# TCP/IP SMTP User Exits Support APAR PQ04382

# VM TCP/IP Version 2 Release 4

February, 1999

VM TCP/IP Development

VM/ESA and Related Products
Endicott, New York

# Contents

# Background

As the use of e-mail has increased among the internet community, incidents related to the misuse of e-mail and the abuse of e-mail systems have also risen. Likewise, discussions about how to address and resolve problems associated with such misuse have increased. Some more prominent e-mail abuse problems include spamming, spoofing, and unwanted mail forwarding. In all these cases, the resources of your company or institution may be used by others without your permission. The SMTP exits described in this document are designed to address some of these problems by allowing you greater control over each piece of mail that's processed by the SMTP server.

To understand why you might choose to use one or more of these exits, and determine which ones will best address the particular problems at your installation, explanations for some common e-mail terms are included here:

Spamming
: Spamming is the act of sending mail to a large number of e-mail addressees and is often compared to the term “junk mail” used to describe similar activities performed via postal services. *Spam* is a piece of mail that is perceived by its recipient(s) to be unsolicited and unwanted.

Spoofing
: Spoofing is an act performed by a client sending mail. The client connects to a mail transfer agent (the SMTP server is one example) and falsifies the information it's required to provide. Spoofing is usually done in order to cause mail to appear to have come from somewhere else.

Mail Forwarding
: Mail forwarding is the process of receiving and sending on mail that is directed to your mail server but is ultimately destined to somewhere other than your own site; this is a normal and accepted practice in many cases. Mail forwarding becomes a problem when the originating site “bounces” its mail off an intermediate site so that the destination site does not reject that mail, because mail from the originating site is not desired.

Source Routing
: Source Routing is a particular type of mail forwarding. In this case, the specified mail recipient path includes a mailbox and a list of hosts. The list of hosts is the *route*, or the information about how to get to the final destination. Mail is passed from one host in this list to the next, until it is delivered to the intended recipient.

  For example, suppose a host receives a message with the SMTP command:

  ```
  RCPT TO:<@ALPHA,@BETA:joe@GAMMA>
  ```

  where `ALPHA` and `BETA` are the list of hosts, and `joe@GAMMA` is the mailbox. The mail will be sent to `ALPHA`, then to `BETA`, and finally to `joe@GAMMA`.

  In the past, this technique was often necessary to accomplish mail delivery, and some institutions may still need to use source routing because they have not migrated to using more current practices. However, source routing is now *strongly* discouraged by many “experts” among the internet community, especially as mail processing technologies have improved over time.

## The SMTP Envelope

To better understand the parameters associated with the SMTP exits and how the exits work, it's necessary to understand SMTP transactions.

Mail is sent by a series of request/response transactions between a client, the “sender-SMTP,” and a server, the “receiver-SMTP.” These transactions pass (1) the message proper, which is composed of a header and body (these are separated, by definition, by the first blank line present in the message), and

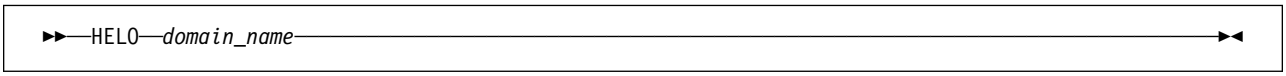
(2) SMTP commands, referred to as the “envelope.” These commands contain additional information, including the host sending the mail, and the source and destination addresses for the piece of mail.

The SMTP envelope is constructed at the “sender-SMTP” site. If this is the originating site, the information is typically provided by a mail user agent (such as the VM NOTE command) when the message is first queued to the “sender-SMTP” client or program. Each intermediate site receives the piece of mail and resends it on to the next site using an envelope that it creates. Thus, the content of the new envelope may not be the same as the one received from a given site.

The envelope addresses may be derived from information in the message header, supplied by the user interface or derived from local configuration information.

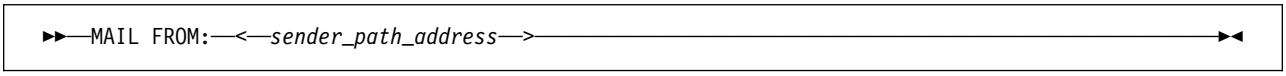
The envelope contains the HELO, MAIL FROM:, RCPT TO:, DATA, “dot” (ASCII period) and QUIT commands. Other commands can optionally appear in the envelope, and some commands can be repeated. Also, more than one piece of mail can be sent using one envelope.

Descriptions of the HELO, MAIL FROM:, RCPT TO:, DATA, “dot” and QUIT commands follow:

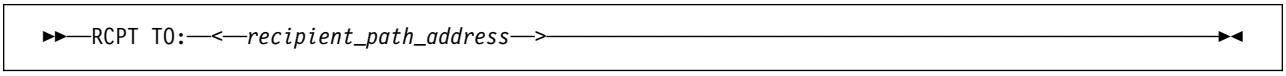



The *domain_name* identifies the domain name of the sending host; it may be specified as either:

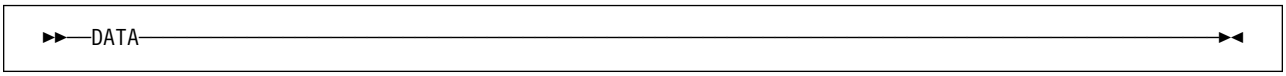
- a domain name
- an IP address in decimal integer form that is prefixed by the number or (US) pound sign (“#” or X'7B'0)
- an IP address in dotted-decimal form, enclosed in brackets.



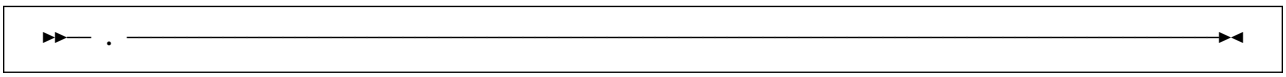

The *sender_path_address* is the full path address of the sender of the mail; it can be specified using any valid path format.



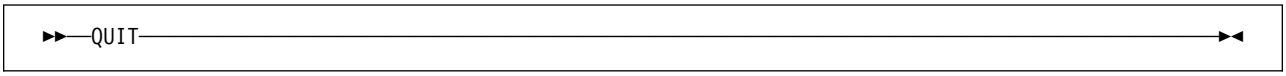

The *recipient_path_address* is the full path address of the sender of the mail; it can be specified using any valid path format. Any number of RCPT TO: commands can be specified to designate multiple recipients.




The DATA command has no parameters. Information that follows the DATA command is construed to be the message text (the header and body of a message).




The “dot” command is a single ASCII period (.) on a line by itself. It is used to terminate or signify the end of the message text. There are no parameters for this command.




The QUIT command has no parameters; it is used to terminate an SMTP connection.

Definitions for “valid path format” specifications can be obtained from the RFCs that define the naming conventions used throughout the Internet. Because both these specifications and even the applicable RFCs are subject to change, they are not be described here. For detailed information about this topic, its suggested you begin with RFC 821, *Simple Mail Transfer Protocol*, and RFC 822, *Standard for the Format of ARPA Internet Text Messages*, which are the basis for modern naming specifications associated with the SMTP protocol.

The following is an example of an SMTP envelope and the contained piece of sent mail. The SMTP commands are in capitalized, bold-faced text. The information after the DATA command and before the single ASCII period (the “dot” command) is the message proper — the header and body. The body is separated from the header by the blank line that follows the “Subject: Update on the Bunch” line.

```
HELO yourname
MAIL FROM: <carol@yourname>
RCPT TO: <msgs@rsch.our.edu>
RCPT TO: <alice@ai.our.edu>
DATA
Date: Thur, 27 Aug 92 21:48:57 EST
From: Carol <carol@yourname>
To:   <msgs@rsch.our.edu>
Cc:   <alice@ai.our.edu>
Subject: Update on the Bunch

Mike -- Cindy stubbed her toe.  Bobby went to baseball camp.  Marsha
        made the cheerleading team.  Jan got glasses.  Peter has an
        identity crisis.  Greg made dates with 3 girls and couldn't
        remember their names.  Bye!
.
QUIT
```

## General Exit Information

The SMTP user exits are comprised of: the client verification exit, the mail forwarding exit and the SMTP commands exit. Sample assembler and REXX exec exit routines are supplied for each exit. An identical exit parameter list definition is used for all three of these exits; thus, all parameters may not be meaningful for all exits. EBCDIC data is supplied for all exit “text” fields.

Assembler exits are loaded into memory, while the REXX exits are EXECLOADed. When deciding which type of exit to use, it's important to consider the performance aspects of the exit, because the exit will “block” the SMTP server. The assembler exit will have better performance characteristics than the REXX exit.

## Client Verification Exit

The client verification exit can be used to verify the host providing mail is as claimed on the HELO command, and to include the result of that verification in mail headers. The exit can be tailored to perform the verification or it can be used enable or disable the built-in client verification function.

This exit can be used to help deal with *spoofing* problems. When a client connects to SMTP, the originating mail domain must be provided. In spoofing, the client provides a falsified domain in order to cause mail to appear to have come from someone else. By using this exit, you can verify the client's domain matches the client's IP address. The exit allows flexibility on the action you take for spoofing; you can choose to include the verification results in mail headers or you can reject future communications on this connection. The exit can be further tailored to perform additional actions that are unique or required for your environment.

In addition to the client verification exit, YES and NO parameters exist that can be used to turn on (or off) client verification. If the YES parameter is used, the built-in client verification function is called for each HELO processed for each mail item.

## Mail Forwarding Exit

When SMTP clients use the VM SMTP server to send mail to hosts their workstations cannot reach directly, this is an instance of *mail forwarding*. The mail forwarding exit provides a mechanism to control mail forwarding. When SMTP determines the addressee specified on a RCPT TO: record is not *defined on* the local system, it has detected mail forwarding, and it will call the exit routine.

The phrase "*defined on*" in the previous paragraph is used because a user is defined as being a local user (in addition to any other criteria) if that user is defined in the SMTP NAMES file, regardless of whether delivery to that user is via the network or via spool. Also, keep in mind that the determination of whether mail forwarding is occurring is on a recipient by recipient basis, not on the basis of a piece of mail. A piece of mail with multiple recipients can contain occurrences of both mail forwarding and local delivery.

The mail forwarding exit allows flexibility on the action you take when mail forwarding is detected. You can choose to continue to deliver the mail, reject delivery to the specified user, intercept the mail, or reject future communications on this connection. The exit can be further tailored to perform additional actions that are unique or required for your environment.

This exit can also be used to control *spamming*. There are two aspects to consider when trying to control spamming problems:

- Is your system being used to relay spam messages to recipients throughout the internet?
- Are incoming spam messages to your local users seriously taxing or overloading your system?

The relaying of spam messages can be treated like any other type of mail forwarding. The exit can be set up to prevent delivery of all forwarded mail, to prevent delivery of mail from particular sites known for spamming, or to only allow delivery of mail from particular trusted sites. Handling spam messages directed to your local users will require the use of the SMTP commands exit. When you address spamming problems, it's important to realize that one person may consider a piece of mail to be a spam, while the same piece of mail may be valuable to someone else. There are no explicit rules that determine what is and is not spam.

In addition to the mail forwarding exit, YES and NO parameters exist that can be used to turn on (or off) **all** mail forwarding. If mail forwarding is disabled via the NO parameter and SMTP determines the recipient specified on a RCPT TO: record is not defined on the local system, it has detected mail forwarding, and it will reject the delivery of the mail to that recipient.

## SMTP Commands Exit

In addition to the more specific client verification and mail forwarding exit functions, the SMTP server can be set up to call an exit routine whenever certain SMTP commands are received by the server. This capability is provided through the SMTP commands exit; the definition of this exit allows it to be called for any or all of the following commands:

- HELO (the SMTP 'HELO' command)
- MAIL (the SMTP 'MAIL FROM:' command)
- RCPT (the SMTP 'RCPT TO:' command)
- DATA (the SMTP 'DATA' command)
- EOD (the end of data condition, which occurs when a period is received at the server following the data)
- VRFY (the SMTP 'VRFY' command)
- EXPN (the SMTP 'EXPN' command)
- RSET (the SMTP 'RSET' command)
- PUNCH (the point in time when the server is about to deliver mail to a local destination on the same node or RSCS network).

The SMTP commands exit could be used for a wide variety of reasons; several possible uses are included here:

- to disable a particular command by always causing it to be rejected. For example, the exit can be set up to reject the VRFY and EXPN commands.
- to allow a user exit to handle the delivery of local mail. For example, the exit can be set up to deliver local mail to users' OfficeVision “In-Basket”s instead of to their CMS readers.
- to screen the body of the mail for a particular word or phrase and reject the mail. For example, the exit can be set up to reject mail with offensive language.
- to help control incoming spam messages directed to your local users, or to handle spam messages that may be seriously taxing or overloading your system. For example, the exit can be set up to prevent delivery of mail from particular sites known for spamming, or to only allow delivery of mail from particular trusted sites.

## SourceRoutes

Support for a new SMTP configuration file statement, SOURCEROUTES, has been added to address a particular type of mail forwarding called *source routing*. Source routing is the practice of providing a list of specific hosts that mail is to be passed to until it reaches its intended recipient. Before the use of name server MX (mail exchange) records became widespread, it was often necessary to provide a list of such hosts in order to deliver one's mail. Today, the need to specify these routing hops during mail delivery is minimal, and this practice is strongly discouraged. Some internet hosts, however, make use of source routing in order to circumvent processing that other hosts may be using to limit their activities.

This new statement gives you the ability to disable source routing. If source routing is disabled and a source route is encountered, the list of hosts will be ignored, and the mail will be delivered directly to the destination mailbox.

For example, suppose a host receives a message with the SMTP command:

```
RCPT TO:<@ALPHA,@BETA:joe@GAMMA>
```

where `ALPHA` and `BETA` are the list of hosts, and `joe@GAMMA` is the mailbox. When source routing is disabled, the mail will be sent directly to host `GAMMA`, using:

```
RCPT TO:<joe@GAMMA>
```

# Client Verification Exit

It is sometimes desirable to provide some indication that the host sending or forwarding a piece of mail corresponds to the IP address assigned to the client host. The client verification exit can be used to determine if a client host name and IP address match, and to include the result of that determination in the mail headers. The VERIFYCLIENT statement in the SMTP configuration file allows you to specify whether an exit will be called, or if verification will be done via the built-in client verification function.

When used, the client verification exit is called for each HELO command processed for each mail item received from the network; client verification is not performed for mail items received from the SMTP reader.

The changes associated with this exit add the following configuration statements, commands, and samples:

- VERIFYCLIENT statement for the SMTP configuration file (SMTP CONFIG)
- REXX and assembler sample exits.

This exit can be used to perform the following function, plus others you may deem useful in your environment:

- Reject mail from a particular host.
- Mark certain trusted sites as verified, but perform verification on all others.
- Control which users use a particular SMTP server.

## Built-in Client Verification Function

The built-in client verification function of the SMTP server can be used to determine if a client host name and client IP address match, and to include the result of that determination in the mail headers. This function will perform a DNS lookup against client-provided HELO command data, and will then insert a message into the mail header that reflects the result of this lookup.

Client verification performed using the built-in function has three possible outcomes:

**Success** The data the client provided in the HELO command corresponds to the client address. The following line is inserted into the mail header:

```
X-Comment: localhost: Mail was sent by host
```

**Failure** The data the client provided in the HELO command is not associated with the client IP address. In this case, a reverse name lookup is done against the client IP address to determine the actual host name. The following line is inserted into the mail header:

```
X-Comment: localhost: Host host claimed to be helodata
```

**Unknown** The validation could not be performed. This situation could occur if the name server is not responding, or the verification could not be performed in the allotted time (as controlled by the VERIFYCLIENTDELAY statement). The following line is inserted into the mail header:

```
X-Warning: localhost: Could not confirm that host [ipaddr] is helodata
```

The terms used in the previously listed mail header messages are described in more detail here:

*localhost* the local VM host name

*helodata* the data the client provided with the HELO command

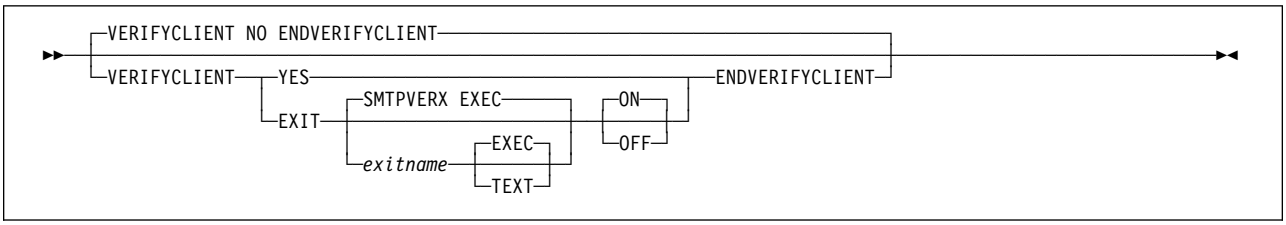
*host* the host name determined by the reverse lookup; if a host name is not found, "`unknown host`" will be used

*ipaddr* the client IP address.

# VERIFYCLIENT Statement

## Purpose

The VERIFYCLIENT statement is used to indicate whether or not client verification is to be performed. Client verification can be performed using the built-in client verification function (VERIFYCLIENT YES), or using a user exit (VERIFYCLIENT EXIT).




## Operands

**NO**
Indicates that no client verification is to be performed; this is the default.

**YES**
Indicates verification of the client name specified on the HELO command is to be performed using the built-in client verification function.

**EXIT**
Indicates a client verification exit routine is being defined.

*exitname*
Indicates the name of the exit routine associated with this command; the default exit name is **SMTPVERX**.

**EXEC**
Indicates the exit routine name specified on this command is the name of an EXEC; this is the default.

**TEXT**
Indicates the exit routine name specified on this command is the name of a text deck.

**ON**
Indicates the specified exit (being defined with this command) is to be enabled (turned on).

**OFF**
Indicates the specified exit (being defined with this command) is to be disabled (turned off).

## Examples

- The SMTP configuration file entry that follows will enable the client verification exit routine SMTPVERX EXEC; this exit will perform all verification.

```
VerifyClient
   exit smtpverx exec on
EndVerifyClient
```

  When this entry is processed, the following text is displayed during server initialization:

```
Client Verification            : Exit SMTPVERX EXEC ON
```

- This next entry defines the client verification exit routine SMTPVERX TEXT, but disables its use once SMTP server initialization is complete. Thus, client verification will not occur.

```
VerifyClient
   exit smtpverx text off
EndVerifyClient
```

  When this entry is processed, the following text is displayed during server initialization:

```
Client Verification            : No (Exit SMTPVERX TEXT OFF)
```

## Client Verification Exit Parameter Lists

Sample assembler and REXX exec exit routines are supplied for the client verification exit. The assembler exit will have better performance characteristics than the REXX exit.

The parameter lists passed to the REXX and the assembler exit routines follow. When you tailor either of these exits, keep in mind the following:

- Because an identical exit parameter list definition is used for all three exits — the client verification exit, the mail forwarding exit and the SMTP commands exit, all parameters may not be meaningful for this exit. Those not used by this exit are indicated in the parameter lists; their values should be ignored.
- For the REXX exit, the value of an unused parameter will be such that any token-wise parsing will not be affected.

Parameter descriptions that pertain to both the REXX and assembler exits are provided on page 11.

# REXX Parameter List

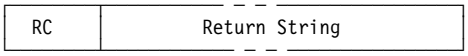
## Inputs

*Table 1. Client Verification REXX Exit Parameter List*

| Argument | Description |
|---|---|
| ARG(1) | Parameter list defined as follows:<br>• Exit type<br>• Version number<br>• Reserved field<br>• Port number of SMTP server<br>• IP address of SMTP server<br>• Port number of client<br>• IP address of client<br>• '*'<br>• Verify Client status<br>• Maximum length of Return String |
| ARG(2) | SMTP command string |
| ARG(3) | HELO name |
| ARG(4) | Not used |
| ARG(5) | Not used |
| ARG(6) | Not used |

## Outputs

The following are returned to the caller in the RESULT variable via a REXX RETURN statement:




| Argument | Description |
|---|---|
| RC | The exit return code; this must be a 4 byte numeric value. |
| Return String | An exit-specified string; the returned value must have a length less than or equal to the maximum length passed to the exit. |

## ASSEMBLER Parameter List

**Note:** General Register 1 points to the parameter list.

*Table 2. Client Verificaiton ASSEMBLER Exit Parameter List*

| Offset in Decimal | Len | In/Out | Type | Description |
|---|---|---|---|---|
| +0 | 4 | Input | Char | Exit type |
| +4 | 4 | Input | Int | Version number |
| +8 | 4 | Input | Int | Reserved field |
| +12 | 4 | Input | Int | Port number of SMTP server |
| +16 | 4 | Input | Int | IP address of SMTP server |
| +20 | 4 | Input | Int | Port number of client |
| +24 | 4 | Input | Int | IP address of client |
| +28 | 4 | Input | Int | Address of SMTP command string |
| +32 | 4 | Input | Int | Length of SMTP command string |
| +36 | 4 | Input | Int | Address of HELO name |
| +40 | 4 | Input | Int | Length of HELO name |
| +44 | 24 | | | Not used |
| +68 | 4 | Input | Int | Verify Client status |
| +72 | 4 | Output | Int | Address of Return String |
| +76 | 4 | Output | Int | Length of Return String |
| +80 | 4 | Input | Int | Maximum length of Return String |
| +84 | 8 | | | Not used |
| +92 | 4 | Input/ Output | Char | User Word 1 |
| +96 | 4 | Input/ Output | Char | User Word 2 |
| +100 | 4 | Output | Int | Return code from exit |

## Parameter Descriptions

**Exit type**

A four-character field used to indicate the type of exit called. For the client verification exit, this will be **VERX**.

**Version number**

The parameter list version number; if the parameter list format is changed, the version number will change. Your exit should verify it has received the version number it is expecting. The current version number is **1**.

**Port number of SMTP server**

The port number used by the SMTP server for this connection.

**IP address of SMTP server**

For the REXX exit, a dotted-decimal format IP address is provided; for the assembler exit, this is an IP address in decimal integer form. For multi-homed hosts, this address can be compared with the client IP address to determine in which part of the network the client host resides.

**Port number of client**

If the connection no longer exists, **-1** is supplied. Otherwise, this is the port number used by the foreign host for this connection.

**IP address of client**

For the REXX exit, a dotted-decimal format IP address is provided; for the assembler exit, this is an IP address in decimal integer form.

**Verify Client status**
A number that indicates client verification results. For this exit, client verification results are unknown when the exit receives control; thus, this field will contain a **3**. Possible values and their meaning are:

0 Client verification passed.
1 Client verification failed.
2 Client verification was not performed.
3 Client verification results are unknown.

**SMTP command string**
Contains the HELO command and the domain specified on the HELO command. The string has been converted to uppercase (for example, “HELO DOMAIN1”).

**HELO name**
A string that contains the name specified on the HELO command; this string may be either:

- a domain name
- an IP address in decimal integer form that is prefixed by the number or (US) pound sign (“#” or X'7B'0)
- an IP address in dotted-decimal form, enclosed in brackets.

For example, if the command `HELO #123456` is provided by an SMTP client, this parameter would contain `#123456`.

The name has already been verified to have the correct syntax.

**User Word 1**
Provided for use by the assembler exit only. The user word specified upon return from this exit will be passed back in this field for any future calls; **0** is the initial value. The SMTP server does not use this value in any way.

**User Word 2**
Provided for use by the assembler exit only. The user word specified upon return from this exit will be passed back in this field for any future calls; **0** is the initial value. The SMTP server does not use this value in any way.

**Return String**
When the exit returns a return code of 3, this value is appended to the `'X-Comment'` that is inserted in the mail header. When the exit returns a return code of 5, the *Return String* value is appended to the 550 reply code.

**Maximum length of Return String**
The current maximum is 512 bytes; ensure the *Return String* length is less than this value. If the returned string is longer than the indicated maximum, the return string is truncated and the following message is displayed on the SMTP sever console:

```
Return data from exit exitname exittype too long, data truncated
```

Normal processing continues.

## Client Verification Exit Return Codes

*Table 3. Client Verification Exit Return Codes*

| Return Code | Explanation |
|---|---|
| 0 | Do not verify client. A comment will not be inserted in the mail header. |
| 1 | Perform verification using the built-in client verification function. |
| 2 | Mark as verified. The following comment will be inserted in the mail header:<br>`X-Comment: localhost: Mail was sent by host` |
| 3 | The following comment will be inserted in the mail header:<br>`X-Comment: Return String`<br>where the value for *Return String* can be specified by the exit. |
| 4 | Disable the exit. The following message will be displayed on the SMTP console:<br>`VERIFYCLIENT EXIT function disabled`<br>The exit will no longer be called. The client will not be verified and no comment will be inserted in the mail header. |
| 5 | Reject this command with: `550 Return String`<br>If a return string is not provided by the exit, then the default message will be displayed:<br>`550 Access denied`<br>All future communications on this connection will be rejected with this 550 message. |
| *x* | Any return code other than the above causes SMTP to issue this message:<br>`Unexpected return from user exit exitname exittype, RC = rc`<br>SMTP treats this return code as if it were a return code of 0. |

## Client Verification Sample Exits

Sample exit routines are supplied with the APAR; the supplied samples are:

**SMTPVERX SEXEC** REXX exit routine

**SMTPVERX ASSEMBLE** assembler exit routine

# Mail Forwarding Exit

When SMTP clients use the VM SMTP server to send mail to hosts their workstations cannot reach directly, this is an instance of *mail forwarding*. The mail forwarding exit provides a mechanism to control mail forwarding. When SMTP determines the addressee specified on a RCPT TO: command is not *defined on* the local system, it has detected mail forwarding, and it will call this exit routine.

The phrase "*defined on*" in the previous paragraph is used because a user is defined as being a local user (in addition to any other criteria) if that user is defined in the SMTP NAMES file, regardless of whether delivery to that user is via SPOOL or the network. Also, keep in mind that the determination of whether mail forwarding is occurring is on a recipient by recipient basis, not on the basis of a piece of mail. A piece of mail with multiple recipients can contain occurrences of both mail forwarding and local delivery.

**Note:** The mail forwarding exit is only called for mail items received from the network; it is not called for mail items generated on the VM system or received via RSCS.

The changes associated with this exit add the following configuration statements, commands, and samples:

- FORWARDMAIL statement for the SMTP configuration file (SMTP CONFIG)
- REXX and assembler sample exits.

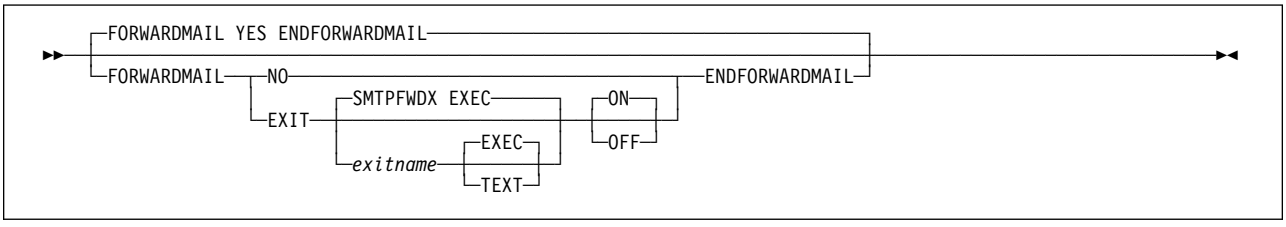
This exit can be used to perform the following function, plus others you may deem useful in your environment:

- Disallow mail forwarding from a known sender of "junk" mail.
- Intercept mail from specific clients and forward that mail to a local VM user ID for further analysis.
- Restrict the ability to forward mail to a particular set of hosts.

## FORWARDMAIL Statement

### Purpose

The FORWARDMAIL statement is used to enable or disable mail forwarding, or to identify a user exit to be used to control mail forwarding.




### Operands

**YES**
Indicates mail forwarding is to be performed; this is the default.

**NO**
Indicates that no mail forwarding is to be performed. When SMTP determines a mail recipient is not on the local system, the RCPT TO: command will be rejected.

**EXIT**
Indicates a mail forwarding exit routine is being defined.

*exitname*
Indicates the name of the exit routine associated with this command; the default exit name is **SMTPFWDX**.

**EXEC**
Indicates the exit routine name specified on this command is the name of an EXEC; this is the default.

**TEXT**
Indicates the exit routine name specified on this command is the name of a text deck.

**ON**
Indicates the specified exit (being defined with this command) is to be enabled (turned on).

**OFF**
Indicates the specified exit (being defined with this command) is to be disabled (turned off).

## Examples

- The SMTP configuration file entry that follows will enable the mail forwarding exit routine SMTPFWDX EXEC.

```
ForwardMail
   exit smtpfwdx exec on
EndForwardMail
```

  When this entry is processed, the following text is displayed during server initialization:

```
Forward Mail                    : Exit SMTPFWDX EXEC ON
```

- This next entry defines the client verification exit routine SMTPFWDX TEXT, but disables its use once SMTP server initialization is complete.  Thus, mail forwarding will be allowed and performed.

```
ForwardMail
   exit smtpfwdx text off
EndForwardMail
```

  When this entry is processed, the following text is displayed during server initialization:

```
Forward Mail                    : Yes (Exit SMTPFWDX TEXT OFF)
```

## Mail Forwarding Exit Parameter Lists

Sample assembler and REXX exec exit routines are supplied for the mail forwarding exit. The assembler exit will have better performance characteristics than the REXX exit.

The parameter lists passed to the REXX and the assembler exit routines follow. When you tailor either of these exits, keep in mind the following:

- Because an identical exit parameter list definition is used for all three exits — the client verification exit, the mail forwarding exit and the SMTP commands exit, all parameters may not be meaningful for this exit. Those not used by this exit are indicated in the parameter lists; their values should be ignored.
- For the REXX exit, the value of an unused parameter will be such that any token-wise parsing will not be affected.

Parameter descriptions that pertain to both the REXX and assembler exits are provided on page 18.

# REXX Parameter List

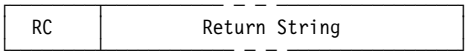
## Inputs

*Table 4. Mail Forwarding REXX Exit Parameter List*

| Argument | Description |
|---|---|
| ARG(1) | Parameter list defined as follows:<br>• Exit type<br>• Version number<br>• Reserved field<br>• Port number of SMTP server<br>• IP address of SMTP server<br>• Port number of client<br>• IP address of client<br>• Filename of note on disk<br>• Verify Client status<br>• Maximum length of Return String |
| ARG(2) | SMTP command string |
| ARG(3) | HELO name |
| ARG(4) | MAIL FROM string |
| ARG(5) | Client domain name |
| ARG(6) | Not used |

## Outputs

The following are returned to the caller in the RESULT variable via a REXX RETURN statement:




| Argument | Description |
|---|---|
| RC | The exit return code; this must be a 4 byte numeric value. |
| Return String | An exit-specified string; the returned value must have a length less than or equal to the maximum length passed to the exit. |

## ASSEMBLER Parameter List

**Note:** General Register 1 points to the parameter list.

*Table 5. Mail Forwarding ASSEMBLER Exit Parameter List.*

| Offset in Decimal | Len | In/Out | Type | Description |
|---|---|---|---|---|
| +0 | 4 | Input | Char | Exit type |
| +4 | 4 | Input | Int | Version number |
| +8 | 4 | Input | Int | Reserved field |
| +12 | 4 | Input | Int | Port number of SMTP server |
| +16 | 4 | Input | Int | IP address of SMTP server |
| +20 | 4 | Input | Int | Port number of client |
| +24 | 4 | Input | Int | IP address of client |
| +28 | 4 | Input | Int | Address of SMTP command string |
| +32 | 4 | Input | Int | Length of SMTP command string |
| +36 | 4 | Input | Int | Address of HELO name |
| +40 | 4 | Input | Int | Length of HELO name |
| +44 | 4 | Input | Int | Address of client domain name |
| +48 | 4 | Input | Int | Length of client domain name |
| +52 | 4 | Input | Int | Address of MAIL FROM string |
| +56 | 4 | Input | Int | Length of MAIL FROM string |
| +60 | 8 | Input | Char | File name of note on disk |
| +68 | 4 | Input | Int | Verify Client status |
| +72 | 4 | Output | Int | Address of Return String |
| +76 | 4 | Output | Int | Length of Return String |
| +80 | 4 | Input | Int | Maximum length of Return String |
| +84 | 8 | | | Not used |
| +92 | 4 | Input/ Output | Char | User Word 1 |
| +96 | 4 | Input/ Output | Char | User Word 2 |
| +100 | 4 | Output | Int | Return code from exit |

## Parameter Descriptions

**Exit type**

A four-character field used to indicate the type of exit called. For the mail forwarding exit, this will be **FWDX**.

**Version number**

The parameter list version number; if the parameter list format is changed, the version number will change. Your exit should verify it has received the version number it is expecting. The current version number is **1**.

**Port number of SMTP server**

The port number used by the SMTP server for this connection.

**IP address of SMTP server**

For the REXX exit, a dotted-decimal format IP address is provided; for the assembler exit, this is an IP address in decimal integer form. For multi-homed hosts, this address can be compared with the client IP address to determine in which part of the network the client host resides.

**Port number of client**

If the connection no longer exists, **-1** is supplied. Otherwise, this is the port number used by the foreign host for this connection.

**IP address of client**
For the REXX exit, a dotted-decimal format IP address is provided; for the assembler exit, this is an IP address in decimal integer form.

**Verify Client status**
A number that indicates client verification results. Possible values and their meaning are:

0 Client verification passed.
1 Client verification failed.
2 Client verification was not performed.
3 Client verification results are unknown.

**SMTP command string**
Contains the name specified on the RCPT TO: command. The recipient path, enclosed in angle brackets (< and >), is included. The recipient path may be in any valid path format; it has already been verified to have the correct syntax. Because the recipient address has been resolved, this string may not exactly match the data provided with the RCPT TO: command.

For example, if the following has been specified by the SMTP client:

```
RCPT TO: <usera@host1>
```

the SMTP command string might contain: `<usera@host1.com>`

**HELO name**
A string that contains the name specified on the HELO command; this string may be either:

- a domain name
- an IP address in decimal integer form that is prefixed by the number or (US) pound sign ("#" or X'7B'0)
- an IP address in dotted-decimal form, enclosed in brackets.

For example, if the command `HELO #123456` is provided by an SMTP client, this parameter would contain: `#123456`.

The name has already been verified to have the correct syntax.

**MAIL FROM string**
Contains the name specified on the MAIL FROM command. The sender path, enclosed in angle brackets (< and >), is included. The sender path may be in any valid path format; it has already been verified to have the correct syntax. Because the sender address has been resolved, this string may not exactly match the data provided with the MAIL FROM command.

For example, if the following has been specified by the SMTP client:

```
MAIL FROM: <userb@host2>
```

the SMTP command string might contain: `<userb@host2.com>`

**Client domain name**
The domain name that corresponds to the client IP address. This field will be a null string if:

- client verification was not performed
- the results of client verification are unknown
- a reverse lookup failed

In all other cases, this will be a domain name.

**User Word 1**
Provided for use by the assembler exit only. The user word specified upon return from this exit will be passed back in this field for any future calls; **0** is the initial value. The SMTP server does not use this value in any way.

**User Word 2**

Provided for use by the assembler exit only. The user word specified upon return from this exit will be passed back in this field for any future calls; **0** is the initial value. The SMTP server does not use this value in any way.

**Return String**

When the exit returns a return code of 1 or 5, this value is appended to the 551 or 550 reply code. When the exit returns a return code of 2, the *Return String* value should contain a VM user ID to which mail should be transferred.

**Maximum length of Return String**

The current maximum is 512 bytes; ensure the *Return String* length is less than this value. If the returned string is longer than the indicated maximum, the return string is truncated and the following message is displayed on the SMTP sever console:

```
Return data from exit exitname exittype too long, data truncated
```

Normal processing continues.

## Mail Forwarding Exit Return Codes

*Table 6. Mail Forwarding Exit Return Codes*

| Return Code | Explanation |
|---|---|
| 0 | Accept and attempt mail delivery. |
| 1 | Reject mail with: `551 Return String`<br>If a return string is not provided by the exit, the following default message will be used:<br>`551 User not local; please try user@otherhost`<br>If the server has already responded to the command, this return code will result in error mail being sent back to the sender. |
| 2 | Accept and forward to the local VM user ID specified by *Return String*. If the VM user ID is null or is not valid, the mail will be delivered to the local postmaster; the mail will not be delivered to the addressee. |
| 4 | Disable the exit. The following message will be displayed on the SMTP console:<br>`FORWARD MAIL EXIT function disabled`<br>The exit will no longer be called. SMTP will attempt to deliver this mail. |
| 5 | Reject this command with: `550 Return String`<br>If a return string is not provided by the exit, then the default message will be displayed:<br>`550 Access denied`<br>All future communications on this connection will be rejected with this 550 message. |
| *x* | Any return code other than the above causes SMTP to issue this message:<br>`Unexpected return from user exit exitname exittype, RC = rc`<br>SMTP treats this return code as if it were a return code of 0. |

## Mail Forwarding Sample Exits

Sample exit routines are supplied with the APAR; the supplied samples are:

**SMTPFWDX SEXEC** REXX exit routine

**SMTPFWDX ASSEMBLE** assembler exit routine

# Source Routes

## SOURCEROUTES Statement

### Purpose

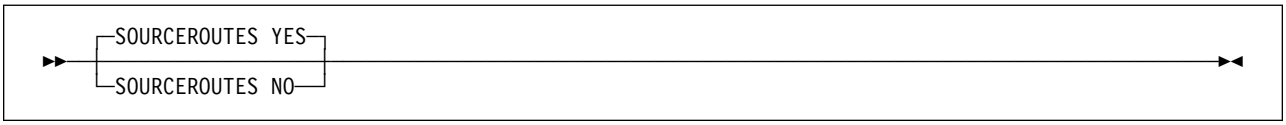
The SOURCEROUTES statement specifies whether the SMTP server should honor source routes.

A source route is a path that contains a routing list of hosts and a destination mailbox. The list of hosts is the *route* — information about how to the mail is to arrive at its final destination; the mail is passed from one host in this list to the next until it is delivered to the intended recipient.

The specification that follows is an example of a **source route**:

```
<@HOST1,@HOST2,@HOST3:USER@HOST4>
```

The list of hosts is HOST1, HOST2 and HOST3, and the destination is USER@HOST4. When source routes are honored the mail will be sent to HOST1, then to HOST2, then to HOST3 and finally to USER@HOST4. When source routes are not honored, mail is sent directly to USER@HOST4; the list of hosts is ignored.




### Operands

**YES**

Indicates that source routes received from clients will be honored when forwarding mail. For the previous sample source route, SMTP will send the mail to HOST1 for further processing by HOST1. This is the default.

**Note:** In a future release, the default will be changed to NO.

**NO**

Indicates that source routing is not to be honored; any host list will be ignored and only the destination host will be used. The mail recipient(s) will not see the host list that was ignored.

For the previous sample source route, SMTP will send the mail directly to USER@HOST4; the HOST1, HOST2 and HOST3 hosts will be ignored.

**Note:** Mail containing source routes will not be rejected.