Network Working Group
Request for Comments: 524
NIC: 17140

J. White
SRI-ARC

13 June 1973

A Proposed Mail Protocol

AUTHOR'S INTENT

This is the document I offered in (15146,) to write. It's a proposed specification for handling mail in the Network -- a Mail Protocol.

Main handling is currently implemented as two FTP commands, MAIL and MLFL, which permit an FTP user process to deliver a file or string of text to an FTP server process, designating it as mail to be made available to a user, identified by a local name, in its host. The protocol proposed here is much richer than that, both in terms of the functions it supports, and in terms of the flexibility it provides.

Although one can (I think) and might, implement software on the basis of this document, this REALLY IS a Request for Comments. Comments, questions, position papers are solicited. There are, I'm sure, bugs in the protocol specified here, and I hope that readers will point them out via RFC as they discover them.

Various members of the Network community have, during the last few months, pointed out to me specific inadequacies in the existing mail commands and asked me to be conscious of them in designing a new protocol. I've tried to do that. If anyone feels that his concern wasn't properly dealt with here, or that it slipped through the cracks entirely (for which I apologize in advance), I would appreciate it if he would prod me once more.

INTRODUCTION

THE MAIL PROTOCOL ENVIRONMENT

The Mail Protocol (MP) is implemented by Mail user and server processes, in keeping with the model for previous high-level protocols. The Mail user and server processes are further specified to be also FTP user and server processes, respectively. That is, MP is implemented as a set of commands accessible from within the FTP command space.

The MP command set is defined to lie conceptually within a subsystem, invoked from the FTP command space with the command MAIL <CFLF>.

NOTE: Since a command called 'MAIL' already exists within the FTP command space, the command name 'XMAIL' might substitute for 'MAIL' while the current mail commands are being phased out.

The MP command set may or may not (according to the implementation of a particular server) be implemented by a process distinct from that which implements FTP proper.

The following are implications of the 'subsystem' concept, of which the reader (and implementer) must be aware:

(1) Names of MP commands are known only within the MP subsystem. MP commands must (and should naturally) be rejected by the server if the user process presents them outside of the subsystem.

(2) Exit from the Mail subsystem (to the FTP command space) is effected with and only with the command EXIT <CRLF>. FTP commands must be rejected by the server if the user presents them while inside the subsystem (i.e., before EXIT is issued).

(3) The same command name may be assigned without ambiguity to two entirely different commands, provided that one lies within the FTP command space and the other within MP, the two being distinguishable by their contexts. MP and FTP therefore do not compete for command names, and MP command names may be chosen without regard for the environment in which the subsystem resides.

NOTE: It so happens that there are commands DEFINED within MP which duplicate the functions of FTP commands and bear the same names. The effective result is that some commands are explicitly allowed within MP. The reader will understand that this fact is consistent with the conventions described in 1-3 above, and that no ambiguities result.

The subsystem concept (if not the name 'subsystem') is taken from Mike Padlipsky's Unified User Level Protocol (UULP), which the author of the present document strongly supports. The fact that MP is accessed from FTP, rather than both FTP and MP being accessed independently from a more general executive program, is simply a concession to the fact that FTP is widely implemented and UULP isn't. The author hopes that protocol development will, in the near future, begin to proceed along the lines exemplified by UULP.

MP conforms to FTP in general syntax. In particular, commands and their responses are strings of NVT characters; command names are limited to four or fewer, upper- or lower-case, alphameric characters, and are terminated by the character SP; commands are generally terminated with the TELNET New Line sequence (CR LF); command responses contain both numeric (process readable) and text (human readable) portions. Both reader and implementer are referred to the FTP protocol document for a detailed description of such matters; no attempt has been made to duplicate the discussion in the present document.

The FTP protocol document assigns those replies whose second digit is '6' to RJE functions. In like manner MP appropriates those reply codes whose second digit is '7' for reporting results peculiar to its functions. It is, however, the author's position that FTP, MP, and the RJE protocol are all best implemented as subsystems under a common UULP executive, in which case a single subset of the reply space could be used unambiguously by all three protocols (and any yet to be defined), since every reply would implicitly be qualified by the name of the subsystem from which it emanates.

THE MAIL MODEL

MP defines mail to be text communicated between users (both human and processes) in less that (but ideally approaching) real time. The definition excludes so-called console-to-console communication, where users exchange information at the character or line level.

Pieces of mail are characterized by such attributes as title, content, author, and recipient. A piece of mail may be a one- or two-line message sent from one individual to another, a draft of a document sent by one individual to a design group for review, a polished, formal document sent from one group to another, a message sent to a human user by a process (e.g., an RJE server process might notify a user by Network Mail when his job has completed), etc. All such forms of communication are mail and are supported by MP.

Pieces of mail can be forwarded from one location to another

Pieces of mail can be replied to.

The identity of the author of a piece of mail can be verified, avoiding forgery and misrepresentation.

Pieces of mail can be permanently recorded, assigned a long-term identifier by which they can be forever be retrieved for reference, and entered in catalogs. And access to such recorded mail can be restricted to a specified subset of the user community.

Some hosts accept mail whose recipients reside elsewhere in the Network, and assume responsibility for delivering the mail to them (worrying about retrying delivery when hosts are down, etc.), and acknowledging its delivery to the sender.

The picture being painted for the reader is one in which processes cooperate in various ways to flexibly move and manage Network mail. The author claims (without proof, of course) that the picture will in the future get yet more complicated, but that the proposal specified here can be conveniently enlarged to handle that picture too.

ORGANIZATION OF THIS DOCUMENT

The rest of this document consists of the following components:

GLOSSARY

The concepts introduced briefly in the section above are more formally defined, and their manner of representation in the protocol specified.

MP FUNCTIONS

The command sequence is defined by which a user process initiates each of the logical functions (e.g., Distribution, Recording, Delivery) which can be performed by a Mail server process.

EXAMPLE

An example of the command-response exchange between a user and server is given.

COMMAND SUMMARY

A summary of MP commands is given.

COMMAND REPLIES

Reply code assignments are given and briefly explained.

FORMAL SYNTAX

The formal syntax of the command language is specified.

In all sections but the last (i.e., the formal syntax presentation), verbose keyword forms are employed, in the interests of clarity. These verbose forms have no existence anywhere but in this document; in implementing a Mail user or server process, the terse keyword forms which appear in the formal syntax presentation are to be employed

GLOSSARY

Terms are listed here in alphabetical order. Words or phrases which appear in the definitions with initial letters capitalized are themselves formally defined elsewhere in the glossary.

ACCESS LIST (for a piece of Recorded Mail)

That set of individuals with access to a piece of Recorded Mail, and for each such individual, the type(s) of access granted to him.

An Access List is represented in the Protocol as a series of command pairs (juxtaposed in the command stream), each pair consisting of an ACCESS command followed immediately (and optionally) by an ACCESSTYPES command. Each pair of commands corresponds to one individual in the set.

ACCESS <individual> <CA>

ACCESSTYPE <accesstypes> <CA>

Command arguments identify the Individual to whom access is granted, and specify the kind(s) of access allowed him. Either Read Access, Controlling Access, or both may be granted.

If no Individual is specified, All is implied. In the absence of an explicit ACCESSTYPES command, one with only Read Access specified is to be assumed.

In the absence of an explicit Access List, one granting Read Access to All and Controlling Access to the Author(s) and the Clerk is to be assumed.

ACKNOWLEDGMENT (for a piece of Mail)

A form of Unrecorded Mail, generated by a Distribution Agent, whose Recipient is the Monitor for a previous piece of Mail, which acknowledges Delivery -- successful or otherwise -- to the Recipient(s) of that first piece of Mail.

An Acknowledgment bears the Serial Number of the Mail it acknowledges, as the Reference Serial Number.

ACKNOWLEDGMENT CONDITION (for Acknowledgments)

The attribute of an Acknowledgment which determines the circumstances under which it will be generated by the Distribution Agent.

The following Acknowledgment Conditions are defined:

ALWAYS

Acknowledgment is given when all Deliveries are complete, regardless of whether or not they are all completed successfully.

FAILURE

Acknowledgment is given when all Deliveries are complete if and only if Delivery to one or more Recipient(s) fails.

NEVER

An Acknowledgment is never made.

An Acknowledgment Condition is represented in the Protocol by the command:

ACKCONDITION <ackcondition> <CA>

In the absence of an explicit ACKCONDITION command, one with an argument of FAILURE is to be assumed.

ACKNOWLEDGMENT TYPE (for Acknowledgments and Progress Reports)

The attribute of an Acknowledgment or Progress Report which determines the nature of its Content.

The following Acknowledgment Types are defined:

TERSE

The Content of a TERSE Acknowledgment or Progress Report is specified by the Protocol to be an unembellished list of the Mail's Recipient(s), and the current Delivery Status for each (except that those Recipient(s) whose Delivery Status is SUCCESSFUL shall not be included in the list).

The Content of a TERSE Acknowledgment is one or more instances of the following:

<deliverystatus> <individual> <CRLF>

TERSE Acknowledgments and Progress Reports are intended to be process-readable.

VERBOSE

The Content of a VERBOSE Acknowledgment or Progress Report is not specified by the Protocol, but might include a list of those Recipient(s) to whom the Mail could not be delivered and why, the times at which Delivery was made to others, etc.

VERBOSE Acknowledgments and Progress Reports are intended to be human-readable.

An Acknowledgment Type is represented in the Protocol by the command:

ACKTYPE <acktype> <CA>

In the absence of an explicit ACKTYPE command, one with an argument of TERSE is to be assumed.

ALL

Every conceivable Individual.

AUTHOR (of a piece of Mail)

An Individual (there may be more than one) given formal recognition for having authored a piece of Mail.

AUTHOR LIST (for a piece of Mail)

That set of Individuals who are Author(s) of a piece of Mail.

An Author List is represented in the Protocol as an Individual List of type AUTHOR.

CATALOG (of Recorded Mail)

A named data base containing the Citation for each member of a set of logically related pieces of Recorded Mail.

CATALOG LIST (for a Piece of Recorded Mail)

That set of Catalogs which each contain the Citation for a piece of Recorded Mail

A Catalog List is represented in the Protocol as a series of instances (juxtaposed in the command stream) of the following command. Each instance corresponds to one Catalog in the set.

```
CATALOG <catalog> <CA>
```

CITATION (for a piece of Recorded Mail)

The Static and Dynamic Attributes of a piece of Recorded Mail.

CITATION COMPONENT

Any one of the Static or Dynamic Attributes which comprise a Citation.

CITATION RETRIEVAL (for a piece of Recorded Mail)

The act of retrieving selected Citation Component(s).

CITATION TEMPLATE

A specified subset of the Citation Component(s) for a piece of Recorded Mail.

A Citation Template is represented in the Protocol by the command:

CITATIONTEMPLATE <citationtemp> <CA>

The argument is a list of Citation Component(s). In the absence of an explicit CITATIONTEMPLATE command (or if the argument is null), one specifying Content only is to be assumed.

CLERK

That Individual who prepares a piece of Mail for Recording, Distribution, or Delivery. Conceptually, the Individual with proof-reading responsibility for the piece of Mail.

A Clerk is represented in the Protocol as an Individual List of type CLERK and length 1.

COMMENTS (for a piece of Mail)

An informal, perhaps verbose description of the Content of a piece of Mail, or any other information the Author(s) wish to have made accessible to the Recipient(s) of the Mail.

Comments are represented in the Protocol by the command:

COMMENTS <comments> <CA2>

In the absence of an explicit COMMENTS command, one with a null argument is to be assumed.

CONTENT (of a piece of Mail)

It’s text.

Content is represented in the protocol by either of the two commands below:

FILE <CA>

The FILE command implies that the Content of the Mail will be transmitted (immediately) as a file using the FTP data transfer commands (e.g., BYTE, SOCK, TYPE) currently in effect. FILE is exactly equivalent in use to an FTP STOR command (in its use of data transfer commands, in its establishment of the data connection, etc.), except that no pathname is required, and the server, rather than depositing the transmitted file in his file system, disposes of it in a manner appropriate for Mail.

TEXT <string> <CA2>

The TEXT command implies that the Content of the Mail follows on the TELNET connection as a series of lines, each delimited from the preceding one by CR LF, and terminated finally by a CA2.

CONTROLLING ACCESS (to a piece of Recorded Mail)

The right of an Individual to modify a Dynamic Attribute of a piece of Recorded Mail.

Recording Agents permit an Individual to modify a Dynamic Attribute of a piece of Recorded Mail if and only if he can properly identify himself as someone having Controlling Access to that Mail.

CREATION DATE (of a piece of Mail)

The date and time at which the final draft of a piece of Mail is completed by the Clerk before he releases it to a Delivery, Distribution, or Recording Agent for further processing. A single Creation Date is associated with each piece of Mail. In general, this date is different from the Delivery or Recording Date, since Mail often must be queued (for another host to come up) within the Delivery, Distribution, or Recording Agent's host before Delivery of Recording can occur.

A Creation Date is represented in the Protocol by the command:

CREATIONDATE <datetime> <CA>

CUTOFF INTERVAL (for Distribution of a piece of Mail)

That period of time, measured from the Distribution Date, after which the Distribution Agent is to abandon Delivery attempts for those Recipient(s) to whom Delivery has not yet been effected.

A Cutoff Interval is represented in the Protocol by the command:

CUTOFF <interval> <CA>

In the absence of an explicit CUTOFF command, one specifying an Interval of 7 days is to be assumed.

DELIVERY (of a piece of Mail)

The act of transmitting a piece of Mail to the host of one of it's Recipient(s).

DELIVERY AGENT

A process which effects Delivery of a piece of Mail. A Distribution Agent is by nature also a Delivery Agent.

DELIVERY DATE (of a piece of Mail to one of its Recipient(s))

The date and time at which a piece of Mail is Delivered by the Delivery Agent to a Recipient's system. A multitude of Delivery Dates, one for each Recipient, are associated with each piece of Mail.

DELIVERY STATUS (for a piece of Mail with respect to a Recipient)

A measure of the extent to which a Delivery Agent has been successful, at a given point in time, in Delivering a piece of Mail to one of its Recipient(s). A multitude of Delivery Status', one for each Recipient, are associated with each piece of Mail.

The following Delivery Status' are defined:

FAILED

Delivery was rejected by the Recipient's system (e.g., the connection request was rejected, the Mail server process doesn't support Delivery, the Recipient was not recognized by the server).

SUCCESSFUL

Delivery was accomplished successfully.

TIMED OUT

Either the Recipient's host was disconnected from the Net at every Delivery attempt, or no Mail server process has ever responded to the connection attempt. Hope of Delivery has been abandoned.

WAITING

Either the Recipient's host has been disconnected from the Net at every Delivery attempt, or no Mail server process has yet responded to the connection attempt. Delivery attempts are continuing periodically.

UNATTEMPTED

No delivery attempt has yet been made.

DELIVERY TYPE (for a Delivery)

The nature of the piece of Mail being delivered.

The following Delivery Types are defined:

FORWARD

A Delivery of type FORWARD represents a piece of Recorded or Unrecorded Mail which is being Forwarded.

MAIL

A Delivery of type MAIL represents a piece of Recorded or Unrecorded Mail whose ultimate source is an Individual. This is the "normal" Delivery type.

NEGATIVE ACKNOWLEDGMENT

A Delivery of type NEGATIVE ACKNOWLEDGMENT represents a piece of Unrecorded Mail generated by a Distribution Agent and acknowledging unsuccessful or partially unsuccessful Delivery of a previous piece of Mail (identified by Reference Serial Number) to it's Recipient(s). The Recipient for this piece of "system" Mail is the Monitor for the previous piece of Mail.

POSITIVE ACKNOWLEDGMENT

A Delivery of type POSITIVE ACKNOWLEDGMENT represents a piece of Unrecorded Mail generated by a Distribution Agent and acknowledging successful Delivery of a previous piece of Mail (identified by Reference Serial Number) to it's Recipient(s). The Recipient for this piece of "system" Mail is the Monitor for the previous piece of Mail.

PROGRESS REPORT

A Delivery of type PROGRESS REPORT represents a piece of Unrecorded Mail generated by a Distribution Agent and reporting the Delivery of a previous piece of Mail (identified by Reference Serial Number) to it's recipient(s). The Recipient for this piece of "system" Mail is the Monitor for the previous piece of Mail.

REPLY

A Delivery of type REPLY represents a piece of Recorded or Unrecorded Mail generated in reply (or pertaining) to a previous piece of Mail (identified by Reference Serial Number).

Delivery Type is represented in the Protocol by the command:

DELIVERYTYPE <deliverytype> <CA>

In the absence of an explicit DELIVERYTYPE command, one with argument of MAIL is to be assumed.

DISTRIBUTE DATE (for a piece of Mail)

The date and time at which a piece of Mail is presented to a Distribution Agent for Distribution.

DISTRIBUTION (of a piece of Mail)

The act of Delivering a piece of Mail to its Recipient(s). Distribution can be effected by either the Clerk's Delivery Agent, or by a Distribution Agent acting on his behalf.

DISTRIBUTION AGENT

A Mail server process which acts as intermediary in the delivery process by accepting Mail for Recipient(s) in hosts other than its own, and then assuming responsibility for Delivering the Mail to them and returning Acknowledgment to the appointed Monitor.

DISTRIBUTION LIST

In the Delivery or Distribution of a piece of Mail, that set of Individuals who are to receive Delivery of the Mail.

In the Recording of Mail, that set of Individuals who have received formal and authorized Delivery of a piece of Mail. The list is kept current by Distribution Agents. Of course, any Individual with Read Access to the Mail can himself Deliver it informally to anyone he chooses without that fact's being reflected in the Distribution list.

A Distribution List is represented in the Protocol as a series of command quintuplets (juxtaposed in the command stream), each quintuplet consisting of a RECIPIENT command, followed immediately (and optionally) by any or all of the following in the order given: a GENERALDELIVERY, a GREETING, a SIGNATURE, and a DISPOSITION command.

Each quintuplet corresponds to one individual in the set.

RECIPIENT <individual> <CA>

GENERALDELIVERY <CA>

This command is appropriate only in the context of the Delivery function. If the previous RECIPIENT command illicits the reply:

474 Recipient unrecognized: is General Delivery OK?

the issuance of the GENERALDELIVERY command constitutes consent to proceed with General Delivery to that Recipient. If no such consent is given, the RECIPIENT command stands rejected. Unsolicited (i.e., unprompted for) GENERAL DELIVERY commands in the Distribution List are treated by the server as NOPs.

GREETING <greeting> <CA>

The GREETING command specifies the Greeting to be seen by the Recipient.

If the first quintuplet in the list contains no GREETING command, one with a null argument is assumed. Thereafter, in the absence of an explicit GREETING command, one identical to that for the previous quintuplet is assumed.

SIGNATURE <signature> <CA>

The SIGNATURE command specifies the Signature to be seen by the Recipient.

If the first quintuplet in the list contains no SIGNATURE command, one with a null argument is assumed. Thereafter, in the absence of an explicit SIGNATURE command, one identical to that for the previous quintuplet is assumed.

DISPOSITION <disposition> <CA>

The DISPOSITION command identifies the intent with which the Mail is Delivered to the Recipient by the Author(s), and may take any, all, or none of the following as arguments:

RSVP

The Author(s) request a Reply from the Recipient.

ACTION

The Author(s) expect some action on the part of the Recipient. If ACTION doesn't appear, then the Mail is intended for the Recipient's information only.

INTERRUPT

The Author(s) declare that examination of the Mail by the Recipient is urgent. In such cases, the Recipient's Mail server process may, upon Delivery, choose to interrupt the Recipient if he happens to be logged in at a terminal.

No specific action in response to the presence of any of these arguments is required; the server is free if he likes to treat DISPOSITION commands as NOPs.

The absence of a DISPOSITION command implies one with no arguments (i.e., for the Recipient's information only, no Reply required, and not urgent).

DYNAMIC ATTRIBUTES (of a piece of Recorded Mail)

Those attributes of a piece of Recorded Mail -- Distribution List, Access List, and Catalog List -- which, though given initial values at Recording Time, can always be modified by an Individual with Controlling Access to the piece of Mail.

FORWARDING (of Mail received for an Individual)

The act of transferring that set of Mail which has been previously Delivered to but not Read by an Individual, to another Individual.

Users who are known at more than one host can cause their unRead Mailto be gathered in to a central location by performing the Forwarding function at each such host (both Individuals being, in this case, instances of the same User). Mail which has been Forwarded is considered to have been Read at its source.

FORWARDEE

That Individual whose Delivered but UnRead Mail is to be Forwarded.

A Forwardee is represented in the Protocol as an Individual List of type FORWARDEE and length 1.

GENERAL DELIVERY (of a piece of Mail to an unrecognized Recipient)

The act on the part of a server of accepting Delivery of a piece of Mail on behalf of an intended Recipient whose name the server doesn't recognize. The server retains the Recipient's name, and makes it and the other information provided by the user process available to some competent person, who attempts to make sense of the Recipient's name. If the Recipient is recognized, the Mail is 'hand' delivered to the appropriate Individual.

General Delivery of a piece of Mail to one of its intended Recipient(s) is performed only after the server informs the user process of its intent and receives the user process' consent. If that consent is not given, or if the server doesn't implement General Delivery, the server rejects the Delivery attempt for that Recipient.

Consent for General Delivery is represented in the Protocol by the command

GENERALDELIVERY <CA>

GREETING (for the Delivery of a piece of Mail to a Recipient)

A short greeting to a Recipient of a piece of Mail. 'Dear Dave' is a valid and perhaps typical Greeting.

A Greeting is represented in the Protocol by the command:

GREETING <greeting> <CA>

ID (for an Individual)

The password which an Individual may have to present to a Mail server process, to prove his identity.

An Id is represented in the Protocol by the command:

ID <id> <CA>

Ids have nothing to do with accounting, and when required by a server, they're required only to protect that server from forgery or misrepresentation.

INDIVIDUAL

An instance of a User, identified by NIC Ident, or by the combination of host and Mailbox Name.

INDIVIDUAL LIST (of type "t" and length "n")

A set of Individuals.

An Individual List is represented in the Protocol as a series of "n" command pairs (juxtaposed in the command stream), each pair consisting of a "t" command, followed immediately by an ID command. Each pair corresponds to one Individual in the set.

The ID command is given by the Mail user process at the option of the Mail server process; and whenever the server requires it, he must prompt for it with an appropriate reply to the preceding "t" command. If no such prompt is given, the user process is not obliged to provide the ID command, but may if it chooses, in which case the server is obliged to treat it as if it had been prompted for and found correct.

The ID command is a mechanism by which the server can assure himself that the user process is not acting without proper authorization from the Individual(s) involved, i.e., a mechanism by which a server can protect himself against forgery, misrepresentation, etc.

"t" <individual> <CA>

ID <id> <CA>

MAIL

A body of text communicated from one set of Individual(s) to another, in less than (but ideally approaching) real time.

MAILBOX NAME

The name a User employs at a host to send and receive Mail.

MONITOR (for a piece of Mail)

The Individual who is the recipient for Acknowledgments and Progress Reports.

A Monitor is represented in the Protocol as an Individual List of type MONITOR and length 1.

Monitor defaults to the Clerk if not explicitly specified.

PROGRESS REPORT (for a piece of Mail)

A form of Unrecorded Mail, generated periodically during the distribution process by a Distribution Agent, whose Recipient is the Monitor for a previous piece of Mail, and whose Content is a list of the Recipient(s) and the current Delivery Status for each. A Progress Report bears the Serial Number of the Mail whose status it reports, as the Reference Serial Number.

PROTOCOL

The Mail Protocol (MP).

READ (a piece of previously-Delivered Mail)

The act, on the part of the User, of examining a piece of Delivered Mail.

READ ACCESS (to a piece of Recorded Mail)

The right of an Individual to retrieve the Content of a piece of Recorded Mail.

Recording Agents permit an Individual to retrieve the Content of a piece of Recorded Mail if and only if he can properly identify himself as someone having Read Access to that Mail. An Individual can retrieve the Citation (except Content) from the Recording Agent independently of whether or not he has Read Access to the Mail.

READ DATE (of a piece of Mail for one of its Recipient(s))

The date and time, necessarily following Delivery, at which a piece of Mail is Read by a Recipient. A multitude of Read Dates, one for each Recipient, are associated with each piece of Mail.

RECIPIENT (of a piece of Mail)

An Individual who has or is to receive Delivery of a piece of Mail.

RECORDED MAIL

A piece of Mail whose Citation is available on a long-term (indefinite) basis from a Recording Agent.

RECORDING

The service provided by a Recording Agent.

RECORDING AGENT

A Mail server process which accepts Mail, permanently Records its Citation, and assigns a pathname by which that information can at any time be retrieved by an Individual with appropriate access.

RECORDING DATE

The date and time at which a piece of Mail is presented to a Recording Agent for Recording. A single Recording Date is associated with each piece of Recorded Mail.

REFERENCE SERIAL NUMBER (for an Acknowledgment, Progress Report, or Reply)

The Serial Number of the piece of Mail to which an Acknowledgment, Progress Report, or Reply refers.

A Reference Serial Number is represented in the protocol by the command:

REFERENCESERIAL <serialnumber> <CA>

In the absence of an explicit REFERENCESERIAL command, no Serial Number is to be assumed.

REPLY (to a previous piece of Mail)

A piece of Recorded or Unrecorded Mail whose Author(s) are Recipient(s) of a previous piece of Mail, and which replies or pertains to that same piece of Mail and bears its Serial Number, as the Reference Serial Number.

REPORT INTERVAL (for a Progress Report)

The interval between Progress Reports.

A Report Interval is represented in the Protocol by the command:

REPORTINTERVAL <interval> <CA>

In the absence of an explicit REPORTINTERVAL command, one with an argument whose value is effectively infinite is to be assumed (i.e., no Progress Reports are to be made).

REQUESTOR

The Individual on whose behalf a Mail user process connects to and interacts with a Mail server process.

A Requestor is represented in the Protocol as an Individual List of type REQUESTOR and length 1.

SERIAL NUMBER (for a piece of Mail)

A short-term identifier, assigned to a piece of Mail by the Clerk (or his system), which accompanies Acknowledgments, Progress Reports, and Replies, and is used to correlate the latter with the former. The lifetime of a Serial Number is conceptually from its assignment by the Clerk until the Delivery of the Recipient(s) Reply(s) to the Author(s) (or until their decision to send no reply).

A serial Number is represented in the Protocol by the command:

SERIAL <serialnumber> <CA>

In the absence of an explicit SERIAL command, no Serial Number is to be assumed.

SIGNATURE (for the delivery of a piece of Mail to a Recipient)

A human-readable indication of the Author(s) of a piece of Mail. The string ’Jim and Dick’ is a valid Signature.

A Signature is represented in the Protocol by the command:

SIGNATURE <signature> <CA>

STATIC ATTRIBUTES (of a piece of Recorded Mail)

Those attributes of a piece of Recorded Mail -- Content, Title, Comments, Author(s), Clerk, and Creation Date -- which are forever fixed at Recording Time, and hence can never be modified.

Static Attributes can be independently specified with commands described elsewhere, or specified collectively by reference to an existing piece of Recorded Mail. The command which follows assigns to the current piece of Mail the Static Attributes of the piece of Recorded Mail it references, and is exactly equivalent to an appropriate set of TITLE, COMMENTS, etc. commands.

LOCATION <fileaddr> <CA>

TITLE (of a piece of Mail)

A concise description of the Content of a piece of Mail.

A Title is represented in the Protocol by the command:

TITLE <title> <CA>

In the absence of an explicit TITLE command, one with a null argument is to be assumed.

UNRECORDED MAIL

Mail which is never presented to a Recording Agent for permanent storage and cataloging, but which is simply Delivered to its Recipient(s) by a Delivery Agent.

UPDATE REQUEST (to a Recording Agent for a piece of Recorded Mail)

A request made of a Recording Agent to add, replace, or delete an Individual from the Access or Distribution List for a piece of Mail; or to add or delete a Catalog from the Catalog List.

An Update Request is represented in the Protocol by the command:

UPDATETYPE <updatetype> <CA>

followed immediately in the command stream by an Access, Distribution or Catalog List.

USER

A process or human who sends and/or receives Mail.

USER VERIFICATION

The act of verifying an ID as that of a specified Individual.

USER VERIFICATION AGENT

A Mail server process which performs User Verification

MP FUNCTIONS

A MP function is the request by a Mail user process and the subsequent performance by a server, of a major task related to the management of Mail. The following functions are defined:

RECORDING
DELIVERY
DISTRIBUTION
FORWARDING
CITATION RETRIEVAL
UPDATE CITATION
USER VERIFICATION

One might expect that within the Network there would be just a few Recording Agents (who implement the Recording, Citation Retrieval, and Update Citation functions); a few Distribution Agents (who implement the Distribution function); one or two User Verification Agents (who implement the User Verification Function); and many hosts who implement the Delivery and Forwarding functions.

In general, a host is free to implement any, all, or none of the functions defined by the Protocol; and a host is free to require a login (for purposes of accounting) before permitting a user process access to any of the function(s) it has implemented.

An FTP server process who chooses to not implement MP or a particular MP function simply rejects the command that requests the unimplemented server with the reply:

400 Function not implemented.

A server who chooses to require login before allowing access to the MP subsystem or to an MP function, simply rejects the command that requests the charged-for service with the reply:

332 Login first, please.

The functions defined in MP are:

RECORDING

The Recording function is invoked with the command:

RECORD <CA>

Once this command is given, the user process shall provide the following (in any order that suits it):

(1) Any Static Attributes desired.

Content and Clerk are required. Defaults for other Static Attributes (applied by the server if the appropriate commands don't appear) are as follows:

Title or Comments as specified in the glossary.

Author to the Clerk.

Creation Date to the Recording Date.

(2) Initial values for any Dynamic Attributes desired.

Defaults (applied by the server if the appropriate commands don't appear) are as follows:

Distribution and Catalog Lists to null.

Access List as specified in the glossary.

The Recording function is terminated with either of the commands:

EXIT <CA> or ABORT <CA>

EXIT represents normal termination, and causes the server to perform the Recording function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with RECORD, is therefore a NOP.

DELIVERY

The Delivery function is invoked with the command:

DELIVER <CA>

Once this command is given, the user process shall provide the following (in any order that suits it):

(1) Any Static Attributes desired.

Content is required. Defaults for other Static Attributes (applied by the server if the appropriate commands don't appear) are as follows:

Title or Comments as specified in the glossary.

Clerk to null

Author to the Clerk.

Creation Date to the Delivery Date.

(2) Any Dynamic Attributes desired.

Distribution List is required. Defaults (applied by the server if the appropriate commands don't appear) are as follows:

Catalog List to null

Access List as specified in the glossary.

Both of these attributes are for the Recipient's information only when presented in the context of Delivery, so defaulting them to null simply implies that the Clerk doesn't desire that they be communicated to the Recipient.

(3) Any or all of the following optional parameters:

(a) Delivery Type

(b) Acknowledgment Type

The specification of this parameter is appropriate if and only if the Delivery Type is POSITIVE or NEGATIVE ACKNOWLEDGMENT or PROGRESS REPORT. In this context, Acknowledgment Type tells the server how to interpret the Content of the Acknowledgment.

(c) Serial Number

The Serial Number assigned to the piece of Mail being Delivered. This parameter is inappropriate unless the Delivery type is FORWARD (in which case the Serial Number is the one preserved from the previous Delivery), MAIL, or REPLY.

(d) Reference Serial Number

The Serial Number assigned to the piece of Mail to which the current piece of Mail is either an Acknowledgment, Progress Report, or Reply. The specification of this parameter is therefore inappropriate if the Delivery Type is MAIL.

The Delivery function is terminated with either of the commands:

EXIT <CA> or ABORT <CA>

EXIT represents normal termination, and causes the server to perform the Delivery function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with DELIVER, is therefore a NOP.

DISTRIBUTION

The Distribution function is invoked with the command:

DISTRIBUTE <CA>

Once this command is given, the user process shall provide the following (in any order that suits it):

(1) Any Static Attributes desired.

Content is required. Defaults for other Static Attributes (applied by the server if the appropriate commands don't appear) are as follows:

Title or Comments as specified in the glossary.

Clerk to null

Author to the Clerk.

Creation Date to the Delivery Date.

(2) Any Dynamic Attributes desired.

Distribution List is required. Defaults (applied by the server if the appropriate commands don't appear) are as follows:

Catalog List to null

Access List as specified in the glossary.

Both of these attributes are for the Recipient(s) information only when presented in the context of Distribution, so defaulting them to null simply implies that the Clerk doesn't desire that they be communicated to the Recipient(s).

(3) Any or all of the following optional parameters:

(a) Delivery Type

MAIL, FORWARD, or REPLY only.

(b) Serial Number

The Serial Number of the Mail being Distributed. The Distribution Agent will relay this Serial Number to each Recipient at Delivery.

(c) Reference Serial Number

The Serial Number of the piece of Mail to which the current piece of Mail is a Reply. The Distribution Agent will relay this Serial Number to each Recipient at Delivery. The specification of this parameter is appropriate if and only if the Delivery Type is REPLY.

(d) Acknowledgment Condition

An Acknowledgment is requested from the Distribution Agent if and only if the Acknowledgment Condition is other than NEVER.

(e) Acknowledgment Type

(f) Cutoff Interval

(g) Report Interval

Progress Reports are requested from the Distribution Agent if and only if this parameter is specified explicitly.

(h) Monitor

This parameter is ignored unless either an Acknowledgment or Progress Reports (or both) are requested.

The Distribution function is terminated with either of the commands:

EXIT <CA> or ABORT <CA>

EXIT represents normal termination, and causes the server to perform the Distribution function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with DISTRIBUTE, is therefore a NOP.

FORWARDING

The Forwarding function is invoked with the command:

FORWARD <CA>

Once this command is given, the user process shall provide the following (in any order that suits it):

(1) Forwardee

(2) Distribution list

This is the set of Individual(s) to whom the Mail is to be Forwarded.

The Forwarding function is terminated with either of the commands:

EXIT <CA> or ABORT <CA>

EXIT represents normal termination, and causes the server to perform the Forwarding function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with FORWARD, is therefore a NOP.

CITATION RETRIEVAL

The Citation Retrieval function is invoked with the command:

RETRIEVE <CA>

Once this command is given, the user process shall provide the following (in any order that suits it):

(1) The pathname of the piece of Mail whose Citation is to be retrieved:

PATHNAME <pathname> <CA>

(2) Any or all of the following optional parameters:

(a) Citation Template

(b) Requestor

This parameter is required if and only if Content is requested and Read Access happens not to be granted to All, in which case the server verifies that the Requestor has Read Access to the piece of Mail.

(c) FILE <CA>

This command is appropriate if and only if Content is requested. The presence of this command implies that the Content of the Mail is to be returned to the user process (following the return on the TELNET connection

of any other Citation Component(s) requested) as a file using the FTP data transfer commands (e.g., BYTE, SOCK, TYPE) currently in effect. FILE is exactly equivalent in effect to an FTP RETR command (in its use of data transfer commands, in its establishment of the data connection etc.) except that no pathname is required.

In the absence of a FILE command, Content is returned on the TELNET connection like any other Citation Component.

The server returns the Citation Components in the order requested by the user process (except that Content, if requested as a file, is always returned after the 'end of citation' indication), each as a reply whose numeric code is 172 and whose text is exactly the command normally used to communicate that same parameter to the server. A reply whose numeric code is 173 terminates the reply list.

Title and Content, which (in general) may each contain the TELNET New Line sequence (CR LF), are represented as continued replies, using the FTP reply continuation convention (see the FTP protocol document). The first four characters of each reply line except the first and last are blanks inserted by the server which must be deleted by the user process to correctly recover the value of the Title or Content.

The Citation Retrieval function is terminated with either of the commands:

EXIT <CA> or ABORT <CA>

EXIT represents normal termination, and causes the server to perform the Citation Retrieval function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with RETRIEVE, is therefore a NOP.

UPDATE CITATION

The Update Citation function is invoked with the command:

UPDATE <CA>

Once this command is given, the user process shall provide the following (in any order that suits it):

(1) Requestor

This parameter is required unless Controlling Access has been granted to All, in which case it is treated as a NOP if given. The server verifies that the Requestor has Controlling Access to the piece of Mail.

(2) One or more Update Requests

The Update Citation function is terminated with either of the commands:

EXIT <CA> or ABORT <CA>

EXIT represents normal termination, and causes the server to perform the Update Citation function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with UPDATE, is therefore a NOP.

USER VERIFICATION

The User Verification function is invoked with the command:

VERIFY <CA>

Once this command is given, the user process shall specify any number of Requestors.

The server prompts for the Id for each, the user process provides it, and the server returns a reply whose numeric code is 272 is the Id is correct or 472 otherwise.

The User Verification function is terminated with either of the commands:

EXIT <CA> or ABORT <CA>

EXAMPLE

In the example below, a short message is recorded for public access, and distributed to a single recipient. The user process is assumed already connected to the server.

Note: This would be the implementation of NIC Journal Submission, where the NIC is understood to be both a Recording and Distribution Agent.

Replies from the server are in brackets.

MAIL <CA>

The Mail system is invoked.
[261 RE DE DI FW CI UP UV -- supported.]

REC <CA>

The Recording function is invoked.
[200 OK.]

TITL SMFS Runs on TENEX 1.31 at the NIC <CA>

A Title is given
[200 OK.]

TEXT The NIC came up on TENEX 1.31 on 1-APR. <CRLF> I tried SMFS here on the new monitor and it <CRLF> works fine. I don't understand why I had <CRLF> problems running your copy of the code at <CRLF> BBN-TENEX. Are you still unable to reference <CRLF> the same archived file from two different <CRLF> TENEXs? <CA2>

The Content of the message is entered.
[200 OK.]

CLER WHITE@SRI-ARC <CR>

The Clerk is identified as White at SRI-ARC.
[330 OK. Now Id, please]

ID id <CA>

His Id is supplied.
[200 OK.]

EXIT <CA>

Exit from the Recording function is effected, and the pathname '15490' is returned by the Recording Agent for the now Recorded Mail.
[270 15490 -- is assigned as the pathname.]

DIST <CA>

The Distribution function is invoked.
[200 OK.]

LOC SRI-ARC 15490 <CA>

The message just recorded is specified for Distribution.
[200 OK.]

RECI * DHC <CA>

The Recipient is specified via NIC Ident to be Dave Crocker at UCLA-NMC.
[200 OK.]

GREE Dave <CA>

A Greeting is given.
[200 OK.]

DISP R

A reply is requested.
[200 OK.]

SIGN Jim

The message is signed.
[200 OK.]

ACKC A <CA>

Acknowledgment of the Mail's Delivery is requested whether Delivery succeeds or fails..
[200 OK.]

ACKT T <CA>

The Acknowledgment is to be terse.
[200 OK.]

CUT 1 D <CA>

If Delivery hasn't been effected within 24 hours, the attempt is to be abandoned (and an Acknowledgment of failure returned). The Monitor (to whom the Acknowledgment is sent) is allowed to default to the Clerk.
[200 OK.]

SERI serial <CA>

A Serial Number is assigned for purposes of coordinating Acknowledgment and Reply. A desirable implementation of the sender's user and server processes is one in which the Serial Number is assigned by the user process, rather than by the human user himself in such a way that his server process can automatically make the association between original Mail, and subsequent Acknowledgment and Reply.
[200 OK.]

EXIT <CA>

Exit from the Distribution function is effected.
[200 OK.]

EXIT <CA>

Exit from the Mail subsystem is effected.
[200 OK.]

COMMAND SUMMARY

Every command requires at least one reply from the server.

THOSE SPECIFIC TO MP

ABORT <CA>
ACCESS <individual> <CA>
ACCESSTYPES <accesstypes> <CA>
ACKCONDITION <ackcondition> <CA>
ACKTYPE <acktype> <CA>
AUTHOR <individual> <CA>
CATALOG <catalog> <CA>
CITATIONTEMPLATE <citationtemp> <CA>
CLERK <individual> <CA>
COMMENTS <comments> <CA>
CREATIONDATE <datetime> <CA>
CUTOFF <interval> <CA>
DELIVER <CA>
DELIVERYTYPE <deliverytype> <CA>
DISPOSITION <disposition> <CA>
DISTRIBUTE <CA>
EXIT <CA>
FILE <CA>
FORWARD <CA>
FORWARDEE <individual> <CA>
GENERALDELIVERY <CA>
GREETING <greeting> <CA>
ID <id> <CA>
LOCATION <fileaddr> <CA>
MAIL <CA>
MONITOR <individual> <CA>
PATHNAME <pathname> <CA>
RECIPIENT <individual> <CA>
RECORD <CA>
REFERENCESERIAL <serialnumber> <CA>
REPORTINTERVAL <interval> <CA>
REQUESTOR <individual> <CA>
RETRIEVE <CA>
SERIAL <serialnumber> <CA>
SIGNATURE <signature> <CA>
TEXT <string> <CA2>
TITLE <title> <CA>
UPDATE <CA>
UPDATETYPE <updatetype> <CA>
VERIFY <CA>

THOSE BORROWED FROM FTP

The following commands borrowed from FTP are defined (also) as MP commands to support the transfer of the Content of a piece of Mail in 'file' form. The reader is referred to the FTP protocol document for a description of their use and syntax. The borrowed commands are:

BYTE, SOCK, PASV, TYPE, STRU, MODE, REST, and SITE.

The following commands borrowed from FTP are defined (also) as MP commands to permit changes of accounting parameters within the MP subsystem. The accounting parameters in force when the subsystem is entered apply (if necessary) within the subsystem until changed. Values to which the parameters may have been changed while in the subsystem continue in effect upon return to the FTP command space. The borrowed commands are:

USER, PASS, and ACCT.

The following miscellaneous commands borrowed from FTP are defined also as MP commands:

HELP and NOOP.

COMMAND REPLIES

This list is undoubtedly incomplete; some crucial reply code assignments may be missing despite the author's attempt to foresee the kinds of interaction that might arise between user and server and the responses from the server that they would require.

172 <A Citation Component>

In response to the EXIT command which terminates the Citation Retrieval function.

173 End of citation.

Following a list of 172 replies.

200 OK.

This is the standard, positive acknowledgment used throughout the Protocol.

270 <pathname> -- is assigned as the pathname.

In response to the EXIT command which terminates the Record function.

271 <functionlist> -- supported.

In response to the MAIL command by which the user process gains entry to the Mail subsystem. This response is mandatory, and from it the user process can quickly determine what function(s) are supported by the server.

272 Requestor is who he says he is.

In response to an ID command in the User Verification function. This reply informs the user process that the Id given is in fact that of the Individual specified.

330 OK. Now Id, please.

In response to the first command in each pair of commands in an Individual List. This reply requires the next command from the user process to be ID.

332 Login first, please.

In response to any command which invokes a Mail function (e.g., RECORD, DISTRIBUTE, DELIVER), or to the MAIL command itself. This reply implies that the requested function is supported by the server, but that the user is required to login before invoking it.

400 Function not implemented.

In response to any command which invokes a Mail function (e.g., RECORD, DISTRIBUTE, DELIVER), or to the MAIL command itself. This reply implies that the requested function is not supported by the server.

431 Incorrect Id.

In response to the ID command in an Individual List command pair. This reply implies that the Id specified was incorrect.

440 <Error relayed from Recording Agent>

In response to the LOCATION command. This reply implies that the server attempted to retrieve the specified piece of Mail from an FTP server but failed because it returned the error reply whose text is duplicated in the current reply.

470 No such pathname.

In response to the PATHNAME command (in the Citation Retrieval function). This reply implies that the specified pathname is not recognized by the server.

471 No unRead Mail to Forward.

In response to the EXIT command which terminates the Forwarding Function.

472 Requestor is NOT who he says he is.

In response to an ID command in the User Verification function. This reply informs the user process that the Id given is NOT that of the Individual specified.

473 You don’t have Read Access to the Mail.

In response to the LOCATION command, or to the PATHNAME command in a Citation Retrieval function. This reply implies that the Requestor doesn’t have Read Access to the piece of Mail.

474 Recipient unrecognized; is General Delivery OK?

In response to an instance of the RECIPIENT command in a Distribution List (in the context of the Delivery function). This response implies that the Recipient in unrecognized, but that the server will attempt General Delivery to him if the user process responds with a GENERALDELIVERY command; otherwise the Recipient is rejected.

475 That Individual is not at this host.

570 No such NIC Ident or Mailbox Name.

In response to any command in which a NIC Ident or Mailbox Name appears as an argument. This reply implies that the Individual specified does not exist.

571 Invalid host.

In response to any command in which a host address or standard host name appears as an argument. This reply implies that no such host exists.

572 No such catalog.

In response to the CATALOG command. This reply implies that no such Catalog exists.

Any '500' reply.

Any of the error replies associated with FTP RETR/STOR commands.

FORMAL SYNTAX

The terse keyword forms to be employed in actually implementing a Mail user or server process are generated by deleting character(s) from the corresponding verbose forms. Those deleted characters are included but enclosed in brackets throughout the description which follows. Spaces can be used freely between terminal elements of the syntax, and in some cases, at least one space must separate two elements whose boundary could not otherwise be distinguished.

```
<CA2>            ::= TELNET Go Ahead character
<CA>             ::= TELNET new line (CR LF)
<CRLF>           ::= CR LF
<accesstypes>    ::= <readaccess> <controlaccess>
<ackcondition>   ::= A[LWAYS] | F[AILURE] | N[EVER]
<acktype>        ::= T[ERSE] | V[ERBOSE]
<action>         ::= A[CTION] | null
<catalog>        ::= <string>
<citationcomp>   ::= D[ISTRIBUTION]L[IST] | A[CESS]L[IST] |
                     C[ATALOG]L[IST] | C[ON]T[ENT] |  T[ITLE] |
                     C[OM]M[ENTS] | AU[THOR] | CL[ERK] |
                     C[REATION]D[ATE]
<citationtemp>   ::= <citationcomp> | <citationcomp>
                     <citationtemp>
<command>        ::= <shortbody> <CA> | <longbody> <CA2>
<comments>       ::= <string>
<controlaccess>  ::= C[ONTROLLING] | null
<count>          ::= decimal integer
<date>           ::= <dayofmonth> / <month> / <year>
<datetime>       ::= <date> <time>
<dayofmonth>     ::= decimal integer, 1-31
<days>           ::= <count> D[AYS]
<deliverystatus> ::= F[AILED] | S[UCCESSFUL] | T[IMED OUT] |
```

```
                        W[AITING] | U[NATTEMPTED]
   <deliverytype>   ::= F[ORWARD] | M[AIL] | N[EGATIVE
                        ACKNOWLEDGMENT] | P[OSITIVE
                        ACKNOWLEDGMENT] | P[ROGRESS]R[EPORT]
                        | R[EPLY]
   <disposition>    ::= <rsvp> <action> <interrupt>
   <fileaddr>       ::= <host> <pathname>
   <functionlist>   ::= <functiontype> | <functiontype>
                        <functionlist>
   <functiontype>   ::= RE[CORDING] | DE[LIVERY] | DI[STRIBUTION] |
                        F[OR]W[ARDING] | CI[TATION RETEiEVAL] |
                        UP[DATE] | U[SER]V[ERIFICATION]
   <globalname>     ::= * <nicident>
   <greeting>       ::= <string>
   <host>           ::= <hostname> | <hostaddress>
   <hostaddress>    ::= decimal integer, 0-255
   <hostname>       ::= standard host name
   <hour>           ::= decimal integer, 0-23
   <hours>          ::= <count> H[OURS]
   <individual>     ::= <localname> | <globalname>
   <interrupt>      ::= I[NTERRUPT] | null
   <interval>       ::= <days> | <hours> | <days> <hours>
   <localname>      ::= <mailbox> @ <host> | <mailbox> @
      NOTE: Host defaults to that of the server
   <longbody>       ::= COM[MENTS] <comments> |
                        TEXT <string>
   <mailbox>        ::= <string>
   <minute>         ::= decimal integer, 0-59
   <month>          ::= decimal integer, 1-12
   <nicident>       ::= <string>
   <id>             ::= <string>
   <pathname>       ::= <string>
   <readaccess>     ::= R[EAD] | null
   <rsvp>           ::= R[SVP] | null
   <serialnumber>   ::= <string>
   <shortbody>      ::= ABOR[T] |
                        ACC[ESS] <individual> |
                        ACKC[ONDITION] <ackcondition> |
                        ACKT[YPE] <acktype> |
                        AC[CESS]TY[PES] <accesstypes> |
                        AUTH[OR] <individual> |
                        CAT[ALOG] <catalog> |
                        CLER[K] <individual> |
                        CR[EATION]DA[TE] <datetime> |
                        CUT[OFF] <interval> |
                        C[ITATION]TEM[PLATE] <citationtemp> |
                        DELI[VER] |
                        DE[LIVERY]TY[PE] <delivverytype> |
```

```
                        DISP[OSITION] <disposition> |
                        DIST[RIBUTE] |
                        EXIT |
                        FILE |
                        FOR[WARDE]E <individual> |
                        FOR[WARD] |
                        GEN[ERAL]D[ELIVERY] |
                        GREE[TING] <greeting> |
                        ID <ID> |
                        LOC[ATION] <fileaddr> |
                        MAIL |
                        MON[ITOR] <individual> |
                        PATH[NAME] <pathname> |
                        RECI[PIENT] <individual> |
                        REC[ORD] |
                        REQ[UESTO]R <individual> |
                        R[EFERENCE]SER[IAL] <serialnumber> |
                        R[EPORT]INT[ERVAL] <interval> |
                        SERI[AL] <serialnumber> |
                        SIGN[ATURE] <signature> |
                        TITL[E] <title> |
                        UPDA[TE] |
                        UP[DATE]TY[PE] <updatetype> |
                        VER[IFY]
   <signature>      ::= <string>
   <string>         ::= any non-zero number of visible characters
                        (in particular, CA and CA2 are excluded)
   <time>           ::= <hour> : <minute> <timezone>
   <timezone>       ::= EST | EDT | CST | CDT | MST | MDT | PST |
                        PDT | GMT
   <title>          ::= <string>
   <updatetype>     ::= A[DD] | R[EPLACE] | D[ELETE]
   <year>           ::= full year in decimal (e.g., 1973)
```