

Inter-Office Memorandum

To	Communication Protocols	Date	February 11, 1979
From	Ed Taft	Location	PARC/CSL
Subject	Pup Mail Transfer Protocol (Edition 6)	File	[Maxc1]<Pup>MailTransfer.bravo

XEROX

The Mail Transfer Protocol (MTP) is a means by which messages may physically be transmitted between machines. Mail transfer involves two operations that cannot be characterized as ordinary file access operations (such as those available via FTP or Pine). These operations are the appending of new messages to a user's *mailbox* and the retrieval of the mailbox contents for absorption into a *Laurel mail file*.

The MTP, in conjunction with a file access protocol (FTP or Pine), is sufficient to support the Laurel message system. All facilities described in this document are now (or will shortly be) implemented by the Maxc and IFS mail servers. Note that this is the *interim* transport protocol; some of the mechanisms defined within it are admittedly clumsy and inflexible. The distributed transport mechanism now under development will be substantially different.

Changes to this edition: Recipient names may be qualified by *.host* to invoke forwarding; Sender property now mandatory.

Overview of the Protocol

A mail transfer is an interaction between a *user* process desiring to append a message to or extract the contents of a mailbox and a *server* process residing in the machine in which the mailbox is located. The interaction is initiated by the user.

The protocol is a variant of the File Transfer Protocol. The user establishes a BSP connection with the server at a well-known socket. Commands and responses consist of a BSP Mark byte followed by some data and terminating at the next Mark. Depending on the command, the data may consist of a property list, a mail file being transferred, or explanatory text. In the third case, the text may be preceded by a machine-readable code byte intended to facilitate mechanical processing of the command or response.

The reader should consult the *FTP specification* (file <Pup>FTPSpec.press) for complete information on command/response sequences and property list syntax. The MTP, while using the FTP framework, assigns some additional command/response Mark types and property names. Therefore, MTP and FTP can coexist in any particular implementation.

Mail Delivery

This portion of the protocol is used to transport a message from user to server for one of two purposes: *delivery* of the message to a mailbox residing in the server system, or *forwarding* of the message to another server. Not all mail servers need implement forwarding, but those that do not are useful only to support self-contained user communities with a single central mail server.

In general, a mailbox *name* is in the form *SimpleName.Host*, where *Host* is the name of the host on

which the mailbox is located and *SimpleName* is the name by which the mailbox is known on that host. It is the user's responsibility to determine the host at which a mailbox resides. Experts will recognize the similarity between this and the *SimpleName.NamingAuthority* scheme of the distributed transport mechanism. However, this similarity is syntactic only; the *NamingAuthority* is independent of mailbox location.

A *SimpleName* unqualified by *.Host*, or qualified by *.Host* where *Host* is the name of the mail server involved in the delivery transaction, implies a request to deliver a message to a mailbox located on the server machine. The server should immediately reject the request (by means of the [Mailbox-exception] mechanism described below) if no such mailbox actually exists.

A *SimpleName* qualified by some other *Host* implies a request to forward a copy of the message to the mail server at that *Host*. In this case (assuming *Host* is valid), the mail server should accept a message for *SimpleName.Host* and, at the earliest convenient time, invoke a MTP user process to deliver the message to *Host*. It is intended that this forwarding operation be accomplished through a single hop, not multiple hops.

A message to be forwarded to an Arpanet recipient is addressed to *Name@ArpaHost.ArpaGateway*, where *ArpaGateway* is simply an alias for *Maxc1*. From the viewpoint of the MTP, *Name@ArpaHost* is the *SimpleName* for a mailbox located at the host named *ArpaGateway*.

Since in general the attempt to forward the message to *Host* takes place after the initial delivery of the message to this mail server, the forwarder must be prepared to cope with the possibility that the recipient name will be rejected by the ultimate *Host* or that the message will otherwise be undeliverable within a reasonable time. The MTP therefore requires that the message be accompanied by a *Sender* property that may be used to direct a failure notification to the sender of the message.

The mail delivery protocol is as follows:

```
User: [Store-mail] <property list> <property list> ... [EOC]
Server (optional): [Mailbox-exception] <code> <index> <text>
Server: [Yes] <code> <text> [EOC] or [No] <code> <text> [EOC]
If the server said [Yes], then
User: [Here-is-file] <message> [Yes] <code> <text> [EOC]
Server (optional): [Mailbox-exception] <code> <index> <text>
Server: [Yes] <code> <text> [EOC] or [No] <code> <text> [EOC]
```

Each of the <property list>s in the [Store-mail] command designates a mailbox to which the message is to be delivered. The use of multiple property lists, rather than a single property list containing multiple recipient names, is an artifact of an earlier design of the protocol; it is no longer useful but is retained for compatibility reasons.

The following properties are defined:

(Mailbox *name*)

Designates the desired recipient of the message. A Mailbox property is required in each property list. In general, *name* should be in the complete form *SimpleName.Host*, but *.Host* may be omitted if the recipient's mailbox is located at this mail server.

(Sender *name*)

Designates the name of the sender of the message, i.e., the name of the mailbox to which notification should be sent if the message cannot be delivered. In general, *name* should be in the complete form *SimpleName.Host*, but *.Host* may be omitted if the sender's mailbox is located at this mail server (as it ordinarily will be for a message delivered by Laurel).

This property must appear once in one of the <property list>s. Current practice is that the Sender need not be authenticated or otherwise registered as a user of this mail server. The server should require only that the Sender property be well-formed.

The <message> should conform to the default FTP text file format (i.e., Type Text, End-of-line-convention CR, though these parameters need not be specified in the <property list>). Furthermore, the message should include an Arpanet-style message header (see RFC #733, "Standard for the format of Arpa network text messages") to facilitate mechanical processing of the message. There are some subtle problems involved in constructing addressee fields in a fashion that permits mechanical generation of replies. See the memo "Interim IFS Mail Forwarding", file [Maxc1]<Pup>IFSForwarding.press, for details.

The <message> should not include a leading *stamp*; it is the server's responsibility to insert any required message-separating information at the moment the message is actually appended to the mailbox. Such information is specific to the server machine and is never sent over the network. On Maxc, the mail server prefixes a standard Tenex-format message stamp for compatibility with existing programs that manipulate mailboxes.

The server may respond [No] to the [Store-mail] request due to an error in the <property list> (e.g., bad format or absence of any valid mailbox name). The server may respond [No] to the [Here-is-file] command due to failure to receive the <message> correctly, implying that the message was not delivered to *any* mailbox. Generally the server should buffer the message until it has been successfully received in its entirety before beginning to append it to mailboxes. This ensures that if the transfer fails in the middle no mailboxes will have been damaged.

The server may precede the [Yes] response to either [Store-mail] or [Here-is-file] with one or more [Mailbox-exception] replies that refer to specific mailboxes to which the server cannot deliver the message for some reason. The parameters of this reply are:

<code>

Optionally provides a machine-interpretable error code that may facilitate automatic handling of the error condition. The <code> is an 8-bit byte analogous to [Yes] and [No] codes, though taken from a different space. Zero indicates an unspecified error.

<index>

Identifies the affected property list in the user's [Store-mail] command. The first property list is numbered 1, the second 2, and so on. Unlike the <code>, the <index> is expressed as a decimal number represented by Ascii digits, followed by a space. This peculiar encoding is used, rather than a property list, for the convenience of existing Tenex mail-sending software that does not presently have an FTP property-list parser.

<text>

Provides a text string for human consumption. Inclusion of an informative message is strongly encouraged.

The server will typically generate [Mailbox-exception] replies in response to [Store-mail] for mailboxes it cannot locate (or that do not reside on the server machine, if the server does not provide forwarding). [Mailbox-exception] replies in response to [Here-is-file] should be comparatively rare; they indicate unexpected failure to deliver mail to known mailboxes. The numbering of mailbox property lists in the second case is unaffected by any previous [Mailbox-exception]s.

Important: The server must not begin generating [Mailbox-exception]s until it has received the entire [Store-mail] argument list; otherwise a deadlock may result. This comment actually applies to the FTP as a whole: the user and server are expected to transmit alternately, never simultaneously.

Mail Retrieval

This portion of the protocol is used to retrieve (send from server to user) the contents of a mailbox and then to reset the mailbox to empty. The two operations must be done atomically so as to prevent loss of new messages.

The command/response sequence is as follows:

```

User: [Retrieve-mail] <property list> [EOC]
Server: [No] <code> <text> [EOC] or
Server: [Here-is-property-list] <property list> [Here-is-file] <first message>
      [Here-is-property-list] <property list> [Here-is-file] <second message>
      ...
      [Yes] <code> <text> [EOC] or [No] <code> <text> [EOC]
If the server sent the file, then
User: [Flush-mailbox] [EOC]
Server: [Yes] <code> <text> [EOC] or [No] <code> <text> [EOC]

```

The <property list> in the [Retrieve-mail] command must contain a Mailbox property identifying the mailbox to be operated upon, plus any User-name, User-Password, etc., properties required to gain access to the mailbox.

The <property list> preceding each message provides auxiliary information about the message that may be of use to the user. The properties include:

(Length <decimal number>)

Gives the length of the message in bytes. This property is required. It has been included for the convenience of Laurel. Its usefulness is presently in some doubt, and generating it is somewhat inconvenient for the Maxc mail server because it requires two passes over the message.

(Date-received <date and time>)

Gives the date and time at which the message actually arrived at the mailbox.

(Opened Yes|No)

Indicates whether or not the message has already been "opened", i.e., examined while still residing in the mailbox. The default is No. This is relevant only for mailboxes stored on Maxc, since they may be examined by Msg.

(Deleted Yes|No)

Indicates whether or not the message has been "deleted" before being removed from the mailbox. Again, this is relevant only for mailboxes stored on Maxc.

The contents of the mailbox are retrieved in the default FTP text format. Any server-specific information in the mailbox (such as separators or stamps) should be stripped off by the server. The [Here-is-property-list] and [Here-is-file] marks serve to delimit the individual messages.

Note that the server must either prevent delivery of new messages to the mailbox during the interval between receipt of the [Retrieve-mail] and [Flush-mailbox] commands, or else take measures to ensure that [Flush-mailbox] deletes only those messages that were retrieved in response to [Retrieve-mail].

Opcode Assignments

The well-known socket number for a mail server is 7.

The following Mark types are assigned, in addition to the ones in the FTP specification (all numbers octal):

```

20    [Store-mail]
21    [Retrieve-mail]
22    [Flush-mailbox]
23    [Mailbox-exception]

```

Assigned [No] codes, in addition to the ones defined in the FTP specification, are:

- 40 No valid Mailbox in property list.
- 41 Illegal Mailbox property syntax.
- 42 Illegal Sender property.

The following [No] codes are added to the FTP specification:

- 107 Transient, non-specific server or file-system failure.
- 110 Permanent, non-specific server or file-system failure.

In general, all [No] codes except those in the range 40-43 and 110 signify transient rather than permanent failures, and the user should persist in attempting to deliver the mail (e.g., queue it for later transmission).

The following codes are defined for [Mailbox-exception] replies. Note that every [Mailbox-exception] reply indicates failure to deliver a message to the designated mailbox, regardless of the code; the purpose of the code is to facilitate automatic diagnosis and handling of exceptional conditions.

- 0 Unspecified failure.
- 1 Cannot locate Mailbox.
- 2 Mailbox not resident on server machine and no forwarding provided.
- 3 Cannot deliver to mailbox due to transient error condition.
- 4 Cannot deliver to mailbox due to permanent error condition.

Related Protocols

Two mail-related protocols exist as part of the Miscellaneous Services collection. These are the Mail Check request, for determining quickly whether or not a mailbox contains new mail, and the User Authentication request, for determining the validity of a name/password combination. See the *Miscellaneous Services* specification for complete information.