timeout in seconds (the maximum time **CDO** will try to establish a connection to the **SMTP** server) objMessage.Configuration.Fields. ...
www.paulsadowski.com/WSH/**cdo**.htm - Cached - Similar

Handling E-mail with HTML, **CDO, and SMTP**
How to play postmaster using Active Server Pages (ASP), Collaboration Data Objects for NT Server (CDONTS), and the Simple Mail Transfer Protocol (**SMTP**) ...
www.devx.com/getHelpOn/10MinuteSolution/20483 - Cached - Similar

Email scripts

If you have installed SMTP service on your computer, sending an email programmatically is extremely easy:

```
Set objMsg = CreateObject("CDO.Message")
objMsg.Subject = "subject of the message here"
objMsg.From = "your email address here"
objMsg.To = "addressee's email address here"
objMsg.TextBody = "the message text here"
objMsg.Send
```

Download the code mail1.vbs and try to run it. If it is not runnable, most likely you do not have a local SMTP or it has not been configured properly (See How to install and Configure SMTP Virtual Servers).

In such a case, the best way to go is specify a remote SMTP server to deliver your message (say the school email server). Try the following script mail2.vbs:

```
Dim objMsg, objConfiguration, objConfigFields
Set objConfiguration = CreateObject("CDO.Configuration")
Set objConfigFields = objConfiguration.Fields

objConfigFields.Item("http://schemas.microsoft.com/cdo/" & _
"configuration/sendusing") = 2
objConfigFields.Item("http://schemas.microsoft.com/cdo/" & _
"configuration/smtpserverport") = 25
'replace the SMPT server address with your SMTP address
objConfigFields.Item("http://schemas.microsoft.com/cdo/" & _
"configuration/smtpserver") = "mail.westernsydney.edu.au"
'or try SCEM's remote SMTP server "mail.scem.westernsydney.edu.au"
objConfigFields.Update

Set objMsg = CreateObject("CDO.Message")
objMsg.Configuration = objConfiguration
objMsg.Subject = "subject of the message here"
objMsg.From = "your email address here"
objMsg.To = "addressee's email address here"
objMsg.TextBody = "the message text here"
objMsg.Send
```

Note that if you use WSU' or SCEM's remote SMTP server, your computer needs to be connected to the school network (you can use vpn if you are working at home).

Simple Mail Transfer Protocol (SMTP) and Collaboration Data Objects (CDO)

There are two basic concepts that are crucial for the understanding of the above code: CDO and SMTP.

Firstly, let's see how an email message is delivered. Typically an email message is delivered across the Internet under the Simple Mail Transfer Protocol (SMTP). Delivery of a message is initiated by transferring the message to a designated SMTP server. Based on the domain name of the recipient e-mail address, the SMTP server initiates communications with a Domain Name System (DNS) server, which looks up and then returns the host name of the destination SMTP server for that domain.

The originating SMTP server communicates with the destination SMTP server directly through Transmission Control Protocol/Internet Protocol (TCP/IP) on port 25. If the user name of the recipient e-mail address matches one of the authorized user accounts on the destination server, the original e-mail message is transferred to that server, waiting for the recipient to pick up the message through a client program.

In the case where the originating SMTP server cannot communicate directly with the destination server, the SMTP service can transfer messages through one or more intermediate relay SMTP servers. A relay server receives the original message and then delivers it to the destination server, or redirects it to another relay server. This process is repeated until the message is delivered or a designated timeout period passes.

Next, let's see how to compose an email message. An email message consists of the following components:

- To: receiver's email address.
- From: sender's email address
- Subject: a brief description of the message
- Cc: carbon copy
- Bcc: blind carbon copy
- Attachments: attached files

In the example code above, we use Microsoft Collaboration Data Objects (CDO) to create an email message and specify a SMTP service. Collaboration Data Objects are a high-level set of COM objects that allow, among other functionalities, easily access of the email system embedded in the Microsoft Windows products.

CDO Configuration

In order to use CDO to send a message, we first need to create a CDO configuration object:

```
Set objConfiguration = CreateObject("CDO.Configuration")
```

A CDO Configuration object contains information such as the method used to deliver messages, the paths to the pickup directories for SMTP and Network News Transfer Protocol (NNTP) servers, a user's mailbox Uniform Resource Identifier (URI), and so forth. The following setting specifies the SMTP server we are going to use for sending our message:

```
Set objConfigFields = objConfiguration.Fields
objConfigFields.Item("http://schemas.microsoft.com/cdo/" & _
"configuration/sendusing") = 2
objConfigFields.Item("http://schemas.microsoft.com/cdo/" & _
"configuration/smtpserverport") = 25
'replace the SMPT server address with your SMTP address
objConfigFields.Item("http://schemas.microsoft.com/cdo/" & _
"configuration/smtpserver") = "mail.westernsydney.edu.au"
'or try SCEM's remote SMTP server "mail.scem.westernsydney.edu.au"
objConfigFields.Update
```

The three configuration fields that this script uses are `sendusing`, `smtpserverport`, and `smtpserver`. The configuration field `sendusing` instructs CDO whether to send the message using the local SMTP service (`sendusing = 1`), send to a remote SMTP (`sendusing = 2`) or send using an Exchange Server mail submission URI (`sendusing =3`).

The field `smtpserverport` specifies the SMTP port to use, which is 25, by default. If your remote SMTP server listens on a different port, you can configure CDO to send to that port instead.

The last configuration field specifies the SMTP server. You should fill in the name of the SMTP server you are going to use to send messages. If you run your program in a SCEM lab, you could use the school email server `mail.scem.westernsydney.edu.au`. This may not work at ITD computer labs, where you should be able to use `mail.westernsydney.edu.au`.

Lastly, we must update the configuration by calling the Update method.

More and more administrators are restricting access to their servers to control spam or limit which users may utilize the server. Adding the following lines to the Mail.vbs if your SMTP server uses the basic authentication, the most commonly used authentication method.

```
'Type of authentication, NONE, Basic (Base64 encoded), NTLM
objconfigFileds.Item("http://schemas.microsoft.com/cdo/configuration/smtpauthenticat
e") = cdoBasic

'Your UserID on the SMTP server
objconfigFileds.Item("http://schemas.microsoft.com/cdo/configuration/sendusername")
= "youruserid"

'Your password on the SMTP server
objconfigFileds.Item("http://schemas.microsoft.com/cdo/configuration/sendpassword")
= "yourpassword"

'Use SSL for the connection (False or True)
objconfigFileds.Item("http://schemas.microsoft.com/cdo/configuration/smtpusessl") =
False
```

For a full list of the CDO configuration fields, please visit CDO Messaging Configuration. Note that if you do not set a CDO configuration, then CDO sets one for you prior to sending out each message using the default values.

Creating an email message

To create an email message, we first create an CDO message object:

```
Set objMessage = CreateObject("CDO.Message")
```

Secondly, we set this message to use the above setting of CDO configuration to send:

```
objMsg.Configuration = objConfiguration
```

Next, we fill-in Sender, Subject and Recipient fields of the headers and the body text which can be either plain text or HTML.

```
objMsg.Subject = "subject of the message here"
objMsg.From = "your email address here"
objMsg.To = "addressee's email address here"
objMsg.TextBody = "the message text here"
```

Finally, you use the Send method to send the message.

```
objMsg.Send
```

Manipulating files with an email

You might like to edit an email message body in a separate file. The following function can be used to load a text file to an email message:

```
function ReadTextFile(strFileName)
 Dim objFSO, objTextFile, strReadLine
 Set objFSO = CreateObject("Scripting.FileSystemObject")
 Set objTextFile = objFSO.OpenTextFile(strFileName,1)
 do while not objTextFile.AtEndOfStream
  strReadLine = strReadLine + objTextFile.ReadLine()
 loop
 objTextFile.Close()

 ReadTextFile = strReadLine
end function
```

You can attach files to an email message as email attachments. See the code for a comprehensive example Mail.vbs. Please note, when using the AddAttachment method in your scripts you must use a fully qualified pathname as the argument to the method. Using just a file name or a relative path will produce the error " The specified protocol is unknown".

If you want to send a message to a large group of persons, say sending an announcement, you can put all the receivers' addresses in a text file and load them to the email message you want to send by using a similar function as above. Even more, you can connect to a database to collect information from the database, generate email messages and send them automatically. Paul Sadowski's website "http://www.paulsadowski.com/" lists a set of sample code to show how to do this.