

Package 20001 - Email Tools
Email Utility (*web/mail)
Details

Calling Sequence:

CALL "*WEB/EMAIL", FROMADDRESS\$, REPLYADDRESS\$, TOADDRESS\$, CCADDRESS\$,
 BCCADDRESS\$, SUBJECT\$, MESSAGE\$, ATTACHMENTS\$, OPTION\$, SMTPSERVER\$,
 SERVERTIMEOUT, LINEWRAPSAT, BODYENCODING\$, ERASEIT, ERRORMESG\$,
 SENDDIRECTORY\$, SAVEDFILENAME\$, err=*next

Where the parameters are:

FROMADDRESS\$	(Input)Address to be used in the mail header to indicate who is sending the mail. (only 1 address allowed).
REPLYADDRESS\$	(Input)Address to be used when the recipient of the message wishes to force any replies to be routed to a different address.
TOADDRESS\$	(Input)Address(es) to send the mail to. (1 or more)
CCADDRESS\$	(Input)Address(es) to send a copy of the mail to. (0 or more)
BCCADDRESS\$	(Input)Address(es) to send a copy of the mail to but whose address is not to be exposed to other recipients. (0 or more)
SUBJECT\$	(Input)The subject line for the message
MESSAGE\$	(Input)The body of the message to be sent. (See below for details on its content)
ATTACHMENTS\$	(Input)The list of files to attach to the email.(Null for no file attachments)
OPTION\$	Input)This may be set to "H"igh, "L"ow, or "N"ormal and is used to set the email X-Priority level. Also used to indicate Read Receipt requested (For priority: H, L, or N; Read receipt: D and/or R)
SMTPSERVER\$	(Input)SMTPSERVER\$ contains the name of the SMTP server. It should be in the form of SERVER;SOCKET such as smtp.pvx.com;25. If no socket number is specified then a default SMTP socket number of 25 is used. If no server name is specified (smtpserver\$ is NULL) or contains the word "<<OUTBOX>>", then the email will be generated and left in the *WEB/OUTBOX directory. If the SMTP server requires authorization , this field can contain the user and password. See 'Authorization' below.
SERVERTIMEOUT	(Input)If non-zero this parameter specifies the timeouts for sending direct to the SMTPServer. If zero the timeout value will be set to 60 seconds. (Values greater than zero only have an affect of Pvx version 4.15 or higher)
LINEWRAPSAT	(Input)The value in this parameter controls the automated line wrapping of text in the message body. If this parameter is set to -1 then no line wrapping will be done. If 0, then a default of 76 chars/line will be used. Wrapping is only performed on the message body, and does not affect attachments.
BODYENCODING\$	(Input)This is the mime type for the MESSAGE\$, we will assume text/plain if not given. By specifying a value in BodyEncoding\$, such as "text/html", the message\$ would be assumed to be HTML.
ERASEIT	(Input)The value specified in this parameter indicates whether the program is to delete or save the generated .EML file that is output by this program. It is used ONLY if the message has been successfully sent. Possible values are: 0 = Leave the message file alone <default>, 1 = Delete the sent message file, 2 = Save the sent message to a Sent Folder
ERRORMESG\$	(Output)If an error is encountered, during the compile of an email, or in the course of sending an email, the program will load this variable with a textual description of the problem and then exit with an error.

SENTDIRECTORY\$	(Input) Indicates the directory name to use to save Sent Items. This is valid when eraseit=2 only.
SAVEDFILENAME\$	(Input) Returns the name on disk of the file where the email file was stored after completion. If saving to the outbox, this returns the location and filename within the *web/outbox. If eraseit is not set or is set to 0, then this returns the location and filename where the generated email was left. If eraseit is set to 1, then the file is not saved and this will return a null value. If eraseit is set to 2, then this will indicate the location of the saved file. This defaults to the *web/outbox/sent or SentDirectory\$, if given.

Purpose of Program:

This utility may be called to generate a mime-compliant e-mail message. The file generated will have a long but unique filename assigned by this utility and will have a file extension of .EML - this utility will only work if run from the ProvideX *web directory.

Up to 3 subdirectories will be created:

*web/outbox = Generated E-Mail may be left here for later delivery
 *web/outbox/new = All email is generated as a file in this directory, until delivery status can be determined at which time it will be moved or deleted
 *web/outbox/sent = E-mail may be kept after being sent if parameters dictate they be saved.

Usage Details:

E-Mail Addresses:

Individual e-mail addresses may be specified in simply or complex forms, such as:

support@pvx.com
 "Pvx Support" <support@pvx.com >

This makes use of the *web/address utility to validate, verify and correctly format or reformat e-mail addresses.

Where multiple e-mail addresses are supported (ToAddress\$, CCAddress\$, BCCAddress\$), you may string the email addresses together separated by any of the following:

\$8A\$ (SEP), \$0A\$, \$0D0A\$ or ";" (Semi-Colon).

Note: This utility accepts up to 100 email addresses for each of the above items, however most SMTP servers and most mail delivery spam filters will reject mail that is addressed to more than 100 users.

Message Body:

The MESSAGE\$ variable may contain up to a maximum of 32000 characters. However, you may embed coding to represent the insertion of files into the message body. By using a body built of files or a combination of text and files, the maximum body size itself is unlimited.

Embedding files within the message body, is accomplished by inserting strings of the following format within the MESSAGE\$:

""<<#FILE:filename>>"" or ""<<#FILE:filename?option>>""

The filename should contain both the path and the filename of the file. If the file has a file extension then the mime database will be checked for an entry matching the file's extension and

the mime type will be retrieved. If no mime type is found, then "application/octet-stream" is assumed.

Files will automatically be encoded in 1 of 3 ways when being inserted into the final .EML file:

1. If the mime type is "text/plain" no encoding will be done, however the linewrap will be performed, unless LINEWRAPSAT is set to -1.
2. All of the other mime types beginning with "text/" such as "text/html" will be encoded in Quoted-Printable format.
3. All mime types that are not "text/" will be encoded in Base64.

The second format with the ?option suffix allows you to override the actual encoding used or performed on the file.

Valid Options are:

- ?Q Force file to be encoded in Quoted-Printable
- ?B Force file to be encoded in Base64
- ?E Do not encode, file is assumed to already exist in Quoted-Printable format
- ?6 Do not encode, file is assumed to already exist in Base64 format
- ?P Declares the file as text/plain, encoding will depend on usage. If the file is an attachment then it will be encoded in Quoted-Printable format. If the file is part of the message body then no encoding will occur and no line wrap will be done, unless the entire message body turns out to be text/plain, in this case line wrapping will occur based on LINEWRAPSAT.
- ?I Include file As-Is. Do not encode the file, do not put mime headers before the file's content. The utility will simply wrap mime boundaries around the inserted file. By using this feature, you can manually encode mime based files of any format or layout, such as multipart/related and include that into this email file. The mime type of the data within the MESSAGE\$ variable is assumed to be plain text unless overridden by the BODYENCODING\$ variable which may contain a mime type for the message body.

Using this technique it is very simple to create a message from an HTML page:

```
MESSAGE$="<html><head><title>..</html> "  
BODYENCODING$="text/html"
```

or simply:

```
MESSAGE$="<<#FILE:/somefile.html>>"
```

Since this message body is built from a file, the mime type will be worked out automatically and therefore BODYENCODING\$ need not be set. However, you can also combine bodies encoded one way, with files encoded another:

```
MESSAGE$="<html><head><title></title>..</head><<#FILE:/somewhere.txt>>  
</body></html>"  
BODYENCODING$="text/html"
```

Whenever there is a file inside the message body, the file will always start on a new line, it will not continue an existing line. If the message body is built of text and files, and both the text and all of the files turn out to be of mime type "text/plain" then the message will be built as a single body, and not as multiple pieces. Other than this rule, the message body and the entire email will be built as a "multipart/mixed" mime with boundaries between each component.

Attachments:

ATTACHMENTS\$ may contain a separated list of file names - the files to be attached to the email message. The separator may be:

\$8A\$ (SEP), \$0A\$, \$0D0A\$ or ";" (Semi-Colon).

No special formatting is used or needed, however the ?options are available just as in the message body, and have the same affects in the message body e.g.:

ATTACHMENTS\$="c:\file1.htm;c:\image.bmp;c:\somefile.mine?B;C:\someotherfile.ok?Q"

If the file has a ?option suffix, then the encoding of the file will be forced, otherwise the file's extension will be used to find the mime type in the mime type database. Encoding will be Quoted - Printable for any mime types starting with "text/" all others are encoded as Base64.

Note: Even though you may force the file encoding, the mime type and mime type database is still very important to have for each file extension you use. If the mime type cannot be found then application/octet-stream will be used. Incorrect or missing mime types may cause unknown results or undisplayable mail in the receivers mail client.

Mime Type Database:

The mime type database is a file named "*web/webserv.mim" this is a keyed file, which us es the file extension as the key. At present the only available method for modifying, adding or deleting mime types is through the use of the ProvideX Web Server Configuration Screen. (Note, it is not necessary to purchase the ProvideX Web Server to access its configuration system.) Alternatively you may modify the mime type file directly, the file should be opened with IOL=* and all filename extensions must be written to the file in lower case.

Mail Priority/Options:

The following table describes the actual internal email priority settings that will be incorporated into the email header based on the value you supply in the OPTION\$ value:

Value	X-Priority	X-MSMail-Priority	Importance
L	5	Low	Low
N	3	Normal	Normal
H	1	High	High
D	Return a Read Receipt, also known as a Disposition header		
R	Return a Delivery Receipt, also known as a Return-Receipt.		

Not all email tools will respond to priority settings, or they may not respond to the specific email header tags of "X-Priority:" or "X-MSMail-Priority:" or "Importance:"

Character Set:

For all text-based messages, or files, the character set will be reported as "us -ascii" unless otherwise overridden by the global variable: %EMAIL_CharSet\$.

Sending or Storing of Generated E-Mail Messages:

Email may be stored in an outbox (*web/outbox) by specifying NULL or "< >" for the SMTPServer\$ argument. At no time will this program read mess ages from the outbox and attempt to deliver them. If an SMTPServer\$ is specified, then the message will attempt to post itself immediately after it has been generated.

Errors and Results:

If an email cannot be completely generated, then it will not be s ent to any recipients. The reason will be placed in ErrorMsg\$. If during the course of generating an email, one of the addresses in any of the address fields is found to be invalid, then the email will not be sent to any recipients. A list of bad email addresses will be returned in ErrorMsg\$. If during the course of posting the email one of the recipients is found to be invalid, then the email will be posted to all valid

SMTP Authorization:

If the SMTP server you are connecting to requires authorization, you can include the userid and password in the SMTPServer\$ value following the server name and port number.

For example given the following:

Server name:	mail.abccorp.com
Smtplib port number:	25
User name:	foo.manchu
Password:	chungking

You would set your SMTPServer\$ variable to:

```
mail.abccorp.com;25;foo.manchu;chungking
```