PRODUCT OVERVIEW

January 2001

internet exchange

Messaging Server

INTERNATIONAL MESSAGING ASSOCIATES

IMA

Table of Contents

Introduction

Internet Exchange Messaging Server

....The Complete E-mail Solution

Electronic mail is the most widely used Internet technology today. As the corporate world becomes more and more competitive, many business organizations are increasingly turning toward electronic messaging as a primary tool for gathering and disseminating information that will help enhance their competitiveness and improve profit margins. This trend underscores the need for e-mail systems capable of communicating rapidly and reliably over the Internet and Intranets.

Internet Exchange Messaging Server from IMA (International Messaging Associates) is a complete, standalone, open architecture messaging system specifically tailored to enable legacy e-mail systems to co-exist with proprietary messaging systems while making full use of Internet messaging and directory services.

Internet Exchange is made up of a number of easy-to-configure components. Each component has specific functions that provide fast and reliable message delivery to and from the Internet.

The following is a list of the components included in Internet Exchange:

- BSMTP (Batch Simple Mail Transfer Protocol)
- cc:Mail Connector (optional)
- Directory Server
- DL (Distribution List) Manager
- Lotus Notes Connector (optional)
- Message Store
- Migration Tools
- MTA (Message Transfer Agent)
- MTA Preprocessor Unit
- SMTP (Simple Mail Transfer Protocol)

This document details all the above-mentioned components.

Batch Simple Mail Transfer Protocol

While most e-mail is directly transported between systems via the SMTP (Simple Mail Transfer Protocol), there are times when other forms of message transport are more desirable. The following are some situations wherein an alternative transport method is useful:

- A dedicated IP (Internet Protocol) address is not available
- Low message volume
- Higher costs associated with a dedicated Internet connection
- Internet connection is available only on a dial-up basis

The Internet Exchange Messaging Server comes with a standard BSMTP (Batch Simple Mail Transfer Protocol) Tunnel Encoder and Decoder, which allows the tunneling of messages across non-SMTP transports, like POP3. The BSMTP Encoder encapsulates messages into BSMTP format before delivering the messages to its destination address. This destination address can be within Internet Exchange or any other server with an RFC-2442-compliant encoder installed. The BSMTP Decoder works with POP3 (Post Office Protocol version 3) client module in picking up remote messages and then de-tunnels the messages by re-injecting them into the Internet Exchange Input Queue.

System Architecture

Internet Exchange's BSMTP Decoder uses POP3 in retrieving tunneled messages. After de-tunneling, the messages are decoded. The BSMTP module then routes the decoded messages into the Internet Exchange Input Queue as though they were received directly via SMTP. Once the messages enter the messaging server, they are handled just like ordinary messages with the appropriate pre-processing and message routing taking place within the MTA.

Once the MTA receives a message addressed to a recipient on the other end of a BSMTP tunnel, it routes the message to the BSMTP Tunnel Encoder. The BSMTP Tunnel Encoder then encapsulates the message together with its envelope information into a new message. The new message is then sent to the MTA Input Queue for further routing and delivery to the BSMTP Decoder, which decodes and then delivers the messages to its intended recipients.

The BSMTP Tunnel is an ideal solution for sites that prefer to use POP3 as their message retrieval protocol rather than SMTP. Unlike other systems that try to recover message envelope information out of message headers, resulting in improper routing of messages, Internet Exchange's BSMTP Tunnel implementation preserves the message envelope while passing through non-SMTP transports, like POP3. This ensures proper message delivery to the final intended recipients.

Why BSMTP?

When a user composes a message, he uses an e-mail client, such as Microsoft Outlook Express, Netscape Communicator, among others. The e-mail client presents an interface which prompts the user to provide addressing information (`To:`, `Cc:` and `Bcc:`) and a `Subject:` field as minimum requirements. After supplying the addressing and labeling information for the message, the user then composes a message.

Once the user has completed composing his message, an instruction is given to the e-mail client to send the message (see **Figure 1A** below). At this point, the mail client constructs a message file and an envelope. The message file contains what is known as the message header, consisting of the original `To:`, `Cc:` and `Subject:` fields (but not the `Bcc:` information), as well as the return address of the sender and the time (`Date:` field) when the message was composed. After the message header, the message file then contains the message body as composed by the sender. The envelope is now also constructed. It initially contains the list of recipients specified by the sender in the `To:`, `Cc:` and `Bcc:` fields of the original message.

After the construction of the message file and envelope, these are sent separately to a local MTA, which transfers the message across the Internet to the intended recipients. It is important to note that with the exception of trace information recorded in the message header, the message file remains unchanged during transit. In particular, the addresses present in the `To:` and `Cc:` fields of the message remain as they were when the user first composed the message. However, for the message envelope, the list of recipient addresses are changed during transit because address expansions are done when mail aliases, user forwarding, or distribution lists are encountered.

When the message reaches its final recipient destination, the last MTA in the chain sends the message to the LMDA (Local Mail Delivery Agent) which then deposits the message file in the appropriate repository or mailbox. Now that final delivery has been performed, the envelope information is discarded. By this time, the message viewed by the recipient indicates the original recipient addresses as specified in the `To:` and `Cc:` fields.

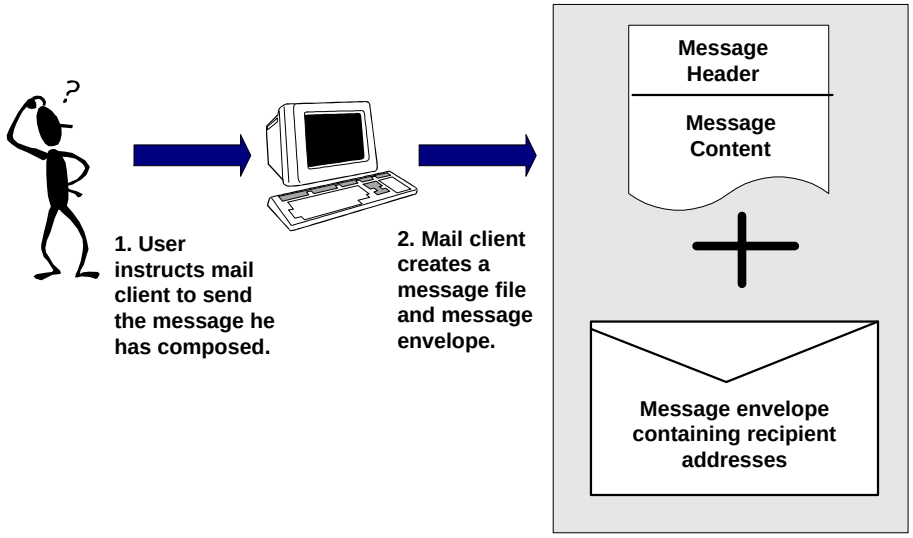

Figure 1A: Message Processing

The separation of the original message file from the envelope information is the basic principle behind the transfer of e-mail messages on the Internet. This allows not only the proper routing and re-routing of messages from one system to another. It also preserves the original labeling of messages as composed by the sender.

So, why is all of this important in the discussion of BSMTP and the use of mail repository, like POP3 accounts, as message transport holding areas? The answer lies in the fact that when messages are delivered to a POP3 account, the envelope information is usually lost. When this message is retrieved for later delivery, the recipient information has to be derived from the message header present in the message file since the envelope has already been discarded. If the POP3 repository is used as a holding area for more than one recipient, such as an entire organization or a group of people, it will be impossible to guarantee the accuracy of the regenerated envelope. The message recipients that were Bcc: in the original message will not show up in the message header, nor will any changes in the envelope that result from alias expansion; user forwarding or distribution list expansion which occurred after the original message was sent. There is simply no reliable way to re-construct the list of recipients to whom the message was intended once the envelope information has been discarded. The solution to this problem lies in the creation of an e-mail tunnel, where messages can be sent across non-SMTP transports and then re-injected (de-tunneled) into the message transport system on the other side. This is precisely what the Internet Exchange BSMTP Tunnel modules provide. The diagram in **Figure 1B** below (borrowed from RFC-2442 which describes the BSMTP Media Type) describes in detail how the BSMTP Tunnel works:

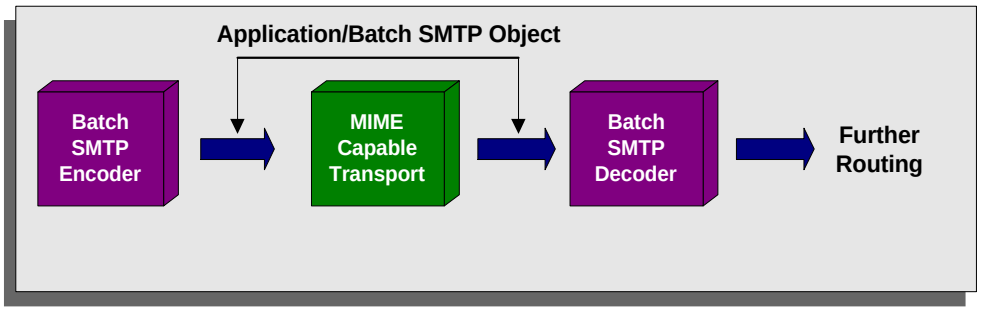

Figure 1B: Tunneling of messages

When a message arrives at the BSMTP Encoder, both the message file and the envelope information is encapsulated into a new message. This new message is addressed to a remote BSMTP Decoder, which de-tunnels the encapsulated message and then routes the message.

In the case of Internet Exchange, the tunneled message is delivered to the MTA Input Queue where it is routed to any BSMTP Decoder with an e-mail address accessible either on the Internet or any other Internet Exchange local channels.

When a BSMTP tunneled message is delivered to a message repository, like a POP3 account, even though the accompanying envelope is discarded, the original envelope remains intact within the tunneled message. The Internet Exchange BSMTP Decoder first retrieves messages of this type via POP3, before de-tunneling and submitting the original message file and the original message envelope to the MTA Input Queue for further routing.

Unique ID Listing

The BSMTP module maintains a database of UIDL (Unique ID Listing). Its function is to synchronize the downloaded messages between the POP3 client and the server. If the message with a UIDL has already been downloaded in the previous session, the BSMTP module will issue a POP3 command to delete the message and starts to download the rest of the messages.

According to RFC-1939, the UIDL command is intended to function as a unique ID (identification) of a message. This enables the POP3 client to handle situations where two identical copies of a message in a maildrop have the same unique ID.

Conclusion

The Internet Exchange Messaging Server comes with a standard BSMTP Tunnel Encoder and Decoder, which allows the tunneling of messages across non-SMTP transports, like POP3. The BSMTP Encoder encapsulates messages into BSMTP format before delivering the messages to its destination address. This destination address can be within Internet Exchange or any other server with an RFC-2442-compliant encoder installed. When these messages arrive at a single POP3 account, they can later be picked up by the BSMTP Decoder module, which decodes the resulting messages and then submits them to the MTA for further routing, with the original envelope recipients retained. The BSMTP Decoder works with the POP3 client module in picking up remote messages and then de-tunnels the messages by re-injecting them into the Internet Exchange Input Queue.

cc:Mail Connector

The cc:Mail Connector is a plug-in module that connects cc:Mail environment to the Internet (see **Figure 2A** on page 6). Using this module, you can send and receive messages to and from the Internet, communicate with other local channels, and provide a rich migration path for moving cc:Mail users to open Internet messaging standards. As a plug-in module, the cc:Mail connector features the following:

- Anti-virus support
- Anti-spam support
- BSMTP
- LDAP (Lightweight Directory Access Protocol)-enabled Directory Server
- ESMTP (Extended Simple Mail Transfer Protocol) support
- Unlimited scalability

Message Flow

Internet Exchange to cc:Mail Environment

Messages received by Internet Exchange destined for the cc:Mail environment are fetched by the CCIN channel, which obtains the proper routing information of the message within the cc:Mail environment. The CCIN channel processor determines the routing information of the message by first translating the recipient address, for example, “jdoe@domain.com” to its equivalent cc:Mail username, which is “John Doe”. After translating the recipient address, CCIN performs directory lookup to determine the receive permission of “John Doe”. After determing the receive permission, CCIN converts “jdoe@domain.com” to “john doe” and uses this username in performing an address book lookup to the cc:Mail Post Office. If CCIN finds that username “john doe” is included in the cc:Mail address book, CCIN forwards the message to the cc:Mail Post Office. Then, the cc:Mail Post Office delivers the message to the user’s mailbox “john doe”.

cc:Mail Environment to Internet Exchange

A message coming from the cc:Mail environment destined for Internet Exchange is processed by the CCOUT channel. CCOUT retrieves the message from the cc:Mail Internet Post Office queue, which obtains the proper routing information of the message. For example, a registered cc:Mail user named “John Doe” sent a message to an Internet Exchange local user “sample@ima.com”, the CCOUT channel processor translates the sender’s cc:Mail username “John Doe” to its equivalent Internet address by performing a directory lookup in the Directory Server. If the sender’s address does not exist in the Directory Server, CCOUT translates the sender’s address by using the Default Mapping method. This method translates the sender’s address in either of the following format:

Firstname_Lastname@domain.com **or** Firstname.Lastname.domain.com.

After translating the sender’s address, CCOUT exports the message from the Internet Post Office Queue to Internet Exchange, which routes the message to its intended recipient “sample@ima.com”.

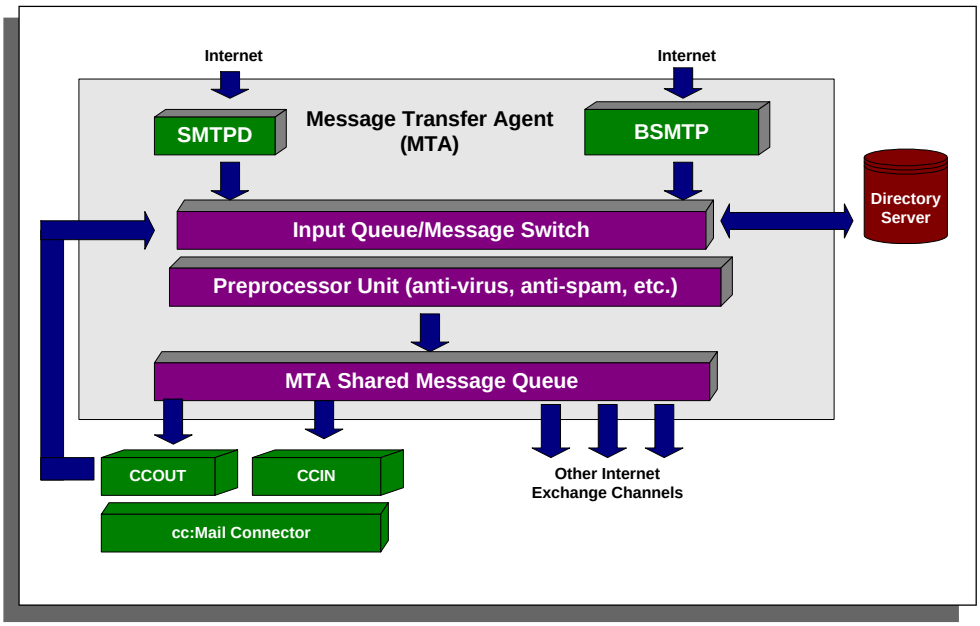

Figure 2A: Connecting cc:Mail environment to the Internet

Migration Strategy

Objectives of the Migration Process

As more and more organizations move from legacy or LAN-based mail systems to non-proprietary, Internet-based messaging systems, it becomes apparent that system administrators need cost-effective tools that:

- provides total control over the entire migration process
- supports non-destructive migration process
- supports automated and batch processes for migrating large user groups
- provides an audit trail of the migration process for tracking which information/users have been moved to the new system

Internet Exchange Messaging Server supports an extensive migration process for moving existing cc:Mail users to non-proprietary, Internet-based messaging systems. The major goals of the migration process are:

- to migrate the cc:Mail user database (address book) to the Internet Exchange Directory Server

- to migrate the cc:Mail user mailbox (including folders) to the Internet Exchange Local Mailbox Storage, which is used by the IMAP4 (Internet Mail Access Protocol version 4) and POP3 servers

Migration Tools

The migration tools included in the software are 32-bit Windows applications which can be run via a GUI (Graphical User Interface) or in batch processing mode. These tools provide logging facilities to:

- track all user addresses that have been exported
- track all user mailboxes that have been migrated
- log all the errors encountered during the migration process

The migration tools uses two standalone programs: a directory and mailbox conversion tool. The directory conversion tool translates the cc:Mail directory to the Internet Exchange Directory format. It uses pre-defined rules to create a valid RFC-822 Internet address for the user being migrated. To support this process, the migration tools allow cc:Mail Post Offices to be mapped to different Internet domains. The mailbox conversion tool converts cc:Mail mailboxes into a format supported by the Internet Exchange Message Store, where users can access the messages using IMAP4 and/or POP3 clients or the Web Mail Client.

Directory Synchronization

The Internet Exchange architecture is specifically designed to support disparate messaging systems (i.e., cc:Mail, Lotus Notes), each having its own local directory for storing user records. To keep all directories in the system synchronized, the Directory Server comes with a replication engine that replicates user records between the cc:Mail and Lotus Notes directories and the Directory Server.

Conclusion

cc:Mail is one of the most widely used e-mail platforms worldwide. The cc:Mail connector provides a tool for connecting cc:Mail environment to the Internet. Using this connector, cc:Mail users can send and receive messages to and from the Internet while reaping all the benefits offered by the Internet Exchange Messaging Server, such as virus scanning, spam control, BSMTP support and Directory Server. In addition, the cc:Mail connector supports cost-effective migration tools for moving users to other messaging systems based on Internet standards. Migrating users from one e-mail system to another can be very tasking on the part of system administrators, if not done with the right migration tools. Moving users from legacy systems to non-proprietary messaging systems based on open Internet standards is very complicated and involves many factors that could spell the success or failure of the migration process. With the migration tools, you can define a trouble-free migration path that is capable of handling all the complexities involved in the migration process.

Directory Server

The Directory Server, which is based on a client-server architecture, is used to search for a person's or an organization's e-mail address or other related information stored in the database. The Directory Server is well integrated with all of Internet Exchange's components, such as the IMAP4 Server, POP3 Server, Message Switch, DL Manager, and LMDA. Once Internet Exchange receives messages, the Message Switch routes the messages to the appropriate channels: SMTPC, Message Store, DL, cc:Mail/Notes connectors, among others, and performs pre-set processing on each message, during which, the message routing is facilitated by the Directory Server.

Internet Exchange provides a separate web interface for the system administrator and end users in configuring the Directory Server. The system administrator can configure the module via the "System Administrator" web interface while the end users are provided with the "End User" web interface.

The system administrator can perform the following functions:

- add new entries
- search for a particular entry
- delete/modify existing entries

The end users can perform the following functions:

- view existing entries
- edit/modify existing entries

The Directory Server also allows the client to issue multiple requests concurrently. If the Directory Server searches the directory and finds multiple matching entries, each entry will be sent to the client. The Directory Server also provides an "authentication" service, restricting access to sensitive information, such as passwords and confidential user profiles. Operations are provided for adding and deleting an entry from the directory, modifying an existing entry and searching for a particular entry. The search operation allows some portion of the directory to be searched for entries that match the criteria specified by the search filter. Information can be requested from each entry that matches the criteria.

Components

The Internet Exchange Directory Server consists of two major subsystems: the *front-end protocol engine* and the *back-end database engine* (see **Figure 3A** on page 9). The front-end protocol engine receives requests from the client and processes these requests by invoking read-and-write functions in the back-end database engine. Among the operations performed by the *front-end protocol engine* are bind, unbind, search, modify, modify RDN (Relative Distinguished Name), delete and abandon operations. The *back-end database engine* searches for information in the directory and modifies it based on commands from the protocol engine. It communicates with

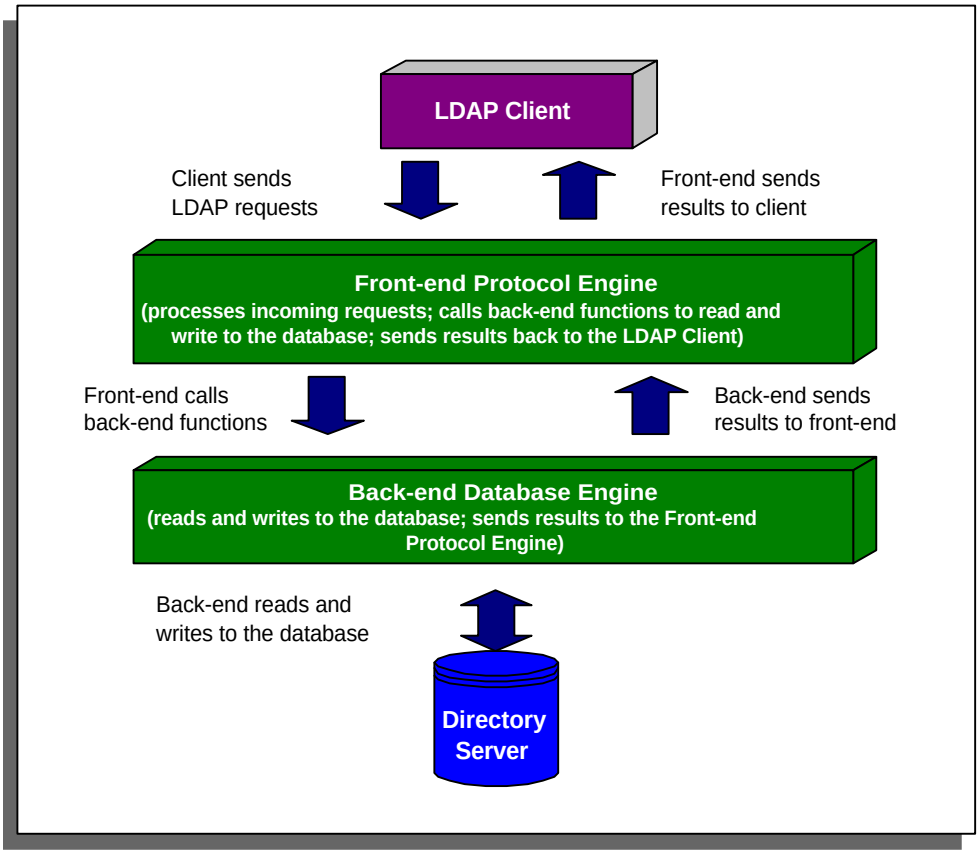

Figure 3A: Internet Exchange Directory Server Components

the front-end engine through a well-defined API (Application Programming Interface).

Directory Information Tree

Directory entries in the Directory Server are organized using a DIT (Directory Information Tree). The root of the DIT is represented by a special entry whose DN (Distinguished Name) is called the directory suffix. The server uses the Internet domain name that is associated with a particular company as the directory suffix for that company (see **Figure 3B** on page 10). The directory suffix should have an attribute of “organization”. For example, if the Internet domain name for IMA is “ima.net”, the directory suffix for the company’s DIT will be “o=ima.net”. The DIT can be divided into sub-domains that correspond to the various departments/divisions in the organization. Therefore, IMA’s Engineering, Sales and Support Departments can be represented in the DIT as ”ou=engr.ima.net”, “ou=sales.ima.net” and ““ou=support.ima.net” where *ou* stands for *organizational unit*. User entries are organized under these nodes.

Directory Data Storage

Internet Exchange provides a default directory schema for e-mail applications. The directory data includes user account information, group information and mail routing information. The user account information consists of the unique user id (e-mail address), user password, e-mail address and other user-related information. The group information consists of data about the users that have the same access right to the same directory. Access to sensitive information, such as password and confidential user profiles, is restricted by an authentication mechanism.

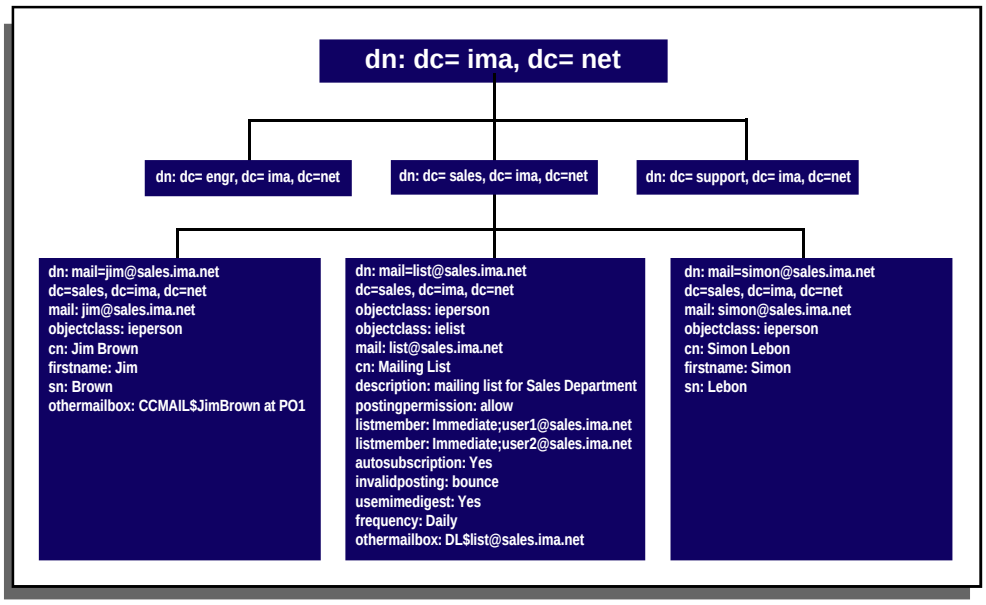

Figure 3B: Directory Information Tree

Naming Style

The Internet Exchange Directory Server utilizes the standard naming style to organize the directory data. The standard naming style, which is based on the name hierarchy of the existing DNS (Domain Name System) infrastructure, recommends that the directory structure be based on the domain component of a user's e-mail address. The DNS provides mapping between textual names (e.g. ima.net) and the IP address (e.g. 123.456.7.8) that the Internet uses behind the scene.

Directory entries in the Directory Server are organized using DIT. The root of the DIT is represented by a special entry whose DN is called the directory suffix. The upper portions of a directory tree is constructed using the registered DNS names in combination with the "mail" and "dc" attributes to define the DN of each registered user. A domain name like "ima.net" is constructed as "dc=ima, dc=net". All of the other sub-domains under "ima.net" are named under this directory tree (e.g. dc=sales, dc=ima, dc=net). They represent the directory tree for the domain "sales.ima.net".

In the example given in **Figure 3C** on page 11, the record entry is represented by an object class attribute with value "ieperson". The first entry specifies the DN "mail=peterchan@ima.net, dc=ima, dc=net". The Internet mail address is "peterchan@ima.net" as defined by the attribute "mail". The CN (Common Name) is "Peter Chan" which is composed of the values "sn" (surname) and "firstname" attributes.

Internet Exchange uses the attribute "othermailbox" to define the mail routing for the different channels, such as CCMAIL, NOTES, LOCAL, SMTPC and BSMTP. The format used in specifying the value of the "othermailbox" is composed of a string identifying the channel name, followed by a "$" character and then the address defined in the corresponding channel. For example, the value "LOCAL$peter@ima.net" specifies a LOCAL user (using the account in the Internet Exchange Message Store) with an account name "peter@ima.net". Multiple values can also be defined such that all incoming messages will be routed to multiple channels.

Example

objectclass: ieperson
dn: mail=peterchan@ima.net, dc=ima, dc=net
mail: peterchan@ima.net
cn: Peter Chan
firstname: Peter
streetaddress: 107 Alfaro Street, Makati City
telephonenumber: 1234567
x-permission: SMTPC$Send/Receive
sn: Chan
othermailbox: SMTPC$peterchan@otherisp.net

Figure 3C: Example of a directory entry

The character "$" is used to separate the channel name and the mail address in the channel. The Message Switch uses this information in determining the channel where the message should be routed to.

Directory Schema

The Directory Schema is a set of rules that defines how the data is stored in the directory and how the client/server program should handle the information during directory operations. It also reduces the duplication of data and provides a well-documented, predictable way for directory-enabled applications to access and modify the collection of directory objects. Before the directory server can modify or store a new entry, the directory first checks the entry's contents against the schema rules. Whenever the client or server program compares two attribute values, it will consult the defined schema to determine the appropriate comparison algorithm to use.

The Directory Schema consists of an attribute type, attribute syntax and object class. Each attribute has a type and one or more values. The *attribute type* describes the kind of information contained in the attribute, and the value contains the actual data. For example, for the attribute "cn", a possible value is "Peter Chan". The *attribute syntax* describes the types of data that may be placed in the attribute values of that type. It also defines how the directory compares values when searching. The Internet Exchange Directory Server uses the "caseIgnoreString" syntax. This means that the case is not significant when searching or comparing values. Hence, when searching for a user named "Peter Chan", you may type in either "peter chan" or "PETER CHAN". The values "peter chan", "PETER CHAN" are equivalent to "Peter Chan". The *object classes* are used to group related information. Each directory entry belongs to one or more object classes. The names of the object classes to which an entry belongs to are listed as values for a special multi-valued attribute called *object class*. It determines which attributes must be included in the entry. Internet Exchange defines several object classes namely: *ieperson*, *ieconfig*, *iemessagestoreuser*, *iemessagestoreshared*, *ielist*, *ielistowner*, *ielistrequest* and *IEMachine*.

Conclusion

Internet Exchange Directory Server allows the system administrator to manipulate stored information in the directory. It allows the system administrator to add new entries, search for a particular entry, and delete/modify existing entries. End users, meanwhile, are allowed to view existing entries and edit/modify existing entries.

Other Internet Exchange modules, such as the IMAP4 Server, POP3 Server, Message Switch and LMDA, access the Directory Server for directory information. The IMAP4 and POP3 servers connect to the Directory Server to request for user account information which is used in validating the authenticity of users who are trying to log on into the system. The LMDA, meanwhile, requests for user information that may be used for local mail delivery.

Distribution List Manager

The DL (Distribution List) Manager allows messages to be sent to all lists subscribers by simply submitting the said messages to a single address. It enables the system administrator to create electronic mailing lists that supports the following features: mail blocking, automatic mailing list subscription/unsubscription, and setting the preferred delivery options.

The DL Manager provides the system administrator with option to accept or reject subscribers to the mailing list. To subscribe to a mailing list, the potential subscriber must submit a subscription request to the system administrator/list owner via e-mail or web-based interface. Before submitting the subscription request to the system administrator/list owner, the DL Manager verifies the authenticity of the request by sending an e-mail to the potential subscriber. The mail indicates that the request has been received and that it must be returned to the DL Manager before the subscriber's address can be added to the mailing list. If the potential subscriber replies to the e-mail, the DL Manager forwards the request to the system administrator/list owner and waits for the latter's reply. If the DL Manager encounters the word deny in the subject header of the message sent by the system administrator, it will reject the application of the attempting subscriber. Otherwise, the potential subscriber's address will be added to the mailing list. This feature is very useful in managing public mailing lists since some of those applying for subscription may be using forged identities.

The DL Manager also has the capability to add subscribers to the list after the authenticity of said subscribers has been established. When the DL Manager receives a request to unsubscribe, it first checks the validity of the request via the same authentication procedure used when adding subscribers to the mailing list. The list owner/system administrator is provided with a web-based user interface for removing subscribers from the list.

The DL Manager allows the system administrator to perform a list of operations, such as:

- Create New Lists
- Delete Lists
- Search for mailing lists
- Provide a descriptive information of the mailing list
- Modify list settings
- Add/edit subscribers
- Update list owner password

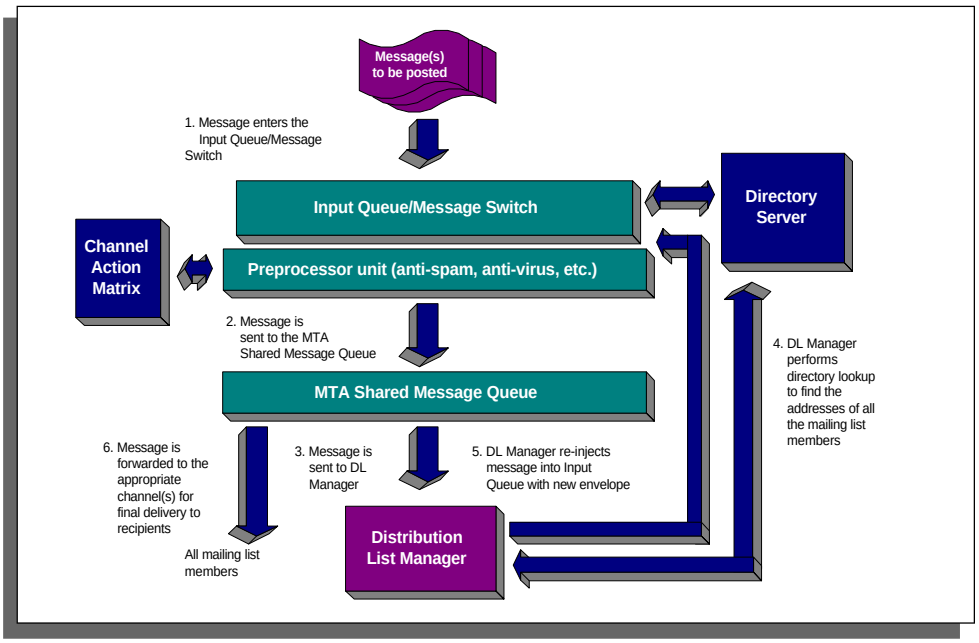

Figure 4A: DL Manager performs directory lookup to determine messages destined for a mailing list

Message Flow

When mail arrives at the Message Switch (see **Figure 4A** above), a directory lookup is performed using the Internet Exchange Directory Server to determine whether there are messages destined for a mailing list. If such messages are found, they are routed to the DL channel via the Preprocessor Unit and the MTA Shared Message Queue. Upon receiving a message destined for a mailing list, the DL Manager performs a directory lookup using also the Directory Server to find the corresponding addresses of the list members. After the mailing list members are identified using directory information, the message is forwarded to the Preprocessor Unit where appropriate actions (i.e., virus scanning, spam control and automatic disclaimer insertion) are performed based on the configuration in the Channel Action Matrix. The message is then sent to the MTA Shared Message Queue and subsequently forwarded to the appropriate channels (i.e., SMTPC, cc:Mail, Local, etc.) for final delivery to the mailing list members.

Types of Distribution Lists

There are two types of distribution lists that are commonly used in the Internet, the open and closed distribution lists. Internet Exchange DL Manager supports both types to provide both system administrator/list owner and subscribers with a wider range of options.

Open Distribution Lists

Open distribution lists or unrestricted e-mail-based discussion groups are very efficient tools for disseminating information and encouraging free exchange of ideas. With electronic mailing lists of this type, even non-members or non-subscribers have the privilege to post messages or access the list archives. However, only the members of the list can receive messages posted by members and non-members.

Closed Distribution Lists

A closed distribution list is accessible only to the list members. Only those people who subscribed to the list can post messages and/or access the list archives.

Those who want to post messages on closed mailing lists must first apply for membership. The list owner/system administrator exercises control over the application process.

Usually, membership in closed lists require the approval of the list owner/system administrator or the recommendation of a current member of the list being subscribed to.

Key Features

Mail Blocking

The mail blocking feature is particularly useful for managing open distribution lists. It allows the list owner/system administrator to prevent specific users from sending mail to the server. This feature can be implemented based on either the address or domain of the sender, which can be obtained from the envelope information provided by the connecting client. Upon receiving a message, the DL Manager performs a directory lookup to determine whether the address of the sender is included in the list of blocked addresses. If a match is found, the message is bounced back to the sender. Otherwise, the DL Manager processes the message and sends it back to the list members. For closed lists, the list owner/system administrator has the right to prevent a list member from posting messages if his messages only serve as a nuisance to the group. By invoking the mail-blocking feature, the member's privilege to post messages can be revoked, although he can still receive messages posted by other members or access the list's archives. Another option for dealing with such situations is to remove the offending subscriber from the list.

Automatic Mailing List Subscriptions

In automatic mailing list subscriptions, when the DL Manager receives a subscription request (see **Figure 4B** on page 16), it first checks the type of list—open list or closed list—the sender is trying to subscribe to. If it is an open list, the DL Manager activates automatic subscription. A confirmation message is then sent to the new member informing him that the subscription request has been approved. Upon receipt of the confirmation message from the subscriber, the subscriber is automatically added to the requested list. If it is a closed list, the DL Manager passes the subscription request to the list maintainer (see **Figure 4C** on page 16). The list maintainer must send an e-mail to the potential subscriber for verification purposes. If the potential subscriber replies to the e-mail sent by the list maintainer, then his e-mail address will be added to the mailing list (see **Figure 4D** on page 16).

Automatic Mailing List Unsubscriptions

In automatic mailing list unsubscription, the DL Manager handles unsubscription request by automatically removing the member from the mailing list. It first checks if the sender is a registered member of the mailing list. If not, the DL Manager logs an error indicating that the sender is not a member of the mailing list. If the DL Manager verifies that the sender is a registered list member, the sender is automatically removed from that list.

Delivery Modes

The DL Manager offers two modes of delivery—immediate and digest. In *immediate* mode, when messages are posted to a mailing list, the DL Manager sends them immediately to the mailing list subscribers. The immediate mode is the default setting. If a subscriber wishes to modify his account to be in the digest mode, he must send a request to the list owner/system administrator.

In the *digest* mode, posted messages are allowed to accumulate in the local archive of the members who selected this mode and are sent to the subscriber based on a

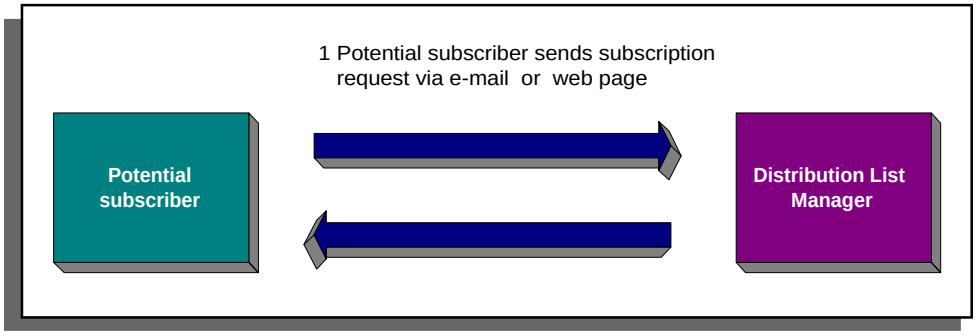

Figure 4B: Step 1 of the subscription process

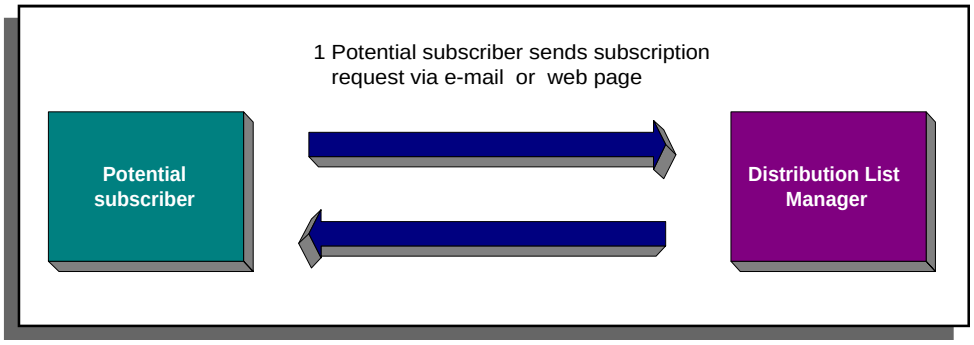

Figure 4C: Step 2 of the subscription process

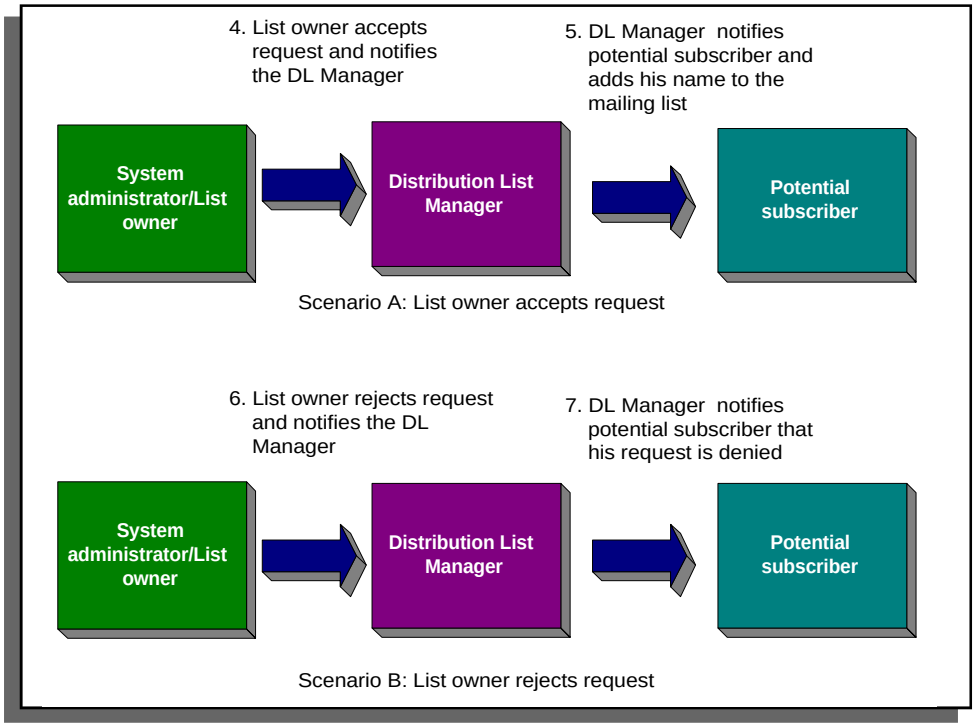

Figure 4D: Step 3 of the subscription process

pre-determined schedule set by the list owner/system administrator as requested by the subscriber. The delivery schedule is based on several parameters configured by the list owner/system administrator, such as the day of delivery, time of delivery, and the maximum number of messages that can be stored as configured in the archive. A web-based user interface is provided to enable the list owner/system administrator to set the option preferred by each subscriber.

Distribution List Manager Engine

The DL Manager engine monitors the file operations, specifically the delivery of messages to the designated lists. It is also responsible for the delivery of messages to members, regardless of the mode of delivery, and for performing automatic subscriptions/unsubscriptions. The DL Manager engine continuously checks the appropriate channels for new mail to ensure minimum delay in mail delivery.

Archiving

The DL Manager engine also performs archiving. The engine, optionally, keeps a copy of a message received by a mailing list. The archived messages are stored in the DL Manager home directory. Every archived message contains important information, such as the From:, To:, Date: and Subject: headers, as well as the message body. Each mailing list has its own archive directory where all the archived messages are stored.

The DL Manager archiving feature allows both system administrators and end users to view previous messages posted to lists; sort messages by Date, Thread or Author; allowing mailing list members and non-members to access mailing list archives of open lists; and allow mailing list members to access mailing list archives of closed lists using any web browser, such as Internet Explorer, Netscape Navigator, among others.

Conclusion

The DL Manager allows messages to be sent to all lists subscribers by simply submitting the said messages to a single address. It enables the system administrator to create electronic mailing lists that suppoprts the following features: mail blocking, automatic mailing list subscription/unsubscription, and setting the preferred delivery options. The DL Manager provides the system administrator with the option to accept or reject subscribers to the mailing list.

Lotus Notes Connector

The Lotus Notes Connector is a plug-in module that connects Lotus Notes environment to the Internet (see **Figure 5A** on page 19). Using this module, you can send and receive messages to and from the Internet, communicate with other local channels, and provide a rich migration path for moving Lotus Notes users to open Internet messaging standards. As a plug-in module, the Lotus Notes connector features the following:

- Anti-virus support
- Anti-spam support
- BSMTP
- Directory Server
- ESMTP support
- Unlimited scalability

Message Flow

Internet Exchange to Notes Environment

If a message received by Internet Exchange is destined for a Lotus Notes user "John Doe/Notes Domain" whose Internet address is "jdoe@domain.com", NOTESIN connects to the Notes Server to perform Notes Public Address Book (PAB) lookup in the Notes Server. If "John Doe/Notes Domain"exists in the Notes Address Book, NOTESIN translates the recipient Internet address "jdoe@domain.com" to its equivalent Notes address "John Doe/Notes Domain". Then, NOTESIN performs directory lookup in the Directory Server to determine the receive permission of "John Doe". After determining the receive permission, the message will be imported from Message Queue to the Notes Server, which will then deliver the message to "John Doe/Notes Domain" mailbox.

Notes Environment to Internet Exchange

A message coming from the Notes environment destined for the Internet Exchange is processed by the NOTESOUT channel. NOTESOUT fetches the message from the Notes Server, which obtains proper routing information of the message. For example, a registered Notes user named "John Doe/Notes Domain" sent a message to an Internet Exchange local user "sample@ima.com", NOTESOUT channel processor translates the sender's Notes username, which is "John Doe/Notes Domain" to its equivalent Internet address by performing a directory lookup in the Directory Server. If the sender's address does not exist in the Directory Server, NOTESOUT translates the sender's address by using the Default Mapping method. This method translates the sender's address in either of the following format:

Firstname_Lastname@domain.com **or** Firstname.Lastname.domain.com.

After translating the sender's address, NOTESOUT exports the message from the Notes Server to Internet Exchange, which routes the message to its intended recipient "sample@ima.com".

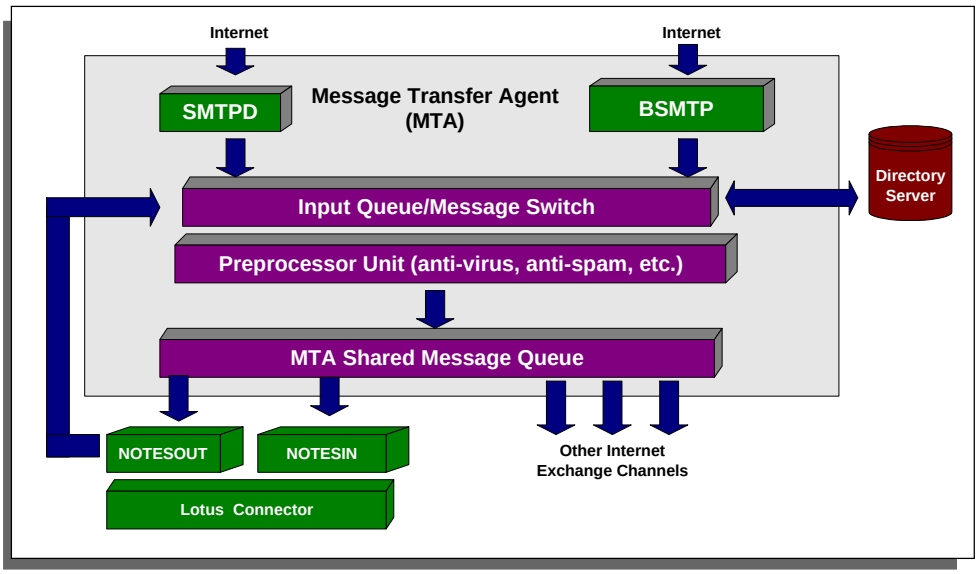

Figure 5A: Connecting Lotus Notes environment to the Internet

Migration Strategy

Objectives of the Migration Process

As more and more organizations move from legacy or LAN-based mail systems to non-proprietary, Internet-based messaging systems, it becomes apparent that system administrators need cost-effective tools that:

- provides total control over the entire migration process
- supports non-destructive migration
- supports automated and batch processes for migrating large user groups
- provides an audit trail of the migration process for tracking which information/users have been moved to the new system

Internet Exchange supports an extensive migration strategy for moving existing Lotus Notes users to non-proprietary, Internet-based messaging systems. The major goals of the migration process are:

- to migrate the Lotus Notes user database (address book) to the Internet Exchange Directory Server.

- to migrate the Lotus Notes user mailbox (including folders) to the Internet Exchange Local Mailbox Storage, which is used by the IMAP4 and POP3 servers.

Migration Tools

The migration tools included in the software are 32-bit Windows applications which can be run via GUI or in batch processing mode. These tools provide logging facilities to:

- track all user addresses that have been exported
- track all user mailboxes that have been migrated
- log all the errors encountered during the migration process

The migration tools uses two standalone programs: a directory and mailbox conversion tool. The directory conversion tool translates Lotus Notes directory to the Internet Exchange directory format. It uses pre-defined rules to create a valid RFC-822 Internet address for the user being migrated. To support this process, the migration tools allow Lotus Notes hierachical certifiers to be mapped to different Internet domains. The mailbox conversion tool converts Lotus Notes mailboxes into a format supported by the Internet Exchange Message Store, where users can access the messages using IMAP4 and/or POP3 clients, or the Web Mail Client.

For Lotus Notes users who already have Internet Exchange 3.x, the migration tools will allow them to re-use their user Alias database (smtpadr.btr) and Domain Mapping database (smtppod.btr) when constructing the directory. The migration tools use the information stored in the individual databases to construct the fully RFC-822-compliant address of a particular Notes user.

Directory Synchronization

The Internet Exchange architecture is specifically designed to support disparate messaging systems (i.e., cc:Mail, Lotus Notes). It was incorporated in its design that each disparate messaging system have its own local directory for storing user records. To keep all directories in the system synchronized, the Directory Server comes with a replication engine that replicates user records between the cc:Mail and Lotus Notes directories and the Directory Server.

Conclusion

The Lotus Notes connector provides a tool for connecting Lotus Notes environment to the Internet. Using this connector, Lotus Notes users can send and receive messages to and from the Internet, communicate with other local channels, and provide a rich migration path for moving Lotus Notes users to open Internet messaging standards.

Message Store

The Message Store is more than just a dedicated mail repository for remotely storing, retrieving and manipulating messages. It allows the system administrator to limit the amount of storage space allocated to a user, preventing the user from consuming all of the available disk space in the server. Its mail filtering utility enables the system administrator to define rules so that the LMDA can direct messages to pre-selected mailboxes or folders other than the user's Inbox.

Moreover, the Message Store includes both the POP3 and IMAP4 servers (see **Figure 6A** below), which are capable of creating multiple threads to support simultaneous access and retrieval of messages in the Message Store. It also enables users to access their mailboxes via POP3- or IMAP4-capable clients, such as Microsoft Outlook, Netscape Navigator, Eudora Mail, as well as the Internet Exchange Web Mail Client.

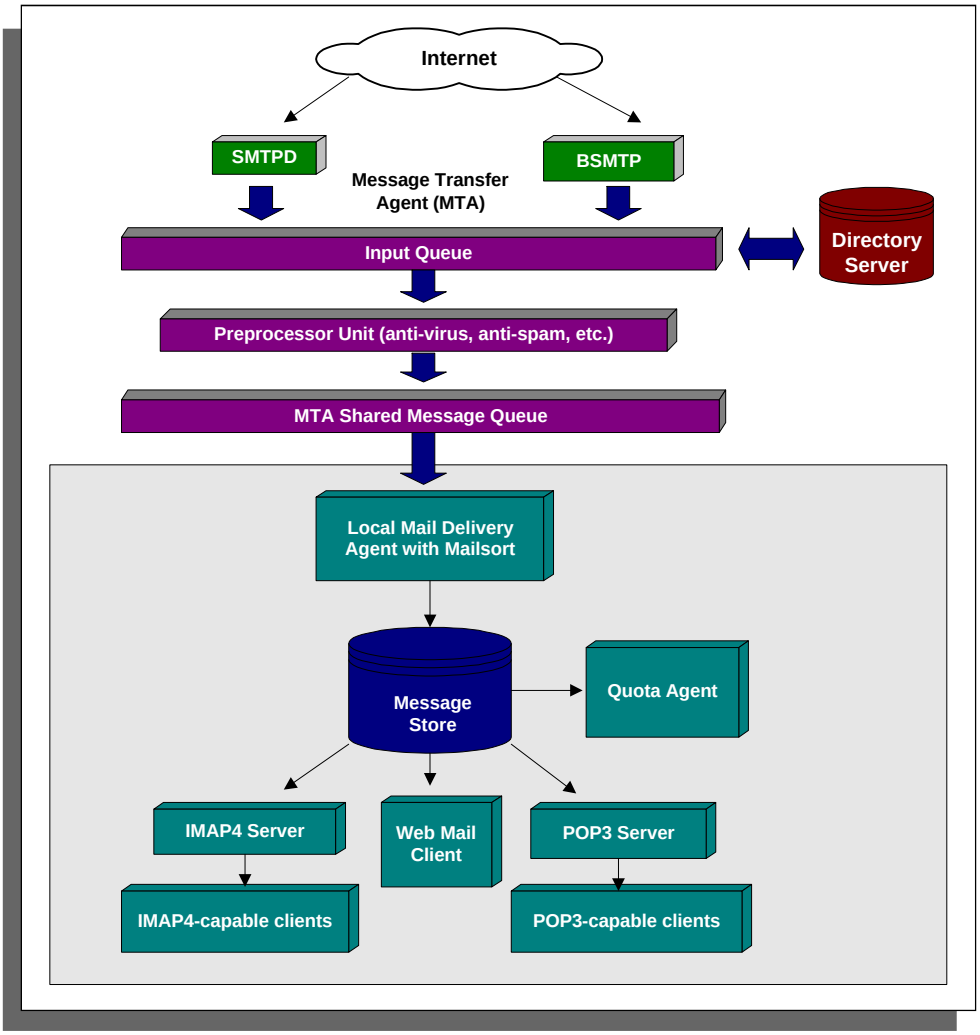

Figure 6A: Message Store Architecture

The Message Store consists of the Message Status Database, Message Envelope Database and Message Body Database. IMAP4-related attributes and RFC-822 header information are stored in the Message Status Database and Message Envelope Database, respectively. The Message Body Database stores the body structure of all messages in a given mailbox. Access to the different databases in the Message Store is carried out via the Message Store API.

Components

IMAP4 Server

The IMAP4 Server allows both the system administrator and end users to access their mailboxes via IMAP4-capable clients, such as Microsoft Outlook Express, Netscape Communicator, among others. By utilizing IMAP4, both the system administrator and end users can manipulate their mailboxes/folders on the server without having to download them to their local hard disk.

POP3 Server

Using POP3, users can retrieve messages from the Internet Exchange Message Store Inbox and store them in their local hard disk so they can be read later on in an offline or disconnected state. The POP3 Server supports multi-threading for fast message retrieval.

Mailsort

The Mailsort utility allows Message Store users to define filtering rules that the LMDA would refer to in copying, forwarding or moving messages to pre-selected mailboxes/folders other than the Inbox. It can also generate automatic replies to incoming messages based on a pre-defined criteria. The Mailsort filtering utility applies rules based on certain attributes (i.e., message sender, recipient or subject) in processing incoming mail at message delivery time. The Mailsort module also has the ability to reject messages coming from user-defined e-mail addresses.

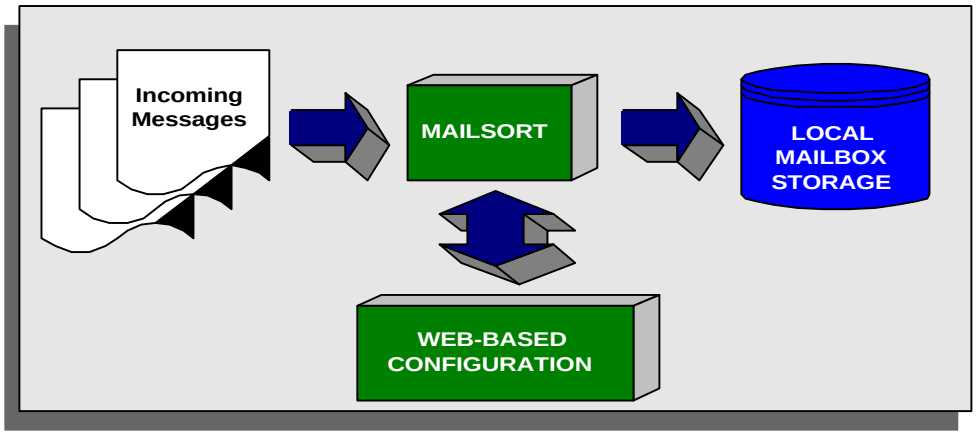

Figure 6B: Message Filtering via the Mailsort Utility

Before delivering a message to the Message Store, the LMDA first checks the message recipient's directory for a Mailsort filter file used for sorting incoming messages (see **Figure 6B**). Based on the filtering rules defined in the filter file, the LMDA will automatically deliver the message to the recipient's preferred folder or forward it to a new e-mail address on a per message basis.

Vacation Utility

The Mailsort module includes a vacation utility that enables individual users to configure their account to automatically reply to incoming messages.

Quota Agent

The Message Store Quota Agent is a system tool that allows the system administrator to set and enforce disk usage quotas on all Message Store user accounts. This feature limits the amount of resources allocated to individual users to prevent them from consuming all of the available disk space in the server. It also allows the system administrator to monitor the total number of registered users and determine the users who have exceeded their disk quotas.

The Quota Agent generates reports in HTML and text file formats, which the system administrator uses in checking and verifying the Message Store's performance and disk space usage. The reports in HTML format are available in the Internet Exchange "Message Store" web interface, while the text file reports are sent as file attachments to the system administrator. The Quota Agent can be run instantly or during a scheduled time of activation.

Web Mail Client

The Web Mail Client has a customizable web interface that allows Internet Exchange users to compose, reply and forward messages via Internet Exchange using any web browser, such as Internet Explorer, Netscape Navigator, among others, anytime, anywhere. Users can now read and send mail at their own convenience.

Messages from the Web Mail Client are routed to the Internet via Internet Exchange. The Web Mail Client uses the Message Queue API in submitting messages to the Message Queue. The Web Mail Client CGI (Common Gateway Interface) uses the Message Store API to directly access the Mailbox Database and Mail File, without using IMAP4 or POP3.

Conclusion

The Message Store is more than just a dedicated mail repository for remotely storing, retrieving and manipulating messages, while also enabling users to access their mailboxes via POP3- and/or IMAP4-capable clients or via the web using the Internet Exchange Web Mail Client. It allows the system administrator to limit the amount of storage space allocated to a user, preventing the user from consuming all of the available disk space in the server. Its mail filtering utility enables the system administrator to define rules so that the LMDA can direct messages to pre-selected mailboxes or folders other than the user's Inbox.

Message Transfer Agent

The MTA (Message Transfer Agent) routes messages received by the Preprocessor to their intended channels (see **Figure 8A** below). To speed throughput, the MTA incorporates a shared queuing structure where messages are moved through logical queues in the system without requiring the re-writing of the message. Upon receiving a message, MTA temporarily stores the message in the MTA Shared Message Queue while analyzing the recipient's address. It will either deliver the message to the recipient's local address or forward the mail to another MTA.

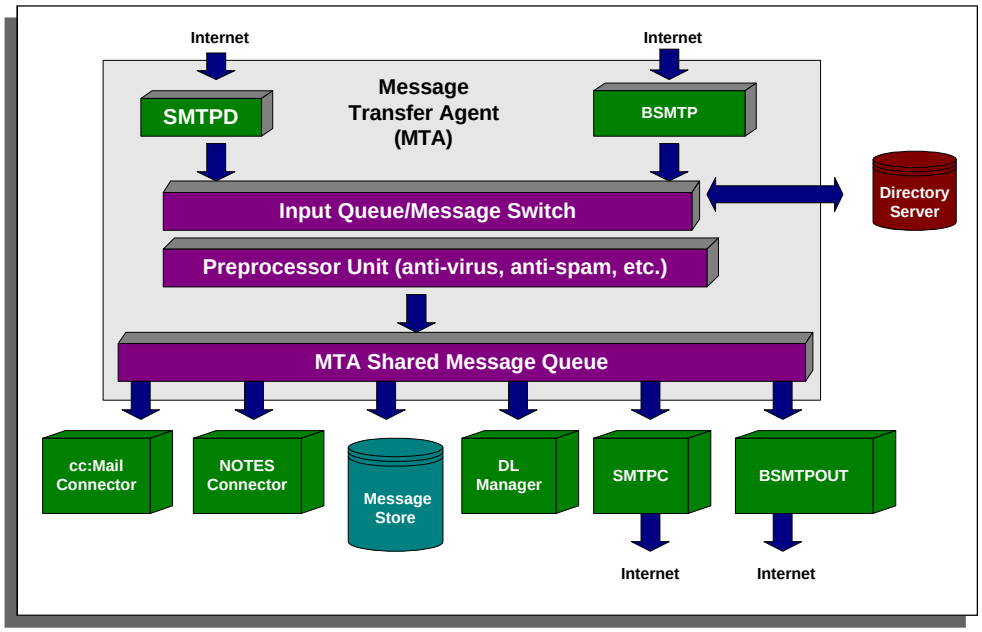

Figure 8A: Message Transfer Agent System Architecture

Components

Input Channels

A channel is a path through which messages flow. It makes use of a specific protocol to format and transfer messages. Internet Exchange MTA makes use of a number of input channels in receiving messages from the Internet or other messaging systems, like cc:Mail and Notes Mail. These include:

- SMTPD - for messages received from the Internet via the standard SMTP
- BSMTPIN - for messages received via a POP3 connection
- CCOUT - for messages received from the cc:Mail environment
- NOTESOUT - for messages received from the Lotus Notes environment

Note: *CCOUT or NOTESOUT export messages from the cc:Mail or Lotus Notes environment and input them into the Internet Exchange MTA.*

- DL - for messages sent to a distribution list. When messages from the Internet are received by the input channels, they are temporarily stored in the input queue after which they shall be fetched by the Preprocessor for further processing

- WEB MAIL CLIENT - for messages coming from the Message Store using the Web Mail Client.

Output Channels

The Internet Exchange also makes use of a number of output channels in routing messages to the Internet or other messaging systems, like cc:Mail and Notes Mail. These include the following: SMTPC, BSMTPOUT, DL, LOCAL, CCIN and NOTESIN. These channel processors are responsible for fetching messages from the MTA Shared Message Queue and delivering them to their intended recipients.

Preprocessor

Once the Preprocessor receives messages, it performs directory lookup from the Directory Server to determine the proper channels/connectors defined to route the message to its intended recipients. After determining the routing information for each message, the Preprocessor performs virus scanning, spam control and automatic disclaimer insertion on the messages.

The Preprocessor engine consists of the following modules: AntiVirus, SpamArchive, SpamDelete, SpamBounce, LoopDetection and AutoInsertion. Each module has its own Channel Action Matrix, which is used to configure the preprocessor module that should run on a defined input/output channel combination.

MTA Shared Message Queue

After the Preprocessor processes the messages, the messages are temporarily stored in the MTA Shared Message Queue before they are fetched and delivered by the different input/output channels to their intended recipients.

Key Features

Monitor Control Responder

The MC (Monitor Control) Responder user interface allows the system administrator to monitor and control the status of various Internet Exchange components. It allows the system administrator to Start or Stop the Responder, which thereby starts or stops all the installed modules at the same time.

The administrator may also activate certain options, such as the Auto Start, Auto Restart and Auto Stop for a specific component. There is also a field where the user can define the Wait Time, which is the amount of time the MC Responder has to wait before running the component. Furthermore, this web interface also has a button for changing the status of the components from Stop to Running or vice versa.

Dialup Scheduler

Having a dedicated Internet connection is ideal. However, in certain cases, it may be impractical to maintain a permanent Internet connection. In these cases,

it is desirable to use a dial-up mechanism, which establishes connection to an ISP (Internet Service Provider) at a particular time to download and upload messages to and from the Internet.

The Internet Exchange MTA utilizes the RAS (Remote Access Service) dial-up mechanism. RAS is the service by which the Windows Operating System allows the local system to dial and connect to another peer over the Internet. The MTA initiates RAS through the Dialup Scheduler that supports the following functions:

- provides a user interface to enable the system administrator to configure dial-up schedules and other RAS connection-related information
- performs RAS dial up at the scheduled dial-up time
- performs RAS connection hang up at the scheduled hang-up time

Conclusion

The widespread use of e-mail over the Internet did not only create a strong demand for messaging systems capable of moving messages between an arbitrary number of channels with minimum delay. It also posed a serious challenge to system administrators who wish to protect the integrity of their networks against threats, such as computer viruses and spam mail. Thus, many organizations are pressured to invest heavily on products that will enable them to maximize throughput without sacrificing the integrity of their systems. The MTA is specifically designed to address these issues.

The MTA features an innovative queuing strategy that enables the messaging server to handle large numbers of messages bound for different channels without experiencing disk I/O bottleneck common in messaging systems. This increases throughput considerably and enables organizations to send and receive timely information that may be critical to business success.

MigrationTools

The ever-decreasing gap between corporate Intranets and the global Internet has opened the door to seamless electronic communication. Using open, Internet-based standards, corporate users can now exploit the richness of the Internet to communicate with other users within their company as well as customers, partners and suppliers worldwide. These standards enable the formation of a rich set of messaging, directory and collaboration services that work inside and outside the company. To seize this growing business opportunity, companies need to move from legacy messaging systems to systems that support the following standards:

- SMTP
- IMAP4
- BSMTP
- POP3
- LDAP
- MIME (Multipurpose Internet Mail Extensions)

Moving from legacy systems to open, Internet-based systems means migrating a variety of information, such as:

- Messages
- Attachments
- Folders and folder hierarchy
- Distribution lists
- Private address books
- Archives and bulletin boards
- Address book/directory

During the migration period, users typically need to access information from both the legacy systems and the new open systems. To make migration as simple as possible, Internet Exchange Messaging Server includes several migration tools to assist IT (Information Technology) managers in migrating existing cc:Mail or Lotus Notes customers to Internet Exchange open messaging environment. The software uses two main migration tools: the directory and mailbox conversion tools.

Directory Conversion Tools

The cc:Mail/Notes directory-to-Internet Exchange directory converter translates the address book information from cc:Mail or Notes directories to a format supported by the Directory Server.

Mailbox Conversion Tools

The cc:Mail/Notes mailbox-to-Internet Exchange Message Store converter translates cc:Mail and Lotus Notes mailboxes into a format supported by the Internet Exchange Message Store where they can be accessed via the IMAP4 or POP3 servers, or Web Mail Client. This tool has two modes, namely:

- Per user mode
  Individual users can convert their mailbox into the standard Internet Exchange mailbox.

- Batch processing
  User's mailbox can be converted into a format supported by the standard Internet Exchange Message Store based on data present in a local configuration file.

Note: *It is important that the system administrator runs the address book converter before running the mailbox converter so that the users are registered in the Internet Exchange directory.*

The migration tools provide logging facilities used to:

- track all user addresses that have been exported
- track all user mailboxes that have been migrated
- log all the errors encountered during migration

Migration Strategy for cc:Mail Users

The cc:Mail migration tools supports both DB6 and DB8 post offices. The tools are a set of 32-bit Windows applications that uses the VIM (Vendor Independent Messaging) interface to access the cc:Mail Post Office (see **Figure 7A** on page 26). It is imperative that the migration tools should run on a PC that has the VIM libraries pre-loaded and available on the path. The migration tools for cc:Mail are designed to run on Windows 95, Windows 98 and Windows NT.

When migrating individual cc:Mail users' mailboxes to the Internet Exchange Message Store, the system administrator must have the following:

- the password of the individual users; and
- the appropriate access rights to the individual user's mailbox

cc:Mail User Name Information

The structure of the cc:Mail address book is shown below, **Figure 7B**.

Name	Location	Comments	cc:Mail address	Last Login
Victor Wong	L	IMA Manager		12/12/97 8:10PM
Mary	R	Remote CC	PO2	
Main-PO	P	Main PO		
Internet	P	Internet PO		
PO2	P	Downstream PO	\\PC2\CCDATA	
Peter	R	Remote User	Internet peter@ima.net	

Figure 7B: cc:Mail Address Book Structure

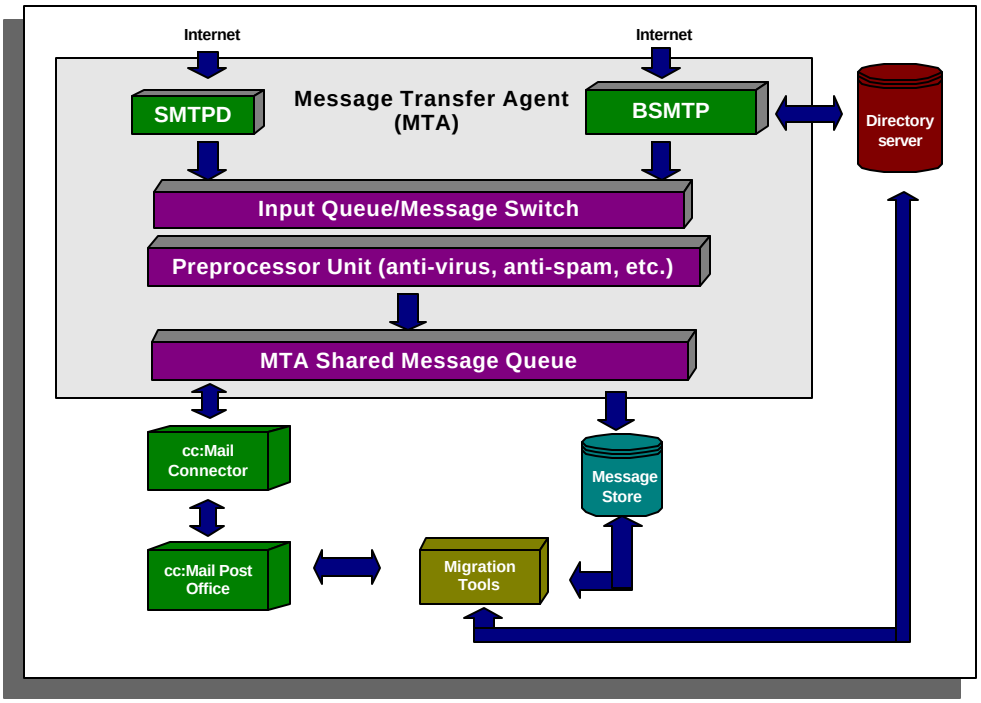

Figure 7A: Migration Strategy for cc:Mail users

The migration tools require that the NAME field in the cc:Mail user record be exported to the directory. Since the cc:Mail user name format is not fully RFC-822 compliant, the migration tools re-construct the cc:Mail name into a fully RFC-822-compliant address. This is carried out by getting the appropriate information from a pre-defined rule configured by the user. In general, the cc:Mail user name is in the following format:

<USER>at<PO NAME>

where user can be anything like:

<FIRSTNAME LASTNAME> or <LASTNAME FIRSTNAME>

In cc:Mail, characters like , and @ can be used in the user name field. This leads to the special requirement of handling characters, which are not valid as per RFC-822 addressing standard. cc:Mail supports multiple languages like European and Asian languages, raising the need to deal with special European characters and Asian characters, such as Japanese and Chinese, which are actually stored in double byte format.

The <PO NAME> of a cc:Mail user address denotes the physical location of the mailbox. The migration tools are capable of mapping different Post Offices to different domains while constructing user's Internet addresses during the migration process. Once a valid Internet address is constructed, a dummy password will be generated, which will either be a MD5-hashed password or an 8-byte long ASCII string.

For users who are already running Internet Exchange 3.x, the migration tools allow the users to re-use the user Alias database (smtpadr.btr), directory database (rulebadr.btr) and Domain Mapping database (smtppod.btr) when constructing the directory. The conversion tool looks up the individual databases to construct the fully RFC-822-compliant address of a particular cc:Mail user.

The migration tools provide users with pre-defined rules to construct a valid RFC-822 address. The pre-defined rules set are as follows:

- FirstName LastName
- FirsName M1 LastName
- F1 LastName
- LastName FirstName
- LastName FirstName M1
- LastName F1
- LastName F1 M1
- FirstName L1 with separator = NONE/UnderScore and/or Dot

Note: *RFC-822 defines the addressing syntax used for Internet electronic mail addresses*

For the Internet domain name part, the migration tools allow the user to define mappings between cc:Mail Post Office name and the Internet domain name. The migration tools then use the rule set and the mappings to construct a fully RFC-822-compliant Internet address.

cc:Mail User Mailbox

The migration tools are designed to convert cc:Mail mailbox files to the Internet Exchange Message Store. First, each message is independently exported from the cc:Mail Post Office via the VIM interface. Then, it is converted into a single part MIME message (if the cc:Mail message contains only 1 item) or is converted as a multipart/mixed MIME message (if the cc:Mail message contains multiple items). The conversion steps are defined as follows:

- Convert cc:Mail notepart item into TEXT/PLAIN MIME body. The conversion tool assigns the character set of the TEXT/PLAIN item to the pre-configured value.
- If the character set is pre-configured to ISO-2022-JP, the notepart item will be converted from 8- to 7-bit JIS based on ISO-2022-JP format.
- The conversion tool converts the cc:Mail attachments into appropriate MIME types.
- If the attachment is in AppleSingle format, the attachment is converted as MacMIME Applesingle (base64 encoded) MIME body. This is stated under the assumption that the user will use MacMIME-compliant IMAP/POP3 agent to access the mailbox.
- All the user address in the cc:Mail headers (To:, Cc: and Bcc:) are converted into an RFC-822-complaint address format.
- The conversion tool retains the directory structure as laid out in the cc:Mail Post Office.

Migration Strategy for Lotus Notes users

The Lotus Notes migration tools support Lotus Notes version 4.x. The tools, which run on Microsoft Windows NT, are a set of 32-bit Windows applications that use the Notes API to access the Notes server/mailbox. It is imperative that the migration tools should run on a PC that uses the Notes API to access Notes server/mailbox (see **Figure 7C** below). It is also recommended that the tools should run on a machine configured as a Notes server rather than as a Notes client.

When migrating individual users mailbox to the Internet Exchange Message Store, the system administrator must have the following:

- user ID file for individual users;
- password of the individual users; and
- appropriate access rights to the individual user mailbox

Lotus Notes User Name Information

The structure of the Lotus Notes PAB is given in **Figure 7D**, page 29.

The migration tools require that the NAME field and the DOMAIN field be exported to the Internet Exchange directory from the Lotus Notes PAB. Since the Lotus Notes user name is not fully RFC-822 compliant, the migration tools re-construct the Lotus Notes name into a fully RFC-822-compliant address. This is performed by getting the appropriate information from a pre-defined rule configured by the user. In general, the Lotus Notes user name is in the following format:

<USER>/<CERTIFIER>@<DOMAIN>

where user can be anything like:

<FIRSTNAME LASTNAME> or <LASTNAME FIRSTNAME>

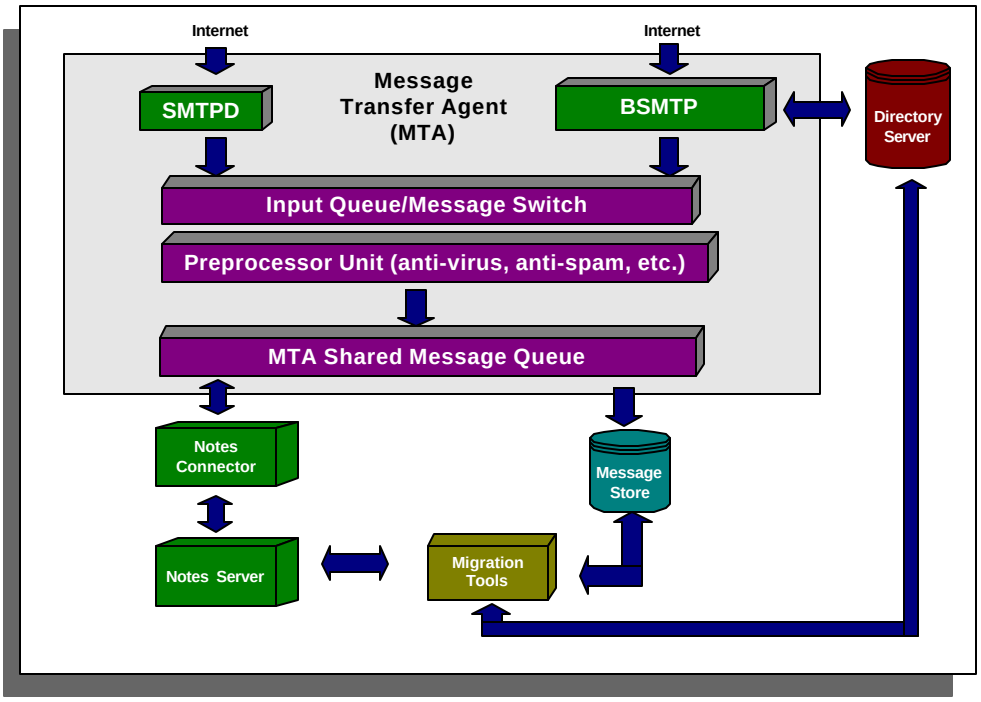

Figure 7C: Migrating from Lotus Notes to Internet Exchange

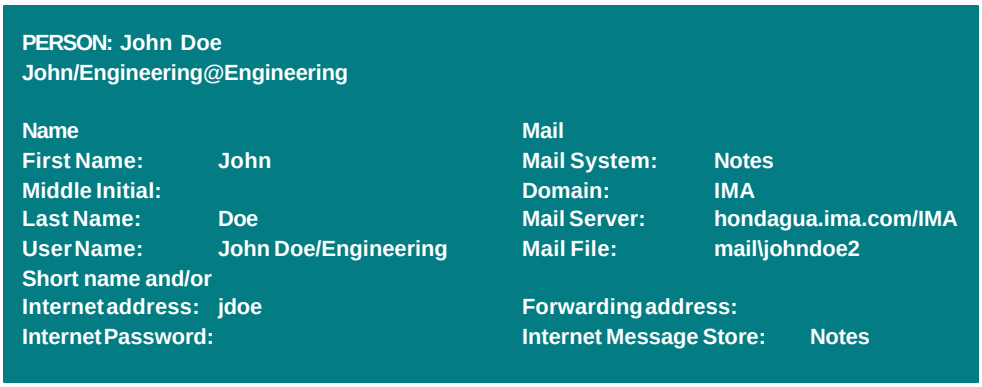

Figure 7D: Structure of the Lotus Notes Address Book

Lotus Notes supports multiple languages like European and Asian languages, raising the need to deal with special European characters and Asian characters, such as Japanese and Chinese, which are actually stored in double byte format.

The <CERTIFIER> of a Lotus Notes user address denotes the departmental location of the mailbox (i.e., John Doe/Engr/IMA@IMA). The migration tools are capable of mapping different certifiers to different domains while constructing the user's Internet addresses during the migration process. Once a valid Internet address is constructed, a dummy password is generated, which will either be a MD5-hashed password or an 8-byte long ASCII string.

For Lotus Notes users who are already running Internet Exchange 3.x, the migration tools allow the users to re-use the user Alias database (smtpadr.btr) and Domain Mapping database (smtppod.btr) when constructing the directory. The migration tools look up the individual databases to construct the fully RFC-822-compliant address of a particular Notes user.

The migration tools provide users with pre-defined rules to construct a valid RFC-822 address. The pre-defined rules set are:

- FirstName LastName
- FirsName M1 LastName
- F1 LastName
- F1 M1 LastName
- LastName FirstName
- LastName FirstName M1
- LastName F1
- LastName F1 M1
- FirstName L1 with separator = NONE/UnderScore and/or Dot

In addition to the aforementioned rules, the user has the option to use the short name/Internet name field from the Notes PAB. If this option is used, the above rules will be disabled. For the Internet domain name part, the migration tools allow the user to define mappings between certifier names and the Internet domain names. The migration tools then use the rule set and the mappings to construct a fully RFC-822-compliant Internet address.

Lotus Notes User Mailbox

The migration tools are designed to convert the Lotus Notes mailbox into a format supported by Internet Exchange Message Store. First, each message is independently exported from the Notes mailbox via the Notes API interface. Then, it is converted into a single part MIME message (if the Lotus Notes message contains only 1 item) or is converted into a multipart/mixed MIME message (if the Lotus Notes message contains multiple items). The conversion steps are defined as follows:

- Convert Notes mail body into TEXT/PLAIN MIME body. The conversion tool assigns the character set of the TEXT/PLAIN item to the pre-configured value.
- If the character set is pre-configured to ISO-2022-JP, the mail body will be converted from 8- to 7-bit JIS based on ISO-2022-JP format.
- The conversion tool converts any attachment into appropriate MIME types.
- If the attachment is in AppleSingle format, the attachment is converted as MacMIME AppleSingle (base64 encoded) MIME body. This is stated under the assumption that the user will use MacMIME-compliant IMAP4/POP3 clients to access the mailbox.
- All the user address in the Notes mail headers (To:, Cc: and Bcc:) are converted into an RFC-822-compliant address format.
- The conversion tool retains the directory structure as laid out in the Notes user mailbox.

Conclusion

The ever-decreasing gap between corporate Intranets and the global Internet has opened the door to seamless electronic communication. Using open, Internet-based standards, corporate users can now exploit the richness of the Internet to communicate with other users within their company as well as customers, partners and suppliers worldwide and seize the growing business opportunity created by shifting away from legacy messaging systems. These standards enable the formation of a rich set of messaging, directory and collaboration services that work inside and outside the company.

Internet Exchange provides comprehensive migration tools to help users move to an Internet standards-based environment smoothly and reliably. All address and mailbox information are transferred to the new system transparently, causing end users very minimal disruption. In addition, it supports the most common client software available in the market, such as cc:Mail and Lotus Notes, allowing end users to immediately make use of the system after the messaging system (backend) migration is completed.

MTA Preprocessor Unit

The Preprocessor unit is an integrated subsystem of the MTA with a highly scalable architecture (see **Figure 9A** below). Each of the unit's programs is a plug-in module that can run on separate machines, ensuring efficient usage of computing resources and maximum throughput. This capability also guarantees that the system can be easily scaled to cope with the changes in the messaging needs of an organization. In addition, the Preprocessor incorporates an open API that permits the development of third party or custom processing modules. Communication between the different Preprocessor modules is carried out via RPCs (Remote Procedure Calls) over TCP/IP (Transmission Control Protocol/Internet Protocol).

The Preprocessor, which runs the anti-spam, anti-virus, autotext insertions engines, and the TNEF (Transport Neutral Encapsulated Format) expander, is equipped with a Channel Action Matrix to provide the system administrator with a flexible tool in configuring which channels/connectors should be run for a particular message. This is based on certain parameters, such as the origin and destination of the message. For example, the system administrator may not wish to run the anti-virus module for messages coming from the local environment and destined for Message Store users. Another scenario would be, the system administrator would like his spam messages coming from the Web Mail Client and destined to the Internet to be deleted. This option can be easily implemented via the Channel Action Matrix's configurable GUI.

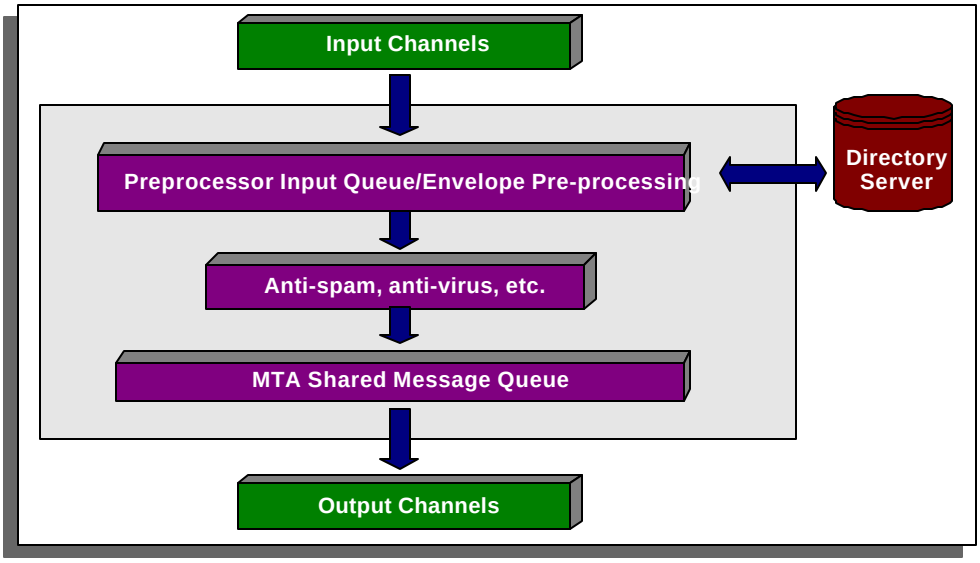

Figure 9A: Preprocessor Architecture

Components

Anti-Virus Module

The anti-virus module is a 32-bit multi-threaded standalone pre-processing module capable of performing simultaneous virus scanning on MIME and non-MIME message attachments. Each thread created by the anti-virus engine is responsible

for processing one message at a time.

When a message enters the anti-virus module, it first decodes the attachment then scans the said attachment by calling the anti-virus program indicated by the administrator. Once a virus is detected, the anti-virus module can optionally delete the message (with an option of notifying the system administrator that the message has been deleted), bounce to the message sender, or archive for later manual processing.

Internet Exchange currently supports the following anti-virus engines:

- McAfee ViruScan
- Sophos anti-virus for Windows 95, Windows 98 and NT
- F-PROT Professional anti-virus package

Anti-Spam Module

The anti-spam module provides the administrator with options to control the reception of unsolicited and unwanted spam messages. Internet Exchange can act as a mail relay, and can be defined to reject mail during the SMTP exchange from:

- Any number of host and domains
- IP (Internet Protocol) addresses
- IP address range
- Hosts with supplied names that cannot be verified via the DNS

or even based on the following message headers after message reception:

- From:
- Sender:
- Reply-To:
- Resent-From:
- Return-Path:

The anti-spam module is designed to protect the entire messaging system against unsolicited junk e-mail. This module enables the system administrator to create a list of banned or unwelcome IP addresses and IP address ranges using a configurable GUI . It is also capable of verifying the corresponding name of an IP address submitted by a remote host during the initial stage of an SMTP session via reverse DNS lookup to filter out forged names, thereby blocking out potential spammers even before they can enter the system. The anti-spam module supports a number of RBL (Real-time Blackhole List) for optimum protection. These are:

- MAPS-RBL (Mail Abuse Prevention System's Real-time Blackhole List)
- MAPS-DUL (Mail Abuse Prevention System's Dial-up User List)
- IMRSS (Internet Mail Relay Services Survey Project)
- ORBS (Open Relay Behavior-Modification System)
- DSSL (Dynamic IP Spam Source List)

Auto Text Insertion Engine

The auto text insertion engine has the capability to insert disclaimer into messages passing through Internet Exchange. The administrator can add different disclaimer

messages based on the message source channel. The engine allows the system administrator to use plain and/or HTML text in the insertion process. The engine currently supports RFC-822 message (non-MIME) and MIME message structure types.

The Preprocessor invokes the auto insertion engine based on the configuration in the Channel Action Matrix. The engine provides a second level of configuration where users can define disclaimers per source channel. The content of the dislcaimer is appended to the appropriate section of the message.

Channel Action Matrix

The Channel Action Matrix provides the system administrator with a flexible tool for configuring which modules the Preprocessor should run for messages flowing through a particular channel in Internet Exchange.

TNEF Expander

The TNEF expander is implemented as a plug-in program. The main purpose of this module is to handle TNEF message attachments by:

- extracting any embedded attachments from MIME applications/ ms-tnef body parts; and
- re-submitting the extracted attachments

After the recipient channel for a message has been determined, the expander performs MIME level parsing to locate the part of the MIME message, which contains *Content-Type: applications/ms-tnef*. Once this label is found, the expander calls on some functions to check and break the TNEF data stream. If the MIME body is found to contain invalid TNEF data stream, the expander skips this stream and goes to the next available TNEF data stream. If there are no valid TNEF body parts in the message file, no further actions will be taken. The message will then be submitted to the recipient like any normal message. Otherwise, the recipients of a message with TNEF attachments will receive two messages. One message contains the contents of the original message and the other message contains the extracted contents of the TNEF attachment.

Key Features

Queue Management Utility

Internet Exchange provides an enhanced Queue Management utility for the system administrator to view pending messages that has not been processed yet by their corresponding channels (DL, BSMTPOUT, Notes, ccMail, Local, SMTPC).

The system administrator may sort the pending messages for a particular channel according to any one of the following criteria: Priority, Sender and Size.

Messages can also be searched according to Sender's Address or Recipient's Address. After the sorting/searching criteria is specified, the queue management utility searches for all messages that matched the specified criterial. The results are displayed on a new page. The system administrator may either view the headers of the message, delete/bounce the message or reset the queue.

Domain Forwarding

The Domain Forwarding feature provides the necessary information about the different domain/channel mappings for a domain-based mail routing. A sample entry of the

Domain Forwarding is shown below.

Domain	Channel	Channel Identifier
smallcorp.com	BSMTPOUT	smallcorp@domain.com
othernet.org	BSMTPOUT	othernet@domain.com

Using the sample, once messages enter the Preprocessor, it determines if the Domain Forwarding is enabled for the domain name in the recipient address. If Domain Forwarding is defined, the Preprocessor forwards the messages to the defined BSMTP channel. The BSMTP Encoder encodes the messages using the domain address (e.g. smallcorp.com) and then re-submits them to the Message Switch for further routing and delivery to the address defined in the Channel Identifier (e.g. smallcorp@domain.com).

In the given example, all messages destined for **smallcorp.com** will be forwarded to the BSMTP channel with the BSMTP identifier's address *smallcorp@ domain.com*, while messages destined for **othernet.org** will be routed to *othernet@domain.com*.

Loop Detection

The Loop Detection feature enables the system administrator to configure the different parameters for defining the rules of message loops in the Internet Exchange. The system administrator may specify the maximum number of received lines (that show the FQDN of the MTA machine) allowed in an incoming message. Only lines containing the MTA FQDN are counted. If this number exceeds, the message will be bounced. If set, any looping messages will be bounced to the local postmaster instead of being returned to the remote sender.

Build Alias Table

The Build Alias Table feature enables the system administrator to extract all mail aliases from the Directory Server into a separate database. After extracting all mail aliases, the Preprocessor module updates an internal database that holds all the e-mail aliases available in the Directory.

This database is required by the Preprocessor to recognize recipients who use an alias name.

An alias is like a multiple identity of a user. You can create your mail alias in the Directory Server configuration page. Let us say your original e-mail address is *username@domain.com* and your alias name is *username@mail.domain.com*. When a message is sent to username@mail.domain.com, the Preprocessor also routes the message to username@domain.com.

Conclusion

The Preprocessor Unit is an integrated sub-system of the MTA with plug-in modules that can be run on separate machines. The Preprocessor plug-in modules include the anti-spam, anti-virus, auto text insertion engine, Channel Action Matrix and the TNEF expander.

The anti-virus module performs virus scanning on messages. The anti-spam protects the system from unsolicited mail. The auto text insertion engine has the capability to insert disclaimer into messages passing through Internet Exchange. The Channel Action Matrix provides the system administrator with a flexible tool for configuring which modules should the Preprocessor Unit run for a particular message. The TNEF expander handles TNEF message attachments.

Simple Mail Transfer Protocol

The Internet Exchange Messaging Server communicates with mail hosts on the Internet using the SMTP. This protocol is used for the submission as well as the reception of messages. To communicate well with the Internet, Internet Exchange implements SMTP as two separate modules. A client program, SMTPC (Simple Mail Transfer Protocol Client), sends messages from the local machine to the Internet. The server program, SMTPD (Simple Mail Transfer Protocol Daemon), receives messages from the Internet bound for the local environment. Both the SMTPD and SMTPC features a multi-threaded architecture (see **Figure 10A** below) that allows for concurrent transmission of messages.

To minimize the delay in message delivery, both the SMTPC and SMTPD components support the following ESMTP service extensions: DSN (Delivery Status Notification), message size extension, ETRN (Extended Turn) remote queue startup (primarily for disconnected or dial-up access) and 8-bit MIME. With ETRN support, dial-up SMTP hosts can notify the SMTP server when to deliver messages, allowing bandwidth resources to be used efficiently.

ETRN is an SMTP command issued by SMTPC when connecting to a remote SMTP servers. This command, which includes the FQDN of the Internet Exchange machine, requests the remote SMTP server to start processing its mail queues for messages that are addressed to the FQDN of Internet Exchange. If any such messages are at the server, the server creates a new SMTP session and sends the messages. This can be useful in a dial-up environment where SMTP servers usually send mail only at specific intervals. The support for ETRN requests ensures that even though there is no outbound mail in the host where the SMTPC runs, the host still receives inbound mail during the time of connection.

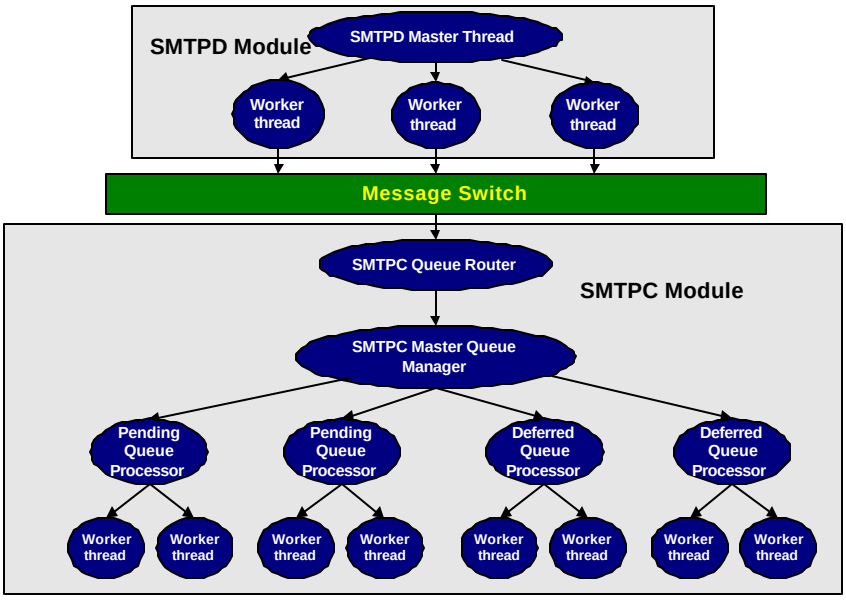

Figure 10A: SMTPD and SMTPC Modular Design

SMTPD

SMTPD is a background server process that receives messages from the Internet. Whenever a new connection request for incoming mail is detected, SMTPD creates a worker thread to manage the new connection. Once a message is received by the worker thread, SMTPD creates a queue entry for the message and performs anti-spam checks before it submits the message to the Internet Exchange MTA Shared Message Queue. Afterwards, SMTPD goes back to waiting for additional connection requests.

Multi-threaded Architecture

This feature enables SMTPD to create multiple threads that can handle multiple connections and manage simultaneous processing of messages. This multi-threaded architecture is made up of the following components:

Master Thread Manager

Continuously listens to the SMTP port for incoming connection requests. Upon receipt of a connection request, the Master Thread Manager creates a new worker thread to establish a new connection.

Worker Thread

Manages the connection and the transfer of messages between the Internet Exchange and the external SMTP servers.

Spam Control

SMTPD performs anti-spam checks on messages before routing them to the MTA Shared Message Queue. Using this feature, Internet Exchange can detect and act upon junk and spam messages even before they enter your system. Currently, the SMTPD module applies the following connection-based detection methods:

Site Network Blacklisting

Internet Exchange allows the system administrator to create a list of all the IP addresses and IP address ranges that are authorized to send messages to the Internet Exchange. If the SMTPD receives a connection request from a remote host whose IP address is not included in list, SMTPD automatically deny the connection.

Third-party relay prevention

The Internet Exchange SMTPD module has a list of remote sites that are authorized to relay messages to the Internet Exchange. During the SMTP transaction, where the message recipients are identified, Internet Exchange checks if the remote site is included in the list of sites authorized to relay messages. If not, the connection will be rejected.

Remote Name Verification

During SMTP start up, Internet Exchange identifies the remote machine that sent a connection request. Via the SMTP-HELO command, the remote machine sends its FQDN (Fully Qualified Domain Name). Internet Exchange, using the supplied FQDN, can perform reverse DNS lookup based on the known network address of the sending site to verify the name. If the supplied name and verified name do not match, SMTPD will terminate the connection.

SMTPC

SMTPC delivers messages to the Internet by regularly checking for messages queued in the SMTP OUT queue. It establishes the required number of connections with external SMTP servers and transfers the messages to the appropriate Internet mail hosts.

Hierarchical Multi-threaded Architecture

SMTPC features a hierarchical multi-threaded architecture, which assures high scalability and performance. This hierarchical multi-threaded architecture consists of two components:

SMTPC Queue Router

Retrieves messages from the Internet Exchange MTA Shared Queue and determines whether the messages should be routed to the Pending Queue or Deferred Queue.

SMTPC Master Queue Manager

Controls and synchronizes the following queues:

- Pending Queue
  Temporarily stores the messages that have to be sent out immediately to the Internet. The Pending Queue has one or more pending queue processors, which process the messages. The pending queue processors are responsible for establishing connections with the external SMTP servers and delivering the messages to the remote hosts via SMTP. If the delivery of a message in the Pending Queue is unsuccessful, it is passed on to the Deferred Queue for a later delivery.

- Deferred Queue
  Contains messages that are either intentionally deferred or whose previous delivery attempts have failed. These messages will be stored in the Deferred Queue and will not be delivered immediately. The delivery attempts are considered “failed” when any of the following conditions are encountered:

  a. The option queue mail before attempting delivery is enabled. When this option is enabled, messages will be placed to the Deferred Queue and will not be delivered immediately. This is particularly useful if the destination domain is an ETRN SMTP domain. Dialup SMTP hosts connect to the Internet intermittently, and an attempt to deliver messages to such hosts when they are not connected to the Internet will usually fail.

  b. There is a temporary DNS error during the domain name resolution process.

  c. A destination host is found, but the SMTP cannot establish a connection.

  d. The destination SMTP server issues a temporary SMTP response code.

  e. The SMTP connection is aborted prematurely due to network problems.

  f. The destination SMTP server did not reply within the configured time.

  Messages in the Deferred Queue are further grouped into different SMTP domain channels using the information in the recipient addresses. This allows server side ETRN support and prevents the deferred messages from delaying the processing of new messages. Like the Pending Queue, the Deferred Queue is also equipped with Deferred Queue Processors which process the messages

on a per channel basis. During each scheduled queue run time, one or more deferred queue processors are created for every SMTPC domain channel by the SMTPC to handle deferred outgoing messages. Messages for each SMTPC domain channel are processed according to their message priority weight.

After the priority weight for each message is determined and the messages are arranged according to their priorities, SMTPC will attempt to deliver the first message from each SMTPC channel. If the first message in the SMTPC channel is delivered successfully, the Deferred Queue Processor will create another thread to deliver the subsequent messages in the channel. If the delivery attempt for the first message failed, the subsequent messages in the entire channel will remain queued. This approach increases the efficiency of the system by eliminating unnecessary message delivery attempts.

Message Priority Handling

SMTPC features a mechanism for message priority handling which guarantees high throughput and the orderly handling of messages with different priorities. This mechanism basically allows the SMTPC to assign a priority weight for each message based on three factors:

- The pre-defined message precedence
- The message size
- The total deferred time

The message priority weight is calculated using the formula:

```
Priority Weight = (precedence*Mp)+(size*Ms)+(deferred_ time*Md)
```

where:
`Mp` is the precedence multiplier;
`Ms` is the size multiplier; and
`Md` is the deferred time multiplier

The priority weight is an integer value. The lower the priority weight, the higher the priority level and the sooner the message is processed. The message precedence and size multiplier are configurable parameters that the system administrator can define, whereas the deferred time multiplier is system generated since it denotes how long the message has been in the Deferred Queue. A message with a longer total deferred time is given higher priority over the messages that arrived recently.

Mail Routing Handling

SMTPC uses several criteria when routing Internet messages. The routing options are:

- DNS host name lookup
  Used to locate and translate an Internet domain name into an IP address. A domain name is a meaningful and easy-to-remember “handle” for an Internet address (i.e., domain.com).

- Host Table lookup of destination host
  An internal host table, usually a text file, is used by the SMTPC to determine the IP address of the recipient host. The exact format of the host table depends

upon the TCP implementation. The location of the host table should be specified upon the installation of Internet Exchange.

- DNS then Host Table lookup
  In the event of a failure to locate and translate an Internet domain name into an IP address, Internet Exchange performs a Host Table lookup to determine the IP address of the recipient host.

- Host Table then DNS lookup
  When the IP address of the recipient host cannot be determined, Internet Exchange performs a DNS lookup to locate and translate an Internet domain name into an IP address.

- Delivery to Default Mail Relay Host
  When the Internet Exchange is configured to use a default mail relay host, messages are sent to a primary mail forwarder for further routing. If this mail forwarder cannot be contacted for some reason and the secondary mail relay host is defined, the machine uses the secondary mail relay host. In this case, it will occasionally check to see if and when it is possible to switch back to use the primary relay host.

Internal Database Storage

SMTPC uses several databases to store messages and peer information. The MESG.BTR is used to store the envelope, priority value and status information of messages. The CHANNEL.BTR is used to store the status information for the SMTP Domain channel. The PEER.BTR is used to store the SMTP Domain Profile configuration information, such as queue run interval, queue run size, maxSMTP sessions, maxMsgPerSession and retryPeriod for each peer domain. Another database, the DNS.BTR is used to store the resolved DNS information.

Conclusion

To communicate well with the Internet, Internet Exchange implements SMTP as two separate modules. SMTPC delivers messages to the Internet. It regularly polls for messages queued in the SMTPOUT channel. The SMTPC Queue Manager guarantees fast mail delivery by processing messages based on their priority weight and by assigning different processors for deferred and pending messages. SMTPD, meanwhile, listens for incoming messages from the Internet. A new thread is created by the SMTPD module whenever a new connection request for incoming mail is detected. The newly created thread is responsible for receiving the messages intended for Internet Exchange.

A PDF copy of each component can be downloaded separately at the following URLs:

- http://www.ima.com/product/v5/bsmtp/bsmtp.pdf
- http://www.ima.com/product/v5/ccmail/ccmail.pdf
- http://www.ima.com/product/v5/dirserver/ldap.pdf
- http://www.ima.com/product/v5/dlmanager/dlmgr.pdf
- http://www.ima.com/product/v5/notes/notes.pdf
- http://www.ima.com/product/v5/message_store/mstore.pdf
- http://www.ima.com/product/v5/migtools/migratn.pdf
- http://www.ima.com/product/v5/mta/ie4mta.pdf
- http://www.ima.com/product/v5/preprocessor.pdf
- http://www.ima.com/product/v5/smtp/smtp.pdf

Information about the anti-spam, anti-virus module, quota agent and web mail client can be downloaded separately at the following URLs:

- http://www.ima.com/product/v5/antispam/aspam.pdf
- http://www.ima.com/product/v5/antivirus/avirus.pdf
- http://www.ima.com/product/v5/quota/quota.pdf
- http://www.ima.com/product/v5/wmc/wmc.pdf

HONG KONG
27/F, China Resources Building
26 Harbour Road
Wan Chai, Hong Kong
Tel.: +852-2520-0300
Fax: +852-2648-5913

PHILIPPINES
6/F, Hanston Building
Emerald Avenue,Ortigas Center
Pasig City, Philippines
Tel.: +63 (2) 637-9090
Fax: +63 (2) 637-9898

USA
Toll Free No.: +1 (800) 549-2762
Fax: +1 (800) 549-2762

INTERNATIONAL MESSAGING ASSOCIATES