

```
!=====
!   C O M M E N T S
!=====
!
! Program: SENDMAIL
! Version: 2.1
!
! Author: Bryce Martens
!
! OPEN SOURCE
!
! Requires: Comet Build 294 or greater
!           Type 3 gateway
!           Internet connection to SMTP server
!
! Description:
!
! This program demonstrates how to send e-mail, via the Winsock gateway, to
! an SMTP mail server (and on to a destination e-mail address). This program
! uses the commands specified in RFC 821, the Simple Mail Transfer Protocol
! specification.
!
! This program reads data from a text file, hereinafter called the "e-mail
! interface file." This file must be prepared prior to running SENDMAIL.
! See below for the specifications of the e-mail interface file.
!
! This is a hybrid program (i.e., a program/subprogram).
!
! 1. When you run this program from the READY prompt, it prompts for
!    the name and directory of an e-mail interface file.
!
! 2. When you run this program as a subprogram (via an ENTER statement),
```

! the calling program must pass the filename, directory name, and a flag
! to this program via the input/output buffer.

!
! Here's the general setup for a calling program:

- !
! a. create a text file (any name, any accessed directory)
! b. write records to the text file (see format information below)
! c. write the following fields to the I/O buffer:
! 1. name of e-mail interface file (12 bytes; 8.3 format)
! 2. directory where e-mail interface file is located (3 bytes)
! 3. display flag (3 bytes)
! (see below for more information)
! d. ENTER "SENDMAIL"
! e. optional: erase the text file

!
! The e-mail interface file must contain the following records:

! Record Contents

! -----

! 1 CONNECT SMTP-server MAIL

!
! This is a Winsock gateway command. It connects the
! SENDMAIL program to the MAIL port on the SMTP server.
! SMTP-server is the name of the SMTP server you are
! using (also known as the "outgoing mail server".) For
! example, if you are connecting to an SMTP server at
! ABC123.COM (a make believe domain name), this record would
! contain:

!
! CONNECT MAIL.ABC123.COM MAIL

!
! 2 HELO domain-name

!
!
! This is an SMTP command. Domain-name is your domain
! name or server name, although the specific value does
! seem to effect the process of sending e-mail. You may set
! this to your domain name, your ISP's domain name, or any
! other value you choose. For example, this record can contain:
!
! HELO MYDOMAIN.COM
!
! 3 MAIL FROM: e-mail address of sender
!
! This is an SMTP command.
! E-mail address is expressed in standard form: name@domain
!
! 4 RCPT TO: e-mail address of recipient
!
! This is an SMTP command.
! E-mail address is expressed in standard form: name@domain
!
! Note: If there are multiple recipients, include multiple
! "RCPT TO:" commands in the e-mail interface file.
!
! 5 DATA
!
! This is an SMTP command. It signifies that the following records
! contain the e-mail data.
!
! 6-n The e-mail data, including the text of the message,
! header lines (optional), and attachment data (optional).
!
! This demo program supports text lines up to 254 characters long.
! There is no limit to the number of lines that may be sent.

Sample headers include:

Date: today's-date

This header is optional. However, most e-mail client software displays this field as the date of a message. Therefore, we recommend using it. The date should be formatted as follows:

Day, DD-Month-YYYY HH:MM:SS GMT

This date/time format is available using the following code in the program that builds the e-mail interface file:

```
NUM2DATE (DATE2NUM (DATE$,0),4)
```

From: "name-of-sender" <e-mail address of sender>

This header is optional. However, most e-mail client software displays this field at the beginning of a message. Therefore, we recommend using it.

Name-of-sender is the name of the person or organization sending the e-mail. E-mail address is expressed in standard form: name@domain

Reply-To: ""name-of-sender" <e-mail address of sender>

This header is optional. However, most e-mail client software displays this field at the beginning of a message. Therefore, we recommend using it.

! Name-of-sender is the name of the person or organization
! sending the e-mail. E-mail address is expressed in standard
! form: name@domain
!
! To: "name-of-recipient" <e-mail address of recipient>
!
! This header is optional. However, most e-mail client software
! displays this field at the beginning of a message. Therefore, we
! recommend using it.
!
! Name-of-recipient is the name of the person or organization
! receiving the e-mail. E-mail address is expressed in standard
! form: name@domain
!
! Subject: subject-line
!
! This header is optional. However, most e-mail client software
! displays this field at the beginning of a message. Therefore, we
! recommend using it.
!
! Note: Most e-mail readers expect a blank line between the headers and
! the text portion of the message.
!
! n+1 When the text is complete, include a blank record in the
! e-mail interface file.
!
! n+2 Following the blank record, you must include a record with a
! single period (.) in the first position, which is an SMTP command
! indicating that the message is complete.
!
! Here's a summary of the records in a typical e-mail interface file:

```
!  
! CONNECT SMTP-server MAIL  
! HELO server-name  
! MAIL FROM: e-mail address of sender  
! RCPT TO: e-mail address of recipient  
! DATA  
! Date: Day, DD-Month-YYYY HH:MM:SS GMT  
! From: "name-of-sender" <e-mail address of sender>  
! Reply-To: "name-of-sender" <e-mail address of sender>  
! To: "name-of-recipient" <e-mail address of recipient>  
! Subject: subject-line  
! (blank line)  
! text of e-mail message  
! text of e-mail message  
! text of e-mail message  
! text of e-mail message  
! (blank line)  
! .  
!  
! SENDMAIL reads the above records from the text file and writes them  
! to the Winsock gateway (a "type 3" gateway that must be configured for this  
! program to work correctly).  
!  
! When SENDMAIL is done processing the e-mail interface file, it exits back to  
! the calling program (if running as a subprogram) or stops (if not running  
! as a subprogram).  
!  
! Progress log:  
!  
! Each SMTP command results in a specific "reply code" that indicates  
! whether the command was successful or not. These codes are defined in  
! RFC 821. SENDMAIL contains an option that displays a progress log of
```

```

! these reply codes.
!
! The option is defined with a variable named DISPLAY$.
!
! If you are using SENDMAIL as a subprogram, write the value of DISPLAY$
! as the third field in the I/O buffer before entering SENDMAIL.
!
! If you are not using SENDMAIL as a subprogram, the value of DISPLAY$ is
! set via a prompt (see the code section below).
!
! Here are the values of DISPLAY$:
!
!     YES    SENDMAIL creates a COSW window and displays the progress log
!             in it. At the end of the program, the window is deleted.
!
!     WEB    SENDMAIL displays the progress log on a web page, using
!             HTML. (This option assumes that SENDMAIL is running as a
!                   subprogram to an XAP program.)
!
! If DISPLAY$ contains another value, the progress log is not displayed.
!
! Buffer data:
!
! This program contains the following buffer variables:
!
! 1. E-mail interface file name (12 bytes; 8.3 format)
! 2. Directory name where e-mail interface file is located (3 bytes)
! 3. DISPLAY$ flag (3 bytes)
!
!=====
!   DATA SECTION
!=====

```

LENGTH	12	& LOCAL	FILENAME\$	! E-mail interface filename
LENGTH	3	& LOCAL	DIRECTORY\$	! Directory name
LENGTH	3	& LOCAL	DISPLAY\$	! Display flag
LENGTH	254	& LOCAL	CMD\$, RESULT\$	! Command & result strings
LENGTH	254	& LOCAL	DATA\$	! Text of message
LENGTH	3	& LOCAL	GateName\$	! Gateway name
LENGTH	3.0	& LOCAL	I	! Work variable
LENGTH	50	& LOCAL	Today\$	! Date/time

```

!=====
!   F O R M A T   S E C T I O N
!=====

```

```

BUFFER:  FORMAT  FILENAME$;DIRECTORY$;DISPLAY$      ! I/O buffer format

```

```

DATA:  FORMAT  DATA$                                ! File format

```

```

!=====
!   C O D E   S E C T I O N
!=====

```

```

CLEAR                                             ! Clear all variables

```

```

READ  (#,BUFFER)                                ! Read filename, directory,
                                                ! display flag from I/O buffer

```

```

IF ENTERLEVEL = 0 THEN                          ! Not running as a subprogram
    Print (CS)

```

```
Print "SENDMAIL 2.1 -- STMP Mail Sender",@(0,0)
```

```
Print "Enter e-mail interface file name:",@(0,2) ! Input file name
```

```
Input @(34,2),FILENAME$
```

```
FILENAME$ = UCASE(FILENAME$)
```

```
IF FILENAME$ = "" THEN
```

```
! If null, then stop
```

```
STOP
```

```
ENDIF
```

```
Print "Directory:",@(0,3)
```

```
! Input directory name
```

```
Input @(34,3),DIRECTORY$
```

```
DIRECTORY$ = UCASE(DIRECTORY$)
```

```
Print "Display progress log? (YES/NO) YES",@(0,5) ! Progress log?
```

```
Input @(34,5),DISPLAY$
```

```
DISPLAY$ = UCASE(DISPLAY$)
```

```
IF SUB(DISPLAY$,1,1) = "Y" THEN
```

```
DISPLAY$ = "YES"
```

```
ENDIF
```

```
ENDIF
```

```
! *** Process the display flag
```

```
!
```

```
! Note: YES = display progress log on screen
```

```
! WEB = display progress log on web browser (HTML format)
```

```
! (otherwise, don't display progress log)
```

```
SELECT CASE DISPLAY$
```

```
CASE "YES"
```

```
Print (Screen=100,50)
```

```
Print (Create Window=0,0,99,49)
```

```

        Print (Window Title Background="E-mail progress log",25)
        Print (ET);(CS)
CASE "WEB"
        Print "<HTML><HEAD><TITLE>E-Mail progress log</TITLE></HEAD><BODY>"
        Print "<B>E-mail progress log<BR>"
ENDSELECT

! *** Open the e-mail interface file

        OPEN (10) FILENAME$,DIR=DIRECTORY$,EXCP=UnexpectedFileNotFound

! *** Find the Winsock (Type 3) Gateway

        GOSUB Find.Gateway                                ! Find gateway

        IF GateName$ = "" THEN                             ! If null, quit the program
                GOTO Abort
        ENDIF

        SELECT CASE DISPLAY$                               ! Progress log
        CASE "YES"
                Print "Gateway = ";Gatename$
        CASE "WEB"
                Print "Gateway = ";GateName$;"</B><BR>"
        ENDSELECT

! *** Terminate all print records with a CR/LF (Gateway command format is: 'WEOL 1')

        CMD$ = "WEOL 1"                                    ! Set command string
        GOSUB CONTROL                                      ! Go to control subroutine
        IF SUB(RESULT$,1,1) NE "+" THEN                    ! Check result string
                GOTO QUIT

```

ENDIF

! *** Connect to SMTP server (Gateway command format is: 'CONNECT smtp.server.name MAIL')

```
READ (10,DATA),EXCP=UnexpectedEOF      ! Read SMTP server name
CMD$ = STRIP(DATA$)                     ! Set command string
GOSUB CONTROL                           ! Go to control subroutine
IF SUB(RESULT$,1,1) NE "+" THEN         ! Check result string
    GOTO QUIT
ENDIF
```

```
GOSUB RECV                              ! To to recv subroutine
IF SUB(RESULT$,1,3) NE "220" THEN       ! Check result string
    GOTO QUIT
ENDIF
```

! *** Start SMTP conversation (SMTP command format is: 'HELO domain.name')

```
READ (10,DATA),EXCP=UnexpectedEOF      ! Read HELO command
CMD$ = STRIP(DATA$)                     ! Set command string
GOSUB SEND                              ! Go to send subroutine
GOSUB RECV                              ! Go to recv subroutine
IF SUB(RESULT$,1,3) NE "250" THEN       ! Check result string
    GOTO ABORT
ENDIF
```

! *** Specify sender (SMTP command format is: 'MAIL FROM: e-mail address')

```
READ (10,DATA),EXCP=UnexpectedEOF      ! Read sender's e-mail addr
CMD$ = STRIP(DATA$)                     ! Set command string
GOSUB SEND                              ! Go to send subroutine
GOSUB RECV                              ! Go to recv subroutine
```

```

        IF SUB(RESULT$,1,3) NE "250" THEN          ! Check result string
            GOTO ABORT
        ENDIF

! *** Continue reading the e-mail interface file, looking for RCPT command(s) and
! DATA command
!
! SMTP command format is: 'RCPT TO: e-mail address'
! 'DATA'
!
! Note: The e-mail interface may contain multiple "RCPT TO:" commands, indicating
! that there are multiple recipients for the e-mail message.

DO
    READ (10,DATA),EXCP=UnexpectedEOF              ! Read the next record
    CMD$ = STRIP(DATA$)                             ! Set command string

    IF SUB(CMD$,1,4) = "RCPT" THEN                  ! If SMTP command is "RCPT"
        GOSUB SEND                                  ! Go to send subroutine
        GOSUB RECV                                  ! Go to rcv subroutine
        IF SUB(RESULT$,1,3) NE "250" THEN          ! Check result string
            GOTO ABORT
        ENDIF
        CONTINUE                                    ! Continue the loop
    ENDIF

    IF SUB(CMD$,1,4) = "DATA" THEN                  ! If SMTP command is "DATA"
        GOSUB SEND                                  ! Go to send subroutine
        GOSUB RECV                                  ! Go to rcv subroutine
        IF SUB(RESULT$,1,3) NE "354" THEN          ! Check result string
            GOTO ABORT
        ENDIF

```

```

        BREAK                                ! Break out of the loop
    ENDIF
LOOP

! *** Read and send the message (text, headers, and attachments)

DO                                            ! Start of loop
    READ (10,DATA),EXCP=UnexpectedEOF        ! Read next record
    CMD$ = STRIP(DATA$)                     ! Strip blanks
    GOSUB SEND                               ! Go to send subroutine

    IF CMD$ = "." THEN                       ! Check for . record
        GOSUB RECV                          ! Receive status for . record
        IF SUB(RESULT$,1,3) NE "250" THEN    ! Check result string
            GOTO ABORT
        ENDIF
        GOTO QUIT
    ENDIF

LOOP                                         ! Loop

!- - - - - EXCEPTION AND EXIT ROUTINES - - - - -

UnexpectedFileNotFound:                     ! Unexpected file not found
    SELECT CASE DISPLAY$                     ! Check display flag
    CASE "YES"                               ! If YES,
        Print "The e-mail interface file was not found."    ! display message.
        Print ""
        Print "Please verify that this file exists."
        Input ""
        Print (Delete Window)               ! Close window
        Print (DE)                          ! Set default screen mode

```



```

GOSUB SEND          ! Go to send subroutine
GOSUB RECV          ! Go to recv subroutine

```

```

CMD$ = "QUIT"      ! Set command string to QUIT
GOSUB SEND          ! Go to send subroutine
GOSUB RECV          ! Go to recv subroutine

```

QUIT:

```

SELECT CASE DISPLAY$ ! Check display flag
CASE "YES"           ! If YES,
    Print "Press <Enter> to exit from this program." ! display message,
    Wait              ! wait for operator input,
    Print (Delete Window) ! and delete window
    Print (DE)        ! Set default screen mode
CASE "WEB"
    Print "</BODY></HTML>" ! end HTML document
ENDSELECT

```

ExitRoutine:

```

Close (10)          ! Close e-mail interface file
Close (20)          ! Close Winsock gateway

IF ENTERLEVEL > 0 THEN ! If running as a subprogram,
    EXIT             ! Return to calling program
ELSE                 ! Otherwise
    STOP             ! Stop
ENDIF

```

!===== SUBROUTINES =====

Control:

```

SELECT CASE DISPLAY$ ! Check display flag

```

```

CASE "YES"                                ! If YES,
    Print (SB);CMD$                        ! Echo control to screen
CASE "WEB"                                ! If WEB,
    Print "<B>";CMD$;"</B><BR>"           ! Echo control in HTML
ENDSELECT

RESULT$ = CONTROL(20,CMD$)                ! Send control to gateway

SELECT CASE DISPLAY$                      ! Check display flag
CASE "YES"                                ! If YES,
    Print (SF);RESULT$;(SB)              ! Echo result to screen
CASE "WEB"                                ! If WEB,
    Print RESULT$;"<BR>"                 ! Echo result in HTML
ENDSELECT

RETURN

```

!- - - - -

Send:

```

Print (20) CMD$                            ! Send command to gateway

SELECT CASE DISPLAY$                      ! Check display flag
CASE "YES"                                ! If YES,
    Print (SB);CMD$                        ! Echo command to screen
CASE "WEB"                                ! If WEB,
    Print "<B>";CMD$;"</B><BR>"           ! Echo command in HTML
ENDSELECT

RETURN

```

!- - - - -

Recv:

```
Input (20) RESULT$           ! Get the result string

SELECT CASE DISPLAY$         ! Check display flag
CASE "YES"                   ! If YES,
    Print (SF);RESULT$;(SB)  ! Echo result to screen
CASE "WEB"                   ! If WEB,
    Print RESULT$;"<BR>"    ! Echo result in HTML
ENDSELECT

RETURN
```

!- - - - -

```
! *** This subroutine searches all of the gateways (G00 through G99) and looks
!     for one that is configured as a Type 3 gateway. If found, the gateway
!     device name is returned in the GateName$ variable. If no Type 3 gateway
!     is configured, GateName$ is set to null.
```

Find.Gateway:

```
FOR I = 100 TO 199
    GateName$ = "G" + SUB(STRIP(STR(I)),2,2)
    IF SUB(DSTAT(GateName$,EXCP=Next.Gateway),4,1) EQ "@03@" THEN
        GOTO Gateway.Found
    ENDIF
```

Next.Gateway:

```
NEXT I

GateName$ = ""
RETURN
```

```
Gateway.Found:  
    CLOSE (20)  
    OPEN (20) GateName$, EXCP=Next.Gateway  
    RETURN
```

```
END
```