Internet Exchange 4

Messaging Server

Administrator's Guide

Version 1.0
July 1999

ISBN: 962-8137-03-5

International Messaging Associates Limited
Hong Kong Computer Center, 20/F
54-62 Lockhart Rd.
Wanchai
HONG KONG

Tel:+1 (408) 481-9985
+1 (888) 562-3564
+852 2520-0300
+63 (2) 811-3999
Fax:+1 (888) 562-3561
+852 2648-5913
+63 (2) 811-3939
Email:*info@ima.com*
WWW:*http://www.ima.com*

IMA Philippines, Inc.
The Peak Tower, 15/F
107 Alfaro Street
Salcedo Village, Makati
PHILIPPINES

USA - Sunnyvale, California
USA - Message Center
Hong Kong
Philippines - Makati
USA
Hong Kong
Philippines - Makati

The following are copyrights of their respective companies or organizations:

Portions of this product are based on software developed by the following universities/organizations:

Table of Contents

Preface

Overview

THE MESSAGING SERVER ADMINISTRATOR'S GUIDE

The Internet Exchange Messaging Server (IEMS) is responsible for sending and receiving messages over the Internet via the SMTP protocol or the Batch SMTP tunnel. Other Internet Exchange modules, such as the Message Store and the cc:Mail and Notes Connectors, connect to the IEMS to send and receive mail and to make use of the IEMS' useful features, including the anti-virus and anti-spam tools.

To provide the system administrator with a well-defined tool for using, configuring, and managing the IEMS, this manual is organized as follows:

Part 1, *"System Architecture"*, provides an overview of the technologies used in the IEMS together with a complete diagram showing how the other Internet Exchange module connects to the IEMS. It also outlines the different components of the IEMS and their features.

Part 2, *"Installation"*, describes in detail the steps that must be followed by the user in installing and setting up the IEMS.

Part 3, *"Administration and Operation"*, describes the procedures for configuring the many features of the IEMS and managing its operations. It also includes a section on error handling.

Part 4, *"Troubleshooting"*, describes the tools needed by the system administrator for troubleshooting purposes.

Part 5, *"Appendix"*, provides a detailed description of every technology and standard used by the Internet Exchange Message Store.

PART 1

System Architecture

Messaging Server Architecture

INTRODUCTION

The **Internet Exchange 4** Messaging Server is an open architecture, stand-alone messaging system whose design is specifically tailored to enable legacy email systems to coexist with proprietary messaging systems while making full use of Internet messaging and directory services. The Messaging Server is responsible for the sending and receiving of messages over the Internet using either the SMTP protocol or a Batch SMTP tunnel. Once messages are received by Internet Exchange, the Message Transfer Agent (MTA) then routes the messages to the appropriate output channel (SMTP, Message Store, Distribution List Manager, cc:Mail/Notes Connector, etc.) and performs preset preprocessing on each message, after which the message routing is facilitated by data in the Internet Exchange Directory Server.

SYSTEM ARCHITECTURE

Internet Exchange 4 is an open-enterprise messaging server made up of several easy-to-configure modules. Each module has specific functions that assure fast, secure, and reliable message delivery to and from the Internet.

- Preprocessor Unit

  Anti-spam Engine
  Anti-virus Engine
  Channel Action Matrix

- Message Switch
- IMAP4 optimized message store

  IMAP4 Server
  POP3 Server
  Mailsort

- LDAP-based Directory Service and Synchronization
- Distribution List Manager
- SMTPD (Simple Mail Transfer Protocol - Daemon)
- SMTPC (Simple Mail Transfer Protocol - Client)
- Batch SMTP Tunnel

Following is a description of the above mentioned system components of the Messaging Server:

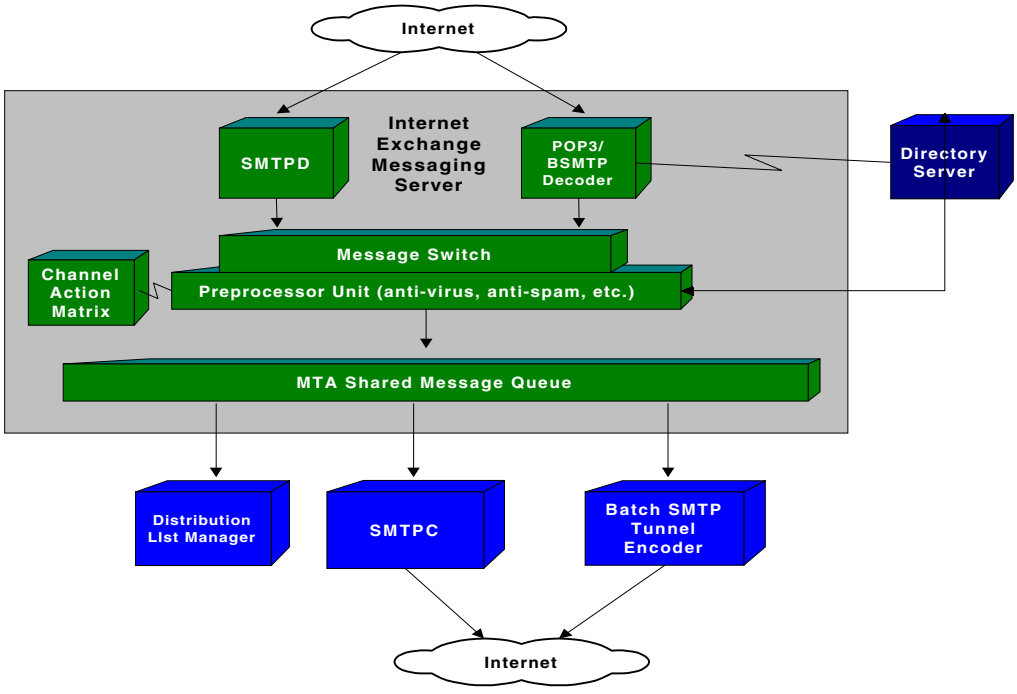

Figure 1. System Architecture of Internet Exchange 4.0

THE PREPROCESSOR UNIT

Internet Exchange 4's Preprocessor Unit is an integrated subsystem of the Message Transfer Agent (MTA), which features a highly scalable architecture. Each of the Preprocessor programs are plug-in modules which can be run on separate machines, ensuring efficient utilization of computing resources and maximum throughput. In addition, this capability ensures that the system can easily be scaled to cope with the changes in the messaging needs of an organization. The Preprocessor subsystem incorporates an open API allowing for the development of third party or custom processing modules.

Anti-virus Module

For speedy processing of incoming mail, **Internet Exchange 4's** Anti-virus Module is designed as a multi-threaded application capable of creating multiple threads for simultaneous virus scans. Each thread will process one message at a time. The module will decode the attachment(s) and then invoke an external virus scan engine. If a virus is found in a message, the engine can either bounce the mail, copy the mail to a predefined quarantine location/folder, or delete the mail (with the option to notify the postmaster after deletion) as configured by the user.

Anti-spam Module

The Anti-spam Module is a stand-alone unit that provides the system administrator with options to create a list of unwelcome IP addresses/address ranges or a list of banned IP

addresses/address ranges using simple Graphical User Interfaces (GUI's). It is also capable of verifying the corresponding name of an IP address during the initial stage of the SMTP session via reverse DNS lookup to filter out forged names, thereby blocking out potential spammers even before they can enter the system. Moreover, the Anti-spam Module supports Real-time Blackhole Lists (RBLs) for optimum anti-spam protection.

Auto Text Insertion Engine

The Auto text insertion engine is another Preprocessor DLL which provides the capability to insert disclaimer messages into messages passing through Internet Exchange Message System (IEMS). The Administrator can add different disclaimer messages based on the message source channel. The Auto text engine allows the system administrator to use simple text and/or HTML text for insertion process. The engine currently supports normal RFC822 message (non-MIME) and most of the MIME message structure types.

The Preprocessor invokes the insertion engine based on the configuration in the Channel Action Matrix. The Auto insertion engine provides a second level of configuration where users can define a simple text file name and/or a HTML text file name per source channel. The content of the simple text file and/or the HTML text file is appended to the appropriate section of the message.

Channel Action Matrix

Internet Exchange 4 provides a Channel Action Matrix for each module in the Preprocessor Unit. With the Channel Action Matrix, system administrators are provided with a flexible tool for configuring which modules in the Preprocessor Unit should be run for a particular message based upon message flow through the IEMS.

THE MESSAGE SWITCH

The **Internet Exchange 4's** Message Switch connects disparate email systems. It is capable of supporting multiple mail systems, including the following channels: Local Message Store, cc:Mail Connector, Lotus Notes Connector, Distribution List Manager, outbound SMTP, and the Batch SMTP Tunnel Encoder. The Message Switch is LDAP enabled and relies on user information provided by the Directory Service to determine which channel(s) an incoming message should be routed to. It enhances the scalability of **Internet Exchange 4** by allowing new functions and channels to be added arbitrarily to the system.

IMAP4 OPTIMIZED MESSAGE STORE

Internet Exchange 4's IMAP4 Optimized Message Store is a dedicated mail repository for remotely storing, retrieving, and manipulating messages. It has been optimized for access via the IMAP4 remote access protocol, allowing many users to access the system concurrently. Unlike other IMAP4 based servers, Internet Exchange 4 gracefully handles large mailbox sizes efficiently.

IMAP4 Server

Internet Exchange 4's IMAP4 Server allows users to access their mailboxes via IMAP4-capable clients such as Microsoft Outlook Express, Netscape Communicator, and others. By utilizing IMAP4, users can manipulate their mailboxes/folders on the server without having to download them to a local hard disk. They can also create multilevel mailboxes on the

server that can be can easily renamed, moved, or deleted by them (with the proper authorization from the system administrator), as well as shared mailboxes which can be viewed concurrently in real time from multiple platforms.

POP3 Server

Internet Exchange 4's POP3 Server provides POP3 capable clients with another means for accessing their incoming mailboxes. Using POP3, users retrieve messages from the Internet Exchange Message Store inbox and store them in a local hard disk so they can be read in an off-line or disconnected state. The POP3 Server supports multithreading for fast message delivery.

Mailsort

Internet Exchange 4's features Mailsort for defining rules so that the local mail delivery agent can direct messages to preselected mailboxes/folders other than the INBOX and generate automatic replies to incoming messages based on predefined criteria. The Mailsort filtering utility implements these rules, enabling users to process incoming mail in the background based on certain attributes, such as the message sender and subject, on the Internet Exchange Server at message delivery time.

LDAP-BASED DIRECTORY SERVICE AND SYNCHRONIZATION

Internet Exchange 4's Directory Service is based on a client/server architecture that uses LDAP (Lightweight Directory Access Protocol), an open directory access protocol running over TCP/IP. It is specifically designed for managing information about users, groups, mailing lists, aliases processing, and mail routing. Other Internet Exchange modules, such as the IMAP4 Server, POP3 Server, Message Switch, and Local Mail Delivery Agent (LMDA), access the Directory Service for directory information.

DISTRIBUTION LIST MANAGER

Distribution lists allow messages to be sent to all of the list's subscribers simply by submitting the messages to a single address. To maximize the potential of distribution lists, the Internet Exchange 4 Distribution List Manager allows users to create Internet electronic mailing lists that support the following features: mail blocking, adding/removing subscribers, accepting/rejecting subscriber applications, deciding who can post messages on the list, and choosing message delivery options. Configuration of all distribution list features, like other Version 4 modules, is Web-based.

SMTPD

SMTPD is the server module listening for incoming messages on the Internet. Whenever a new connection request for incoming mail is detected, the SMTPD Module creates a new thread that manages the new connection. Internet Exchange 4's SMTPD Module is capable of sustaining simultaneous SMTP connections by creating multiple threads, thereby minimizing delay in message delivery. It also supports the ESMTP service extension DSN (Delivery Status Notification), as well as the 8BITMIME, the message size extensions, and ETRN for downstream dialup connected sites.

SMTPC

SMTPC is the module responsible for delivering messages to the Internet. It is a multi-threaded application that regularly polls for messages queued in the SMTP OUT queues. 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. The SMTPC Module supports the following ESMTP service extensions: DSN, message size extension, ETRN remote queue start-up (primarily for disconnected or dial-up access), and 8BITMIME. With ETRN support, dialup SMTP hosts can notify the SMTP server when to deliver messages, allowing bandwidth resources to be used efficiently.

BATCH SMTP TUNNEL

A common problem encountered in using POP3 as a message transmission technology is that the original message recipient addresses are often discarded during the transfer of the message to its intended recipient(s). This has been addressed in **Internet Exchange 4** through the use of an innovative Batch SMTP (SMTP Tunnel Encoder/Decoder) Module.

The Batch SMTP Tunnel Encoder provides a mechanism for the tunneling of messages for an entire organization or predefined Internet addresses, while preserving the original envelope information for each message. When messages of this type arrives at a single POP3 account, they are picked up by the POP3 Batch SMTP Decoder, which decodes and then submits them to the Internet Exchange MTA with the original envelope recipients retained. This allows the messages to be further routed until they are received by the originally intended recipients.

System Components

SIMPLE MAIL TRANSPORT PROTOCOL MODULE

The Simple Mail Transport Protocol (SMTP) is the protocol used to transport electronic mail on the Internet. In addition to transporting mail between and among directly connected Internet sites, SMTP can also be used as a common mail backbone between organizations that use other types of mail systems, with the intermediate transport performed using Internet mail relays. **Internet Exchange 4** uses SMTP to send and receive messages over the Internet. SMTP is closely associated with a specific message format, defined by RFC822, which is the basic mail message type used over the Internet. Until the arrival of MIME, non-structured RFC822 messages were the only standard message type carried by SMTP.

SMTPD (SIMPLE MAIL TRANSFER PROTOCOL DAEMON)

SMTPD is a background server process, which runs continuously to listen for incoming messages from the Internet. Whenever new connection requests for incoming mail are detected, SMTPD creates a new thread that manages that connection. It is capable of creating multiple threads for simultaneous processing of multiple messages, thereby minimizing delay in message delivery.

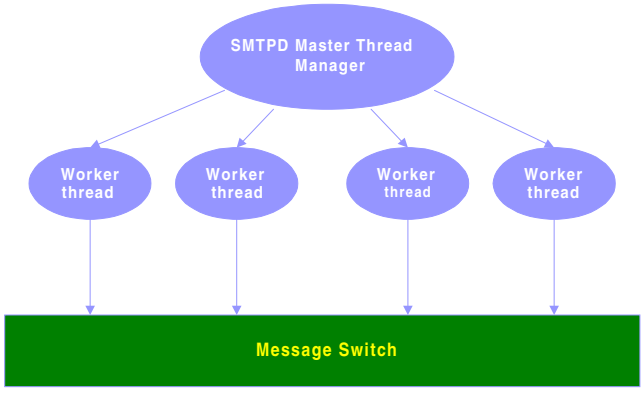

Figure 2a. SMTPD System Architecture

SMTPD is designed to support the ESMTP service extension DSN (Delivery Status Notification), as well as the 8BITMIME, MESSAGE SIZE, and ETRN extensions for downstream dialup connected sites. It features a multithreaded model to achieve high performance and can process multiple SMTP connections simultaneously. Once a message is received by the worker thread, it is submitted to the Internet Exchange MTA Shared Queue. SMTPD also performs several anti-spam checks at the SMTP level before the message enters the Internet Exchange Shared Message Queue.

Multithreaded Architecture

In order to achieve optimal high performance, SMTPD features a multithreaded architecture. This multithreaded architecture allows it to support concurrent, multiple SMTP connections. The Master Thread Manager is responsible for listening to the SMTP port and waiting for incoming SMTP requests from other SMTP MTA's. Once an SMTP connection request is received, the Master Thread Manager creates a new SMTP worker thread to handle that SMTP connection. The number of simultaneous SMTP connections is limited only by system resources such as the TCP stack and memory.

SMTPD supports the following features:

- ESMTP Support

  SIZE (Message Size Declaration)
  ETRN (Remote Message Queue Starting)
  8BITMIME (8bit-MIMEtransport)
  DSN

- Anti-Spam Defense

  SMTP connection restriction
  Mail Relay Authorization
  Reverse DNS lookup verification
  Real-time Blackhole List (RBL) support

SMTPC (SIMPLE MAIL TRANSFER PROTOCOL CLIENT)

SMTPC is responsible for delivering messages to the Internet via the SMTP protocol. It supports the ESMTP service extensions SIZE, 8BITMIME, ETRN and DSN. For fast message delivery, **Internet Exchange 4's** SMTPC Module features an efficient queuing strategy that supports two types of independent queues: the Pending Queue and the Deferred Queue. It also provides a mechanism for message priority handling based on the calculated message priority weight. To achieve high scalability and performance, SMTPC incorporates a hierarchical multithreaded architecture as shown in Figure 2b. The SMTPC Queue Router is responsible for retrieving messages from the Internet Exchange Shared Queue and transferring them to the internal message queue via the Mail Queue Switch, while SMTPC Master Queue Manager is responsible for controlling and synchronizing the Pending Queue Processors and the Deferred Queue Processors.

Each machine running SMTPC is capable of maintaining multiple simultaneous outbound SMTP connections. **Internet Exchange 4's** SMTPC Module features an innovative approach to queue management that supports server-side ETRN requests and provides a mechanism for message priority handling. This architecture guarantees not only high throughput, but also the orderly handling of messages of different priorities. The module comes with the SMTPC Queue Router, which retrieves outgoing messages from the Message Switch and determines whether they should be routed to the Pending Queue or to the Deferred Queue. A shared message queue structure is designed for these queues, so as to achieve efficient usage of system memory. Each queue can have one or more queue pro-

cessors active at a time, each of which will further create multiple SMTPC worker threads to process multiple simultaneous outbound messages and send them to their next destination across the Internet.

Pending Queue

Newly arrived messages that must be sent out immediately are placed in the Pending Queue (see Figure 2c). These messages are then processed by the Pending Queue Processors, which attempt delivery via SMTP. If the delivery of a message in the Pending Queue is unsuccessful, it is passed on to the Deferred Queue so it can be delivered at a later time. Messages that are destined for intermittently connected hosts with ETRN support, such as dialup accounts, completely bypass the Pending Queue and are sent directly to the Deferred Queue.

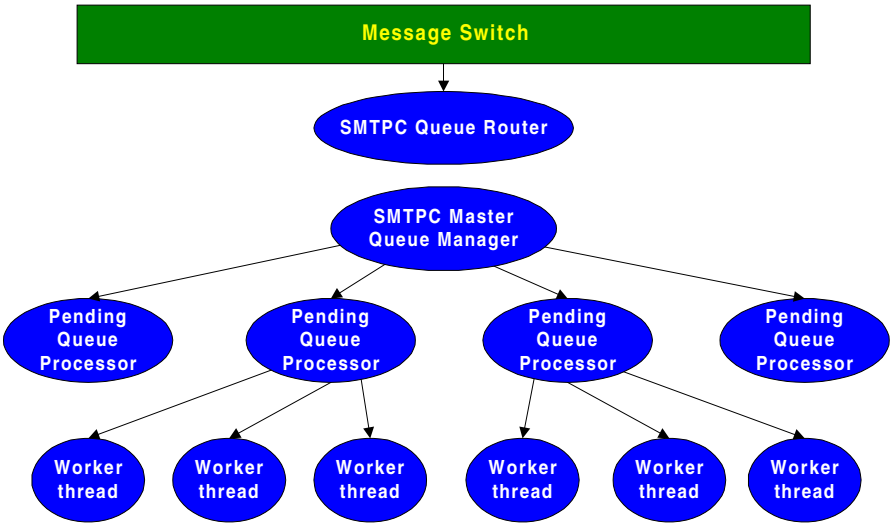

Figure 2b. SMTPC System Architecture

The queue run interval for each Pending Queue Processor can be configured by the system administrator. In addition, the administrator can define the maximum number of Pending Queue Processors that run concurrently and the number of messages to be processed by each processor during each queue run. Each Pending Queue Processor is capable of creating multiple threads for handling multiple SMTP sessions.

Deferred Queue

Messages that are intentionally deferred or whose previous delivery attempt(s) have failed are placed in the Deferred Queue. Messages in the Deferred Queue are further grouped into different SMTP domain channels using information in the recipient addresses. This allows server-side ETRN support and prevents deferred messages from delaying the processing of new messages. A message is placed in the Deferred Queue if any of the following conditions is encountered:

- The destination domain is a predefined ETRN SMTP domain. Dialup SMTP hosts connect to the Internet intermittently, and attempting to deliver messages to such

hosts when they are not connected to the Internet will fail.

- There is a temporary DNS error during the domain name resolution process.
- A destination host is found but SMTP connection cannot be established.
- The destination SMTP server issues a temporary SMTP response code.
- The SMTP connection is aborted prematurely due to network problems.
- The destination SMTP server did not reply within the configured time.

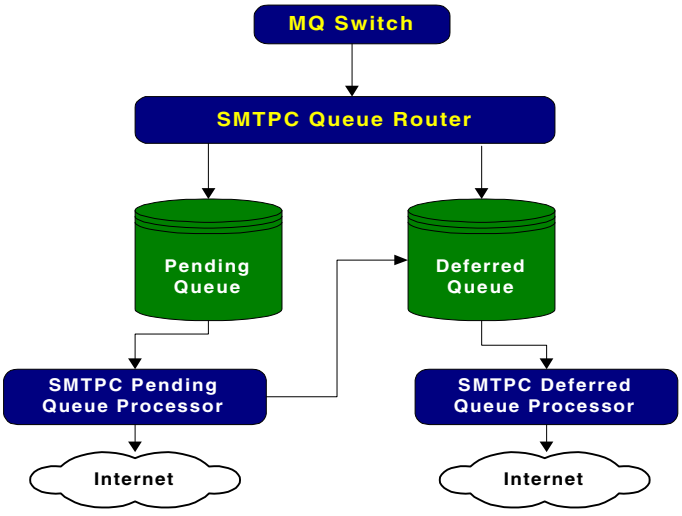

Figure 2c. SMTPC Queue Router

Messages in the Deferred Queue are processed by the Deferred Queue Processors on a per channel basis. During each scheduled queue run time, one or more Deferred Queue Processors are created for every SMTP domain channel by the SMTPC Module to handle deferred outgoing messages. The messages for each SMTP domain channel are processed according to their message priority weight. SMTPC will attempt to deliver the first message from each SMTP Domain Channel. If the delivery attempt is successful, the Queue Processor will create other child SMTPC threads to deliver subsequent messages. Otherwise, all subsequent messages in the entire channel will remain queued. This approach greatly improves the overall efficiency of resource usage by eliminating unnecessary message delivery attempts.

It is advisable to queue all the messages for a particular domain (such as ETRN domains) before attempting delivery. When an ETRN host is connected, it makes an ETRN request to SMTPD, which notifies SMTPC. SMTPC then starts a Deferred Queue Processor to deliver all queued messages for this domain immediately. Since the messages for this domain are already grouped, this approach ensures less processing time and fast delivery.

Message Priority Handling

SMTPC assigns a priority weight to each message based upon three factors, namely:

- the predefined message precedence
- the message size
- the total deferred time (for messages in the Deferred Queue)

The message priority weight is calculated using the following formula:

$$\begin{aligned} \textit{Priority weight} \quad = \quad & \textit{(precedence * Mp)} \\ & \textit{+ (size * Ms)} \\ & \textit{+ (deferred\_time * Md)} \end{aligned}$$

where Mp is the precedence multiplier, Ms is the size multiplier, and Md is the 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 is a configurable parameter that is defined by the system administrator. The message size is also a configurable parameter that provides the system administrator with a mechanism for preventing large messages from delaying the delivery of urgent but smaller messages. The total deferred time (for messages in the Deferred Queue), on the other hand, represents the time a message has been stored in the Deferred Queue. A message with a longer total deferred time is given a higher priority level than those that arrived recently. This parameter is also configured by the system administrator.

Mail Routing Handling

SMTPC is capable of routing Internet messages based on several criteria. The routing options are:

- Domain Name System (DNS) host name lookup
- Host Table lookup of destination host
- DNS followed by Host Table lookup
- Host Table followed by DNS lookup
- Delivery to default mail relay host(s)

Mail routing via the DNS is the preferred method for routing messages on the Internet. The DNS is an Internet service that provides for the storage and retrieval of information associated with domain names and routing information. In the context of Internet mail, the records that are of interest are the mail exchanger (MX) records and address (A) records.

MX records are used to store mail forwarder information for hosts registered on the Internet. An MX record contains the name of the host or domain, and a list of one or more mail forwarding hosts as well as the preference values associated with these hosts. The preference values are used by SMTPC to determine the order in which to attempt delivery in case more than one mail forwarder is identified. MX records are essential for the proper routing of mail, especially in situations where the destination host is not physically connected to the Internet and has to rely upon a mail forwarder for mail delivery. As an exam-

ple, some organizations rely upon the UUCP communications package which comes with the UNIX operating system to physically exchange mail. These sites can, by using MX-records, appear to be connected to the Internet even though mail is the only Internet service they use.

A records, on the other hand, are used to store IP address information for hosts. When configured to use the DNS, SMTPC obtains an MX record for the destination host. If an MX record is found, the list of mail forwarding hosts is used during SMTP connection. If no MX record is found, SMTPC searches for an A record. If an A record is found, then this address is used when the SMTP connection is established.

If SMTPC is configured to use host table lookup, the internal host table, which is usually a text file, is used to determine the IP address of the recipient host. The exact format and path name of the host table depends upon the TCP implementation. The location of the host table is specified when SMTPC is installed. This is the equivalent of doing an A record lookup using the DNS. Most internal host tables, however, cannot provide complete databases, unlike the DNS.

When configured to use a default mail relay host, all messages are sent to a primary mail forwarder for further routing. If this mail forwarder cannot be contacted for any reason and a secondary mail relay host is defined, Internet Exchange uses the secondary mail relay. In this case, it occasionally checks if and when it is possible to switch back to use the primary relay host. Use of this option improves server throughput, as mail forwarding hosts are usually on the same network as the message server. Response time and throughput are typically fast, resulting in minimal or zero backlog of messages at the server. The use of this option, however, places the burden of routing and retries of delayed messages on the mail forwarding machine(s), which will add to their existing workloads.

Internet Exchange 4's SMTPC Module can be configured to use a combination of the above strategies to deliver mail. When not using an email relay, it is recommended to use a strategy where the DNS is consulted first, and then a local host table, in case the attempt to resolve a name with the DNS fails. The opposite configuration can also be used if needed. In any event, if the name cannot be resolved using either of the above methods, SMTPC will fall back to using the mail relay host(s) as the next hop (assuming there is at least one configured), in the hope that resolution can be better handled at that site.

Internal Database Storage

Internet Exchange 4 uses several databases to store message 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 the queue run interval, queue run size, maxSMTPSessions, maxMesgPerSession, and retryPeriod for each peer domain.

Another database, the DNS.BTR, is used to store the resolved DNS information for caching purposes as to speed up the MX and A record lookup process. The maximum number of records in this database is configurable.

BATCH SMTP ENCODER/DECODER

The Internet Exchange Version 4 Messaging Server features the Batch SMTP Tunnel Encoder and Decoder, which support the tunneling of Internet email across non-SMTP message transports. The Batch SMTP Encoder also allows the encoding or tunneling of mail directed to a single address or a complete domain to a predefined Internet address where the proper decoding or detunneling takes place. This destination address can be on an Internet Exchange Messaging Server or on any other server with an RFC2442 compliant decoder installed. The Batch SMTP Decoder works together with the POP3 Client module to pick up remote messages using the POP3 protocol. It then detunnels the messages by reinjecting them into the Internet Exchange Messaging Server input queue.

Principle of Operation

Batch SMTP is a batch mode implementation of any SMTP/ESMTP transactions that support the Batch SMTP Media Type (RFC 2442). This is a MIME content type that is used to tunnel/encapsulate SMTP/ESMTP transactions through any MIME-capable transport. This type can be used in a variety of purposes: extending end-to-end MIME-based security services to cover message envelope information as well as message content, and enabling the transfer of multiple separate messages in a single transactional unit.

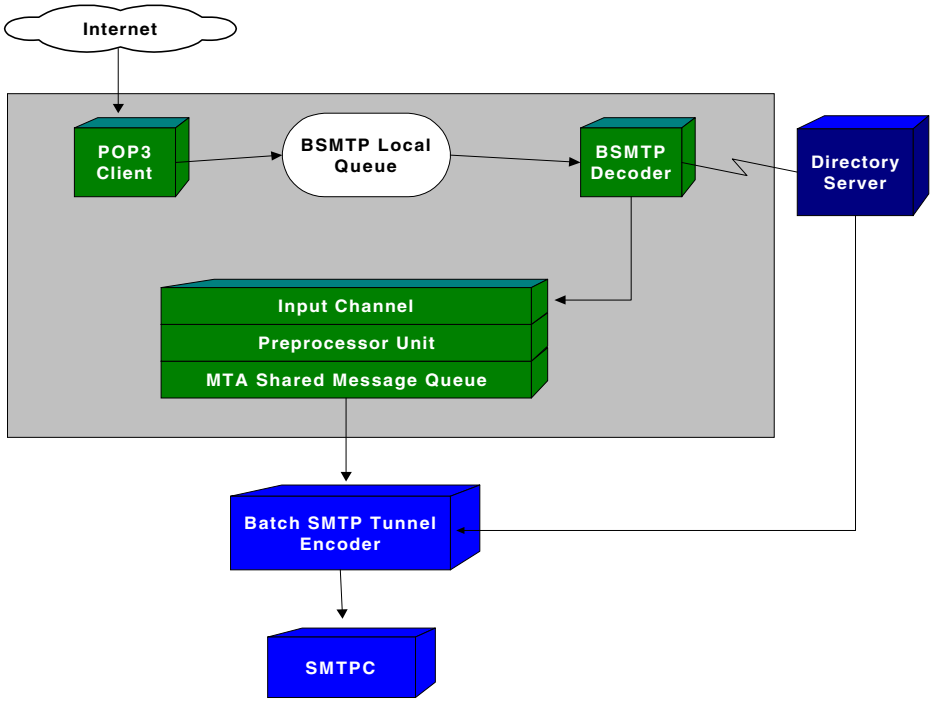

Figure 2d. Message flow for Batch SMTP

The **Internet Exchange 4** BSMTP Module's BSMTP Decoder retrieves Batch SMTP messages from a remote host using POP3 Client access. It checks the MIME content type headers in the messages to ensure that they are labeled as an Application/Batch SMTP. A BSMTP local message queue is used in order to handle messages passing between the POP3 Client and the BSMTP Decoder. The BSMTP Processor/Decoder then retrieves the

messages from the local queue and processes the messages as defined in RFC2442 (the Batch SMTP media type).

The Batch SMTP Tunnel Encoder supports the tunneling of messages for an entire organization or predefined addresses, while preserving the original envelope or delivery information for each message. It converts a conventional SMTP message into an application/Batch SMTP object, which is then encapsulated in a new Internet message with a destination address capable of decoding the tunneled message and performing further delivery. After reaching its destination, the object is converted back into a conventional SMTP message by a Batch SMTP Decoder.

PREPROCESSOR UNIT

Internet Exchange 4's Preprocessor Unit is an integrated subsystem of the MTA with a highly scalable architecture. Each of the unit's programs is a plug-in module that can be run on separate machines, ensuring efficient utilization of computing resources and maximum throughput. This capability also guarantees that the system can easily be scaled to cope with the changes in the messaging needs of an organization. In addition, the Preprocessor Unit incorporates an open API that permits the development of third-party or custom processing modules. Communication between the different the Preprocessor Unit's modules is carried out via Remote Procedure Calls (RPC) over TCP/IP.

Anti-spam Module

Internet Exchange's Anti-spam Module is a stand-alone unit that provides the system administrator with options to create a list of banned IP addresses/address ranges or a list of allowed IP addresses/address ranges using simple GUI's. It also has the capability to verify the corresponding name of an IP address during the initial stage of the SMTP session via reverse DNS lookup to filter out forged names. This blocks out potential spammers even before they can enter the system.

In addition, the Anti-spam Module features Real-time Blackhole List (RBL) support. An RBL is a blacklist of IP addresses that have been confirmed to send spam mail, be friendly to spammers, and/or totally open to mail relaying. It makes use of the Domain Name System (DNS) to distribute databases of blacklisted IP addresses. **Internet Exchange 4** supports an arbitary number of RBL-style systems, though at present if enabled, five RBL style systems are supported:

- MAPS-RBL (Mail Abuse Preventions System's Real-time Blackhole List)
- ORBS (Open Relay Behavior-modification System)
- MAPS-DUL (Mail Abuse Prevention System's Dial-up user List)
- Internet Mail Relay Services Survey (IMRSS)
- DynamicIP Spam Sources List (DSSL)

MAPS-RBL is a system that creates intentional network outages so that the transport of unwanted mass email is prevented. ORBS is a database that lists SMTP servers that have been confirmed to permit third-party relay. MAPS-DUL, on the other hand, lists dial-up and other dynamically assigned IP addresses to prevent trespassing by people and/or orga-

nizations who send unsolicited email using direct connections to their victims' mail servers without using their ISP's mail server as a relay or gateway.

The Preprocessor Unit's Anti-spam Module is configured by the system administrator using the Channel Action Matrix.

Anti-virus Module

Internet Exchange 4's Anti-virus Module is a 32-bit multithreaded, stand-alone preprocessing module capable of performing concurrent virus scanning for MIME and non-MIME message attachments. To minimize delay in message delivery, the Anti-virus Module is designed to create multiple threads for performing simultaneous virus scans, with each thread processing one message at a time. For each thread, the module determines what decoding method to use on each attachment based on the MIME headers of the MIME/RFC822 message. If the attachment is embedded in a non-RFC822 message, either UUDECODE or BINHEX is used. After decoding the attachment, an external virus scan engine is invoked by the module. If a virus is found in a message, the engine either bounces the mail, copies the mail to a predefined location/folder, or deletes the mail (with the option to notify the postmaster after deletion) as configured by the user. After all attachments have been scanned, the module returns the appropriate error code.

Like the Anti-spam Module, the Anti-virus Module is configured via the Channel Action Matrix.

AutoText Insertion Engine

The AutoText insertion engine is another Preprocessor DLL that provides the capability to insert disclaimers into messages passing through the Internet Exchange Message System (IEMS). The Administrator can add different disclaimer messages based on the message source channel. The AutoText engine allows the system administrator to use simple text and/or HTML text for insertion process. It currently supports insertion into normal RFC822 messages (non-MIME) and most of the MIME message structures.

The Preprocessor invokes this module based on the configuration in the Channel Action Matrix. The insertion engine provides a second level of configuration where users can define a simple text file name and/or a HTML text file per source channel. The content of the simple text file and/or the HTML text file is inserted into the appropriate section in the email message.

KEY FEATURES

- *Different disclaimer messages based on the channel source*
  The administrator can configure the AutoText insertion engine to add different disclaimer messages based on the source channel name. With this feature, it is possible that messages generated in cc:Mail environment will have a different disclaimer from those that came from Lotus Notes environment.

- *Using simple text or HTML text*
  The administrator can define simple plain text file and/or an HTML version of the disclaimer for the insertion engine.

- *Support for various message types*
  The AutoText insertion engine supports the following message types:
  - Normal RFC822 messages
  - Single part MIME messages
  - Multipart/mixed MIME messages
  - Multipart/alternative MIME messages

Channel Action Matrix

Internet Exchange 4 provides a Channel Action Matrix for each module in the Preprocessor Unit. With the Channel Action Matrix, system administrators have a flexible tool for configuring which modules in the Preprocessor Unit should run for a particular message, based upon message flow or routing within the MTA. For example, to minimize delay in message delivery, the system administrator may not want to run the Anti-virus for messages coming from a cc:Mail user and destined to another cc:Mail user or to a Lotus Notes user within the system. Or he may want to run the Anti-virus Module only for messages coming from the Internet and not for messages bound for the Internet. These options are easily configured in the appropriate Channel Action Matrix.

DISTRIBUTION LIST MANAGER

Internet Exchange 4's Distribution List (DL) Manager allows messages to be sent to all of a list's subscribers simply by submitting the said messages to a single address. The DL Manager also enables the system administrator/list owner to create Internet electronic mailing lists that support the following features: mail blocking, adding and removing subscribers, and setting the preferred delivery options. These features are configured by the system administrator using a Web-based interface.

Message Flow

When mail arrives at the Message Switch, **Internet Exchange 4** consults the Directory Server to determine whether there are messages destined for a mailing list. If such a messages are found, they are routed to the Distribution Lists 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 all the list's members. After the mailing list's members are identified, the message is forwarded to the Preprocessor Unit where appropriate actions (i.e. anti-virus scans, etc.) 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 channel(s) (i.e. SMTPC, cc:Mail, Local Message Store, etc.) for final delivery to all of the mailing list's members.

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 and unsubscriptions, which would normally be the list maintainer's responsibilities.

Archiving is another task that performed by the engine. The engine can keep a copy of

every message received by a mailing list. The archived messages are stored in the home directory of the DL Manager. 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 engine runs continuously, checking the appropriate channels for new mail, thereby ensuring minimum delay in mail delivery.

Web-based Interface

The Web-based interface is used by the system administrator for configuring the DL Manager's various functions, such as mailing list creation, addition/deletion of a member to/from a particular list, and setting of the preferred delivery option for each member.

Mailing List Categories

Internet Exchange 4's DL Manager supports two types of mailing lists, the open and closed types.

Open distribution lists

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

Closed distribution lists

Unlike an open mailing list, a closed mailing list is accessible only to the list's members. Only those people who subscribed to the list can post messages and/or access the list's 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 requires the approval of the list owner/system administrator or the recommendation of a current member of the list being subscribed to.

Delivery Modes

Through the DL Manager, the system administrator is provided with the means to set the mode of delivery on a per user basis.

Immediate mode of delivery

When messages are posted to the list, the DL Manager immediately sends out the messages to the mailing list subscribers who have selected this delivery mode.

Digest mode of delivery

In the digest mode, messages are allowed to accumulate for a certain time before they are delivered simultaneously to the list members.

Archiving

If archiving is enabled, the DL Manager creates a copy of every message posted to a particular list. The archived messages are stored in a subfolder under the mailing list folder.

DIRECTORY SERVER

Internet Exchange 4's Directory Server is based on the open Internet directory standard LDAP (Lightweight Directory Access Protocol). LDAP is a protocol designed to provide read/write access to open X.500 directory service and proprietary directories that support the X.500 standard without incurring the hefty resource requirements of its predecessor, the DAP or Directory Access Protocol. Unlike the DAP, the LDAP does not require the upper layers of the OSI protocol stack and runs directly on TCP/IP or other reliable transport protocols.

Internet Exchange 4's Directory Server allows the system administrator to manipulate stored information via the Web interface. This provides the system administrator with a user/administrator interface to the Directory Server's front-end engine. The Web Interface uses the LDAP API to access the Directory Server and to update or modify information contained in the directory. By using the Web interface, the system administrator can perform the following functions:

- add new entries
- delete existing entries
- search for a particular entry
- modify existing entries
- start or stop the Directory Server

The Internet Exchange Directory Server consists of two major subsystems: the front-end protocol engine and the back-end database engine. The front-end protocol engine receives requests from LDAP clients 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 the bind, unbind, search, modify, modify RDN, 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 the front-end engine via a well-defined API. The slapd back-end API (SLAPI) consists of twelve commands, nine of which correspond to the LDAP protocol operations. The other three commands are for initializing the back-end engine, shutting down the back-end engine, and handling back-end specific configurations.

Directory data storage

Internet Exchange 4 provides a default directory schema for email applications. The directory data includes user account information, group information, and mail routing information. The user account information consists of the unique user id (Mail Address), user password, mail address, and other user-related profiles. The group information consists of data on users that have the same access rights to the same directory. General information, like the email address and user name, can be accessed by an LDAP client. Access to sensitive information, such as password and confidential user profiles, is restricted by an authentication mechanism.

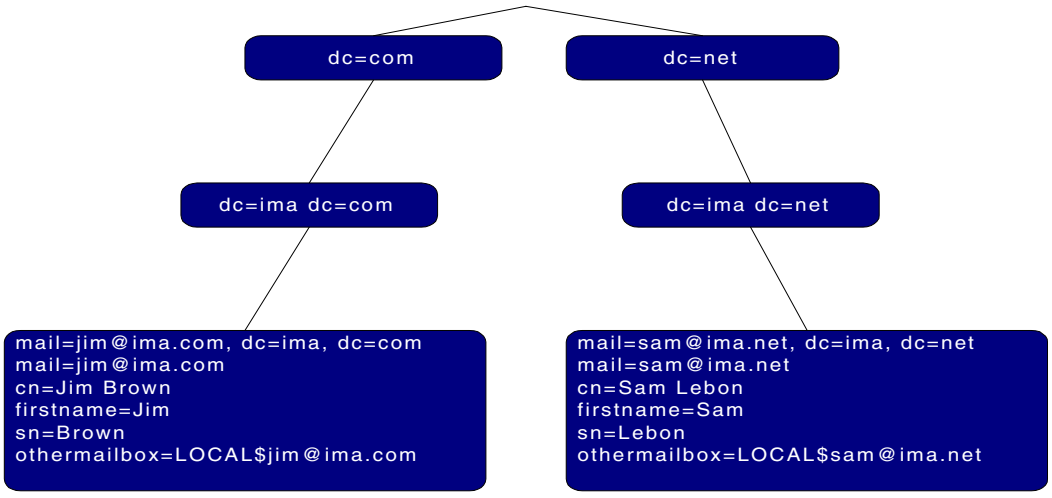

Figure 2e. The Directory Information Tree (DIT)

Directory information tree

Directory entries in the Directory Server are organized using a directory information tree (DIT). The root of the DIT is represented by a special entry whose Distinguished Name is called the directory suffix. **Internet Exchange 4** has a completely new LDAP design, which is based on the recommendations in RFC 2377. This recommendation proposes a LDAP directory structure based on the domain part of a users email address. **Internet Exchange version 4** uses the 'mail' and 'dc' component to construct the LDAP tree.

MESSAGE SWITCH

Internet Exchange 4's Message Switch routes incoming messages to one of several available outbound messaging channels, including the following:

- cc:Mail Connector
- Lotus Notes Connector
- Batch SMTP Tunnel Encoder
- Local Message Store
- Distribution List Manager
- Outbound SMTP

The Message Switch is LDAP-enabled and relies on user information provided by the Directory Service to determine which channel an incoming message should be routed to. The Message Switch enhances the scalability of **Internet Exchange 4** by allowing new functions and channels to be added arbitrarily to the system. Each new channel is able to register itself with the Message Switch as a new message queue. Messages are then delivered to this queue based on the configuration in the LDAP Directory Service.

Key Features

OPTIMIZED QUEUE MANAGEMENT

For fast and efficient delivery, SMTPC supports two types of queues, namely the Pending Queue and the Deferred Queue. All outgoing SMTP messages are logically assigned to either of these queues. They are then processed by the Pending Queue Processors and Deferred Queue Processors. Messages that must be sent out immediately are assigned to the Pending Queue, while messages that are intentionally deferred (ETRN hosts) or whose previous delivery attempt(s) failed are assigned to the Deferred Queue.

Deferred messages are divider further into groups by the SMTP domain channel. The first message from each SMTP Domain Channel is sent out every queue run. If the delivery attempt is successful, the Queue Processor creates other child SMTPC threads to deliver the other messages. Otherwise, all the subsequent messages in the entire channel remains queued. Efficient storage usage is achieved by incorporating a Shared Message Queue mechanism for these queues.

SMTP DOMAIN PROFILING

Each SMTP domain channel is handled by an independent channel processor. As such, SMTP Profiling can be applied to each particular SMTP domain. The message handling parameters for each domain include Message Queued Before Attempt, Queue Run interval, Retry Period, maximum number of SMTP connection, and maximum number of message sent in the SMTP connection.

MESSAGE PRIORITY HANDLING

SMTP supports a mechanism for message priority handling. Each message is assigned a priority weigh based on the following factors: the predefined message precedence, the message size and the total deferred time (for messages in the Deferred Queue). Messages in the queues are processed according to the message priority level.

EFFICIENT SERVER-SIDE ETRN SUPPORT

The design of **Internet Exchange 4's** SMTPC module allows efficient server-side ETRN support for dial-up hosts.

HIGH SCALABILITY

The SMTPC module supports a hierarchical architecture based on a multithreading model, making it highly scalable. It is capable of invoking a configurable number of concurrent

Pending Queue Processors and Deferred Queue Processors, which creates more SMTP worker threads. Its scalability is only limited by system resources.

EXTENSIVE ROUTING OPTIONS

SMTPC supports the following routing options:

- Domain Name System (DNS) host name lookup
- Host Table lookup of destination host
- DNS followed by Host Table lookup
- Host Table followed by DNS lookup
- Delivery to default mail relay host(s)

AUTOMATIC MAILING LIST SUBSCRIPTION

When the DL Manager receives a subscription request it first checks the type of list the sender is trying to subscribe to. If it is an open list, the DL Manager activates automatic subscription and adds the email address of the sender to the mailing list. A message is then sent to the new member informing him/her that the subscription request has been approved. Upon receipt of the confirmation message back from the subscriber, they will be automatically added to the requested list.

AUTOMATIC MAILING LIST UNSUBSCRIPTION

The DL Manager handles unsubscription requests 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.

MAILING LIST SUBSCRIPTION VERIFICATION

The DL Manager handles all the subscription requests by checking the type mailing list being subscribed to. If it is an open list, the subscription sender is automatically added to the mailing list. If it is a closed list, the DL Manager passes the subscription request to the list maintainer. It is the list maintainer's duty to verify the authenticity of the subscription request. The list maintainer must send an email to the potential subscriber for verification purposes. If the potential subscriber replies to the email sent by the list maintainer, then his/her email address is added the mailing list via a Web-based interface.

FLEXIBLE MESSAGE DELIVERY TO MAILING LISTS

Internet Exchange's DL Manager offers two modes of delivery: the immediate and digest modes of delivery. This is to optimize message handling and provide electronic list members with options for handling messages.

Immediate Mode of Delivery

When messages are posted to a mailing list, the DL Manager sends them immediately to the mailing list's subscribers who have selected this delivery mode.

Digest Mode of Delivery

In the digest mode, messages are allowed to accumulate for a certain period of time before they are delivered their recipients. The time/day of delivery is configured in the DL Manager via a Web-based interface.

MAIL BLOCKING FOR MAILING LISTS

This feature allows the list owner to prevent certain mailing list members from posting messages on the list. This is useful in discouraging list members from posting messages that may be deemed offensive by the other members. Blocked members, however, are still able to receive messages posted by the other members.

DIAL-UP SCHEDULER

The most common configuration when using **Internet Exchange 4** is to have a permanent connection to the Internet. However, in certain cases, it may be impossible or impractical to maintain permanent to Internet connection. In such cases, it is desirable to be able to schedule Internet Exchange to dial up to an ISP at a particular time of the day to download and upload messages from and to the Internet.

The purpose of the Dial-up Scheduler, is to allow the system administrator to configure automatic RAS dial-up scheduling via a Web-based interface.

Remote Access Service

Remote Access Service (RAS) is the service by which the Windows operating system allows the local system to dial and connect to another peer over the Internet. For Windows 95/98, RAS is more commonly known as Dial-up Networking, but for Windows NT 3.5x and 4.0, this function has been introduced as Remote Access Service.

The Dial-up Scheduler is used to schedule the start-up of **Internet Exchange 4's** SMTPC, SMTPD, and BSMTP/POP3 modules so as to activate RAS dial-up. Its automatic shut-down feature enables these modules to shut down automatically before the start of the next dial-up schedule.

The Dial-up Scheduler offers two scheduling options: periodic scheduling and fixed-time scheduling. When the scheduled dial-up time comes, the Dial-up Scheduler performs the RAS dial-up. When the scheduled hang-up time comes, it shuts down all TCP-level components, such as SMTPD, SMTPC, and POP3D and then closes down the RAS dial-up connection. The Dial-up Scheduler is implemented as a separate component. It is provided with a Web-based interface for configuring dial-up schedules and profiles.

PART 2

Installation

System Requirements

HARDWARE / SOFTWARE BASE CONFIGURATION

For optimum performance, it is recommended that **Internet Exchange 4** and its components be run using the following minimum configurations:

Windows 95/98

- Pentium or higher
- Minimum recommended RAM: 64 MB
- Minimum recommended hard disk space for applications: 40 MB
- Minimum recommended hard disk space for message storage: 1GB

Windows NT 4.0 Server

- Pentium or higher
- Minimum recommended RAM: 96 MB
- Minimum recommended hard disk space for applications: 40 MB
- Minimum recommended hard disk space for message storage: 1GB

NOTE: For Windows 95, Service Pack 1 and Kernel32 must be installed on the system. For Windows NT 4 Server, Service Pack 4 must be installed on the system. To enable Japanese language support, Japanese Windows OS is required.

INTERNET EXCHANGE 4 COMPONENTS

Internet Exchange 4 consists of the following modules, which are in turn divided into several components.

Message Transfer Agent (MTA)

The MTA consists of the following components:

- SMTP Daemon (SMTPD)
- SMTP Client (SMTPC)
- MQ Router
- LDAP Server
- Distribution List Manager
- Preprocessor
- Btrieve Database Engine
- Anti-virus Module
- Anti-spam Module
- Auto-insertion Engine
- Auto-loop Detection DLL
- Administrative Tools
- Responder

- Web Server

IMAP4 Optimized Message Store

The Message Store consists of the following components:

- Message Store Server
- Local Mail Server
- Local Mail Delivery Agent (LMDA)
- IMAP4 Daemon
- POP3 Daemon

cc:Mail Connector

The cc:Mail consists of the following components:

- CCIN
- CCOUT

Notes Connector

The Notes consists of the following components:

- NOTESIN
- NOTESOUT

MEMORY USAGE

The base hardware/software configuration is only for running the machine's OS and other software needed by the OS to run **Internet Exchange 4** properly. To determine the minimum memory requirement needed by your machine to run the OS and the **Internet Exchange 4** modules installed on the machine, you must add the memory requirements of those modules to the base hardware configuration. Use Table 3a for reference to compute for the minimum memory requirements of your machine.

For example, if you have a machine running Windows 95/98, you need a minimum of 64MB of RAM to run the OS. If you wish to install the Messaging Server on that machine, then you will have to install additional RAM of 6MB for SMTPD, 4MB for SMTPC, 2MB for the MQ Router, 4MB for the LDAP Server, 4MB for the Distribution List Manager, 8MB for the Preprocessor, 4MB for the Btrieve Database Engine, 4MB for the Anti-virus Module, 2MB for the Responder, 2MB for the Web Server, 2MB for the Auto-loop Detection DLL, 2MB for the Anti-spam Module, 2MB for the Auto-insertion Utility, and 8MB for the Administrative Tools (as shown in Table 3a). Thus, the machine needs at least 118MB of RAM in order for the Messaging Server to run smoothly.

Internet Exchange 4 Modules	Memory Usage (MB)
CCIN	8
CCOUT	8
NOTESIN	8
NOTESOUT	8
SMTP Daemon (SMTPD)	6
SMTP Client (SMTPC)	4
MQ Router	2
LDAP Server	4
Local Mail Delivery Agent	4
Local Mail Server	2
Distribution List Manager	4
Message Store Server	2
IMAP4 Daemon	6
POP3 Daemon	4
Preprocessor	8
Btrieve Database Engine	4
Anti-virus Module	4
Responder	2
Web Server	2
Anti-spam Module	2
Auto-loop Detection DLL	2
Administrative Tools	8

Table 3a - Minimum memory requirements of Internet Exchange 4 components

Installing the Messaging Server

INTERNET EXCHANGE WORKSHEET

Before running the **Internet Exchange 4** installation program, it is necessary to gather all the information needed for the installation. To simplify the installation of **Internet Exchange 4**, please review and fill out the installation worksheet on the following page before installing the software. Each item in the worksheet is discussed in the following sections.

Common Parameters

This section of the installation worksheet identify the parameters that are associated with either the installation and/or the overall operation of the gateway.

Certificate Location

The new licensing system used for the current release of **Internet Exchange 4** uses certificates to store the licensing information. Upon registration, a certificate is issued and the location of this file should be noted during installation of the licenses.

Program Directory

The default location in which **Internet Exchange 4** is installed is c:\Program Files\Ima\Internet Exchange4.0. This is where the programs and libraries reside. The directory can be located anywhere. However it is strongly recommended that it be placed on a local hard drive for optimum performance and reliability.

Temporary Directory

The directory in which **Internet Exchange 4** will build messages. It is usually configured to be a subdirectory of the queue directory, but it can be set to a different directory and/or drive depending on the local disk availability.

Installation Worksheet

Common Parameters

Certificate Location _____
Program Directory _____
Temporary Directory _____
Local Character Set _____
Local Time Zone _____

TCP/IP

Host Name _____
Domain Name _____
Host Table Filename Location _____
DNS Server List _____
Mail Relay Host Name _____

MTA Parameters

LDAP Server _____
Preprocessor Host name _____
Local Domains _____
Default Local Delivery Channel _____
Internet Delivery Channel _____
Message Queue Server _____
MQ Server Access Mask _____
Message Queue Local Directory _____
MQ Remote Access Directory _____
MQ Server Account Name _____
MQ Server Password _____

Message Store Parameters

LDAP Server _____
Message Store Hostname _____
Message Store Port _____
Root Directory of User Mailbox _____

cc:Mail Connector

Local Post Office Name _____
Internet Post Office Name _____
Post Office Path _____
Post Office Password _____
Post Office Administrator _____

Notes Connector

Notes Mail Server _____
Local Notes Mail Domain _____
User/Server ID File in Use _____
Password for the ID file _____
Lotus Notes Server Administrator _____
Local Internet Hostname _____
Local Internet Domain _____
SMTP Domain Name _____

Local Character Set

The ISO character set to be used. Most Anglo-Saxon countries can select US-ASCII, while others may choose a different character set. All outgoing email will be tagged using the selected character set.

Local Time Zone

The time zone that covers the location of the machine running **Internet Exchange 4**. Whether this time zone uses daylight saving or not should also be noted. There are several locations configured in the system, including the USA, much of Europe, and Asia. If the local time zone is not listed, it will have to be entered manually into IEMTA.INI with an editor as follows:

[Gateway]
Timezone=*tzn*[[+ | -]]*hh*[[:*mm*[[:*ss*]]]][[*dzn*]]

The *tzn* is a three-letter time-zone name, such as PST, followed by an optionally signed number, *hh*, which gives the difference in hours between UCT and local time. To specify the exact local time, the hours can be followed by minutes, :*mm*; seconds, *ss*; and a three-letter daylight-saving-time zone, *dzn*, such as PDT. Separate hours, minutes, and seconds with colons (:). If daylight saving time is not in effect, set *Timezone* without a value for *dzn*. If the *Timezone* value is not currently set, the default is PST8PDT, which corresponds to the Pacific time zone of the USA.

If the time zone "Use system TZ variable" is selected, the timezone information will then be obtained from the user defined TZ environment variable. Under Windows 95, this can be set in the *autoexec.bat* system startup file. Under Windows NT, it is usually set in the system registry. In either case, the machine must be rebooted in order to make the change effective.

TCP/IP Parameters

This section identifies the parameters associated with the local TCP/IP network.

Host Name

Each host on the Internet must have a unique identifier so that email bound for that site has a single unambiguous destination. This identifier is known as the Fully Qualified Domain Name (FQDN). The host name parameter is the name component of the machine FQDN. For example, if the FQDN of the gateway is *iegate.jade.net*, then the hostname would simply be *iegate.*

Domain Name

The domain component of the machine FQDN. For example, if the FQDN of the gateway is *iegate.jade.net*, then the domain component would be *jade.net*.

Host Table Filename Location

This is the full path name of the TCP/IP host file. Even if the Domain Name System (DNS) is used for host name-to-address translations, a host file be that contains addresses for the loopback, gateway machine, and your mail relay host, is strongly recommended.

MTA Parameters

This section of the installation worksheet identifies the parameters used in setting up the components of the MTA module and the delivery channels.

LDAP Server
The FQDN of the machine running the LDAP server.

Preprocessor Host name
The FQDN of the machine that runs the Preprocessor Unit. The Preprocessor Unit is designed to run on a separate machine.

Local Domains
Domains names that will be treated as “local” by the MTA.

Default Local Delivery Channel
The default channel where messages should be delivered, e.g.

local

Internet Delivery Channel
This refers to the channel for outgoing messages, e.g.

smtpc

Message Queue Server
The FQDN of the machine where the Message Queue is residing.

MQ Server Access Mask
The range of IP addresses that the MTA recognizes as “local”.

Message Queue Local Directory
The actual location of the Message Queue on the machine where it is installed, e.g.

c:\msgqueue

MQ Remote Access Directory
The directory by where the message queue can be accessed remotely, e.g.

*cuenca**msgqueue*

MQ Server Account Name
The account name used to access the Message Queue.

MQ Server Password
The password used for the MQ Server account name.

Message Store Parameters

This section identifies the parameters used in setting up the Message Store servers and databases.

LDAP Server

The FQDN of the machine running the LDAP Server.

Message Store Hostname

The FQDN of the machine that runs the Message Store Server (*msgstors.exe*). The Message Store databases should also be located on this machine. The Message Store server must be running in order for the LDAP Server's Web-based interface to create a local user remotely. The Message Store Server is also used by cc:Mail and Notes Connectors' migration tools to create local users.

Message Store Port

This specifies the port number where the Message Store server is listening. This should also appear in the IEMTA.INI file of the server and client applications. This parameter is needed when running the Message Store Server for migrating user address books and mailboxes and for creating local users from LDAP web interface. The default value is 8000.

Root Directory of User Mailbox

This refers to the directory where MsgStore is located. This is used by the migration tools for creating a default user home directory in the local Message Store.

cc:Mail Connector Parameters (If applicable)

This section of the installation worksheet identifies the parameters associated with the local cc:Mail Post Office. This is the Post Office that queues messages on behalf of Internet Exchange (*see the cc:Mail connector guide for additional information*).

Local Post Office Name

The name of the cc:Mail Post Office that **Internet Exchange 4** will log on to retrieve messages.

Internet Post Office Name

The name which **Internet Exchange 4** uses to log on to the cc:Mail Post Office. This name must exist in the cc:Mail directory, and must be defined as a Post Office. Although any unique name may be used here, it is recommended that *Internet* be used for clarity.

Post Office Path

The path name for the directory where the local cc:Mail Post Office resides.

Post Office Password

The password that Internet Exchange uses to log on to the local cc:Mail Post Office.

Post Office Administrator

Internet mail standards require each site to have a mail account that receives messages

addressed to “postmaster”. The postmaster typically receives notices about mail problems, network problems, and inquiries about users and mailboxes. This parameter should be the cc:Mail address of the person managing Internet Exchange.

Notes Connector Parameters (if applicable)

This section identifies the parameters used in setting up the exchange of messages between Lotus Notes and **Internet Exchange 4**.

Notes Mail Server

This is the name of the Notes Server that will be accessed by Internet Exchange, e.g.

Kintak/Jade

Local Notes Mail Domain

This is the local Notes Mail domain name that Internet Exchange connects to. This is created automatically in the server during the installation of the Notes Server, e.g.

Jade

User/Server ID File in Use

This is the name of the ID file specified in the entry “*KeyFileName*=” of IEMTA.INI. Internet Exchange uses the owner of this ID file to access SMTP.BOX and MAIL.BOX in the Notes Server. Normally this file is created in the data directory of the Lotus Notes Server/Workstation during the installation process. If KeyFileName does not contain any path information, Internet Exchange automatically appends the data directory specified in “*Directory*=” from IEMTA.INI as the prefix, e.g.

c:\notes\data\user.id	for normal Notes Workstation Installation
c:\notes\data\server.id	for normal Notes Server Installation

Password for the ID file

This is the password for the User/Server ID file. The password appears as a row of asterisks (****) for security reasons.

Lotus Notes Server Administrator

This is the user name for the gateway administrator. By default, Internet Exchange uses the owner of the User/Server ID file as the name of the gateway administrator. Also, this is the alias for the Internet Postmaster. All Internet sites are required to support the Postmaster alias. When Internet Exchange receives a message addressed to Postmaster, it is sent to this Notes user, e.g.

postmaster/Crop.

Local Internet Hostname

The Internet hostname for the local machine that runs Internet Exchange. This is the leftmost part of the fully qualified domain name for this host, and should not contain dots, e.g.

iegate

Local Internet Domain

The Internet Domain for the local machine that runs Internet Exchange. This is the remaining part of the FQDN for this host after the local Internet hostname is removed, e.g.

jade.net

SMTP Domain Name

This is the domain name of the machine to which Internet Exchange connects. Select the correct entry to enable concurrent multiple SMTP server operations for a single Notes environment, e.g.

ieln

INSTALLATION PROCEDURE

In order to install the **Internet Exchange Messaging Server,** insert your **Internet Exchange 4** CD in your CD-ROM drive and run the installer *Setup.exe*. This program is designed both to be installed and removed from the system should the need arise.

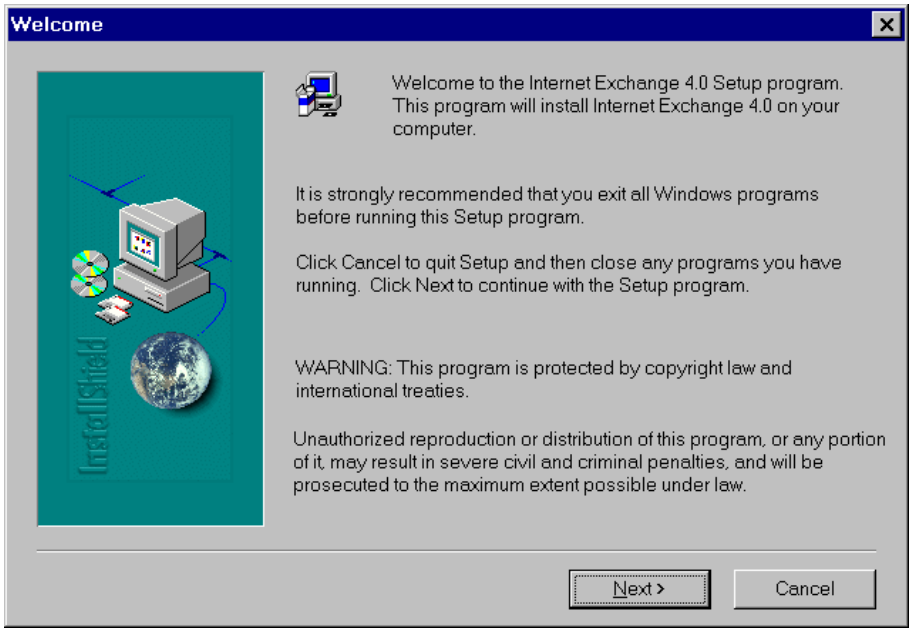

Figure 5a - Welcome

Clicking on the *Next* button of the initial dialog box indicates acceptance of the software usage terms stated in the screen. A new dialog box (see Figure 5b) that allows the system administrator to choose the folder or directory for storing the Internet Exchange executable files will appear.

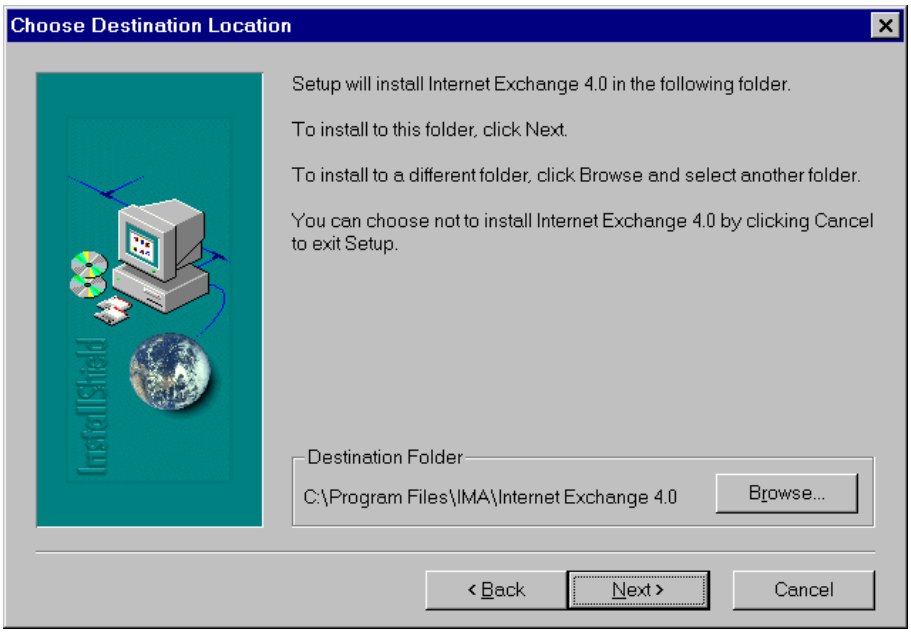

Figure 5b - Choose Destination Folder

After selecting the folder for **Internet Exchange 4**, click on the *Next* button. The next screen (see Figure 5c) allows the system administrator to select which Internet Exchange components to install on the machine that currently runs the installation program. To install the Messaging Server only, select *MTA*. You may then install the other components on other machines. Or you can install all Internet Exchange components on the current machine by checking them all. After choosing which modules to install, click on the *Next* button.

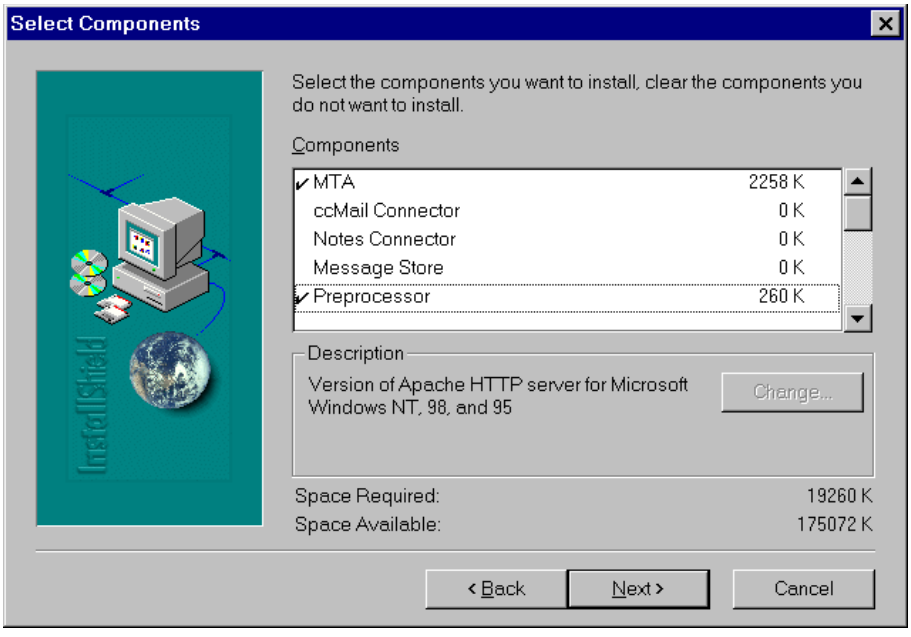

Figure 5c - Select Components

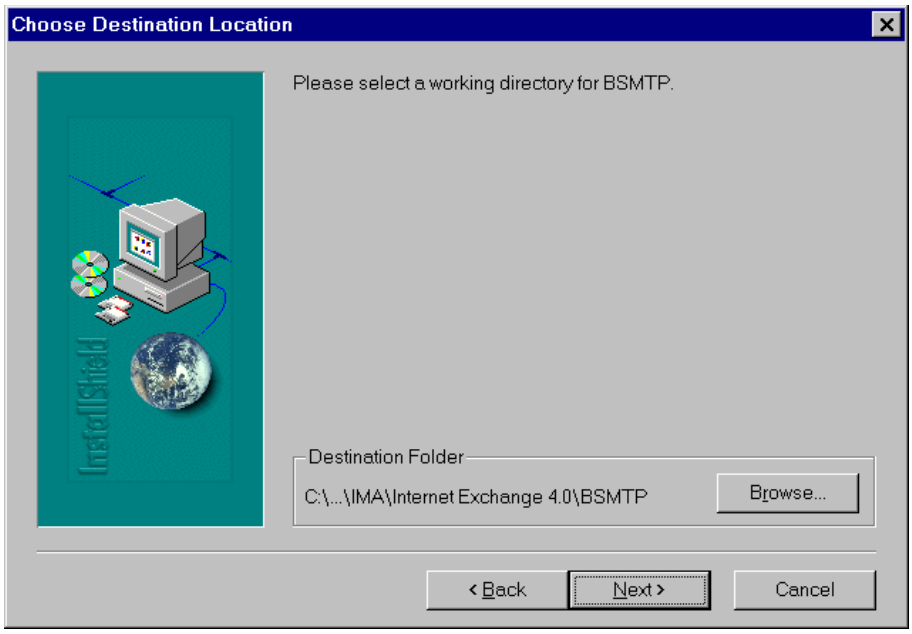

Figure 5d - Select BSMTP Directory

The next screen (see Figure 5d) is for selecting/creating the working directory for the Batch SMTP Module. By default, the BSMTP Module working directory is created as the BSMTP subdirectory under the executable directory. After selecting/creating a directory for the BSMTP Module, click on the *Next* button to continue.

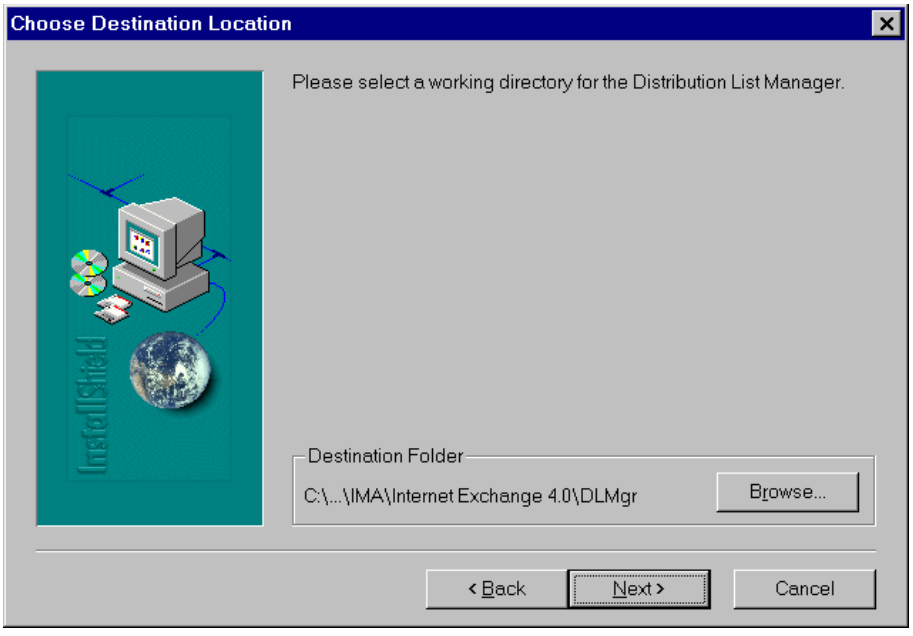

Figure 5e - Select DL Manager Directory

This screen (see Figure 5e) enables the system administrator to select the working directory for the Distribution List Manager. By default, the Distribution List Manager working

directory is created as the DLMgr subdirectory under the executable directory. After selecting/creating a directory for the Distribution List Manager, click on the *Next* button to go to the next screen.

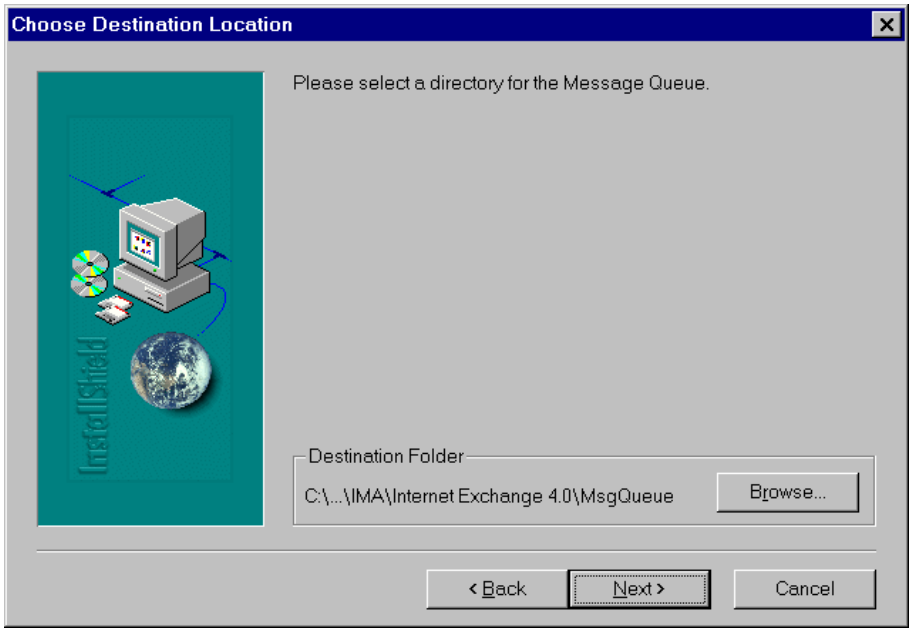

Figure 5f - Select Message Queue Directory

This dialog box (see Figure 5f) enables the system administrator to select/create the working directory for the Message Queue. By default, the Message Queue will reside in the MsgQueue subdirectory under the executable directory. After selecting/creating a working directory for the Message Queue, click on the *Next* button to continue.

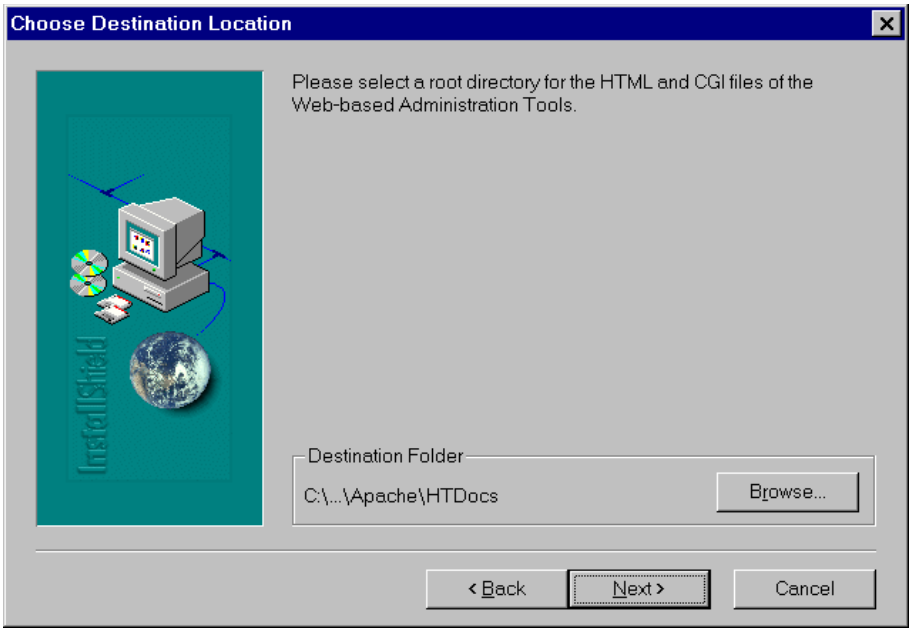

Figure 5g - Select HTML Pages and CGI Script Directory

The next dialog box (see Figure 5g) enables the system administrator to choose/create a root directory for the HTML pages and CGI scripts that comprise the Web-based Administration Tools. By default, web pages and scripts are transferred to the WebAdmin subdirectory under the executable directory. Click on the *Next* button to continue.

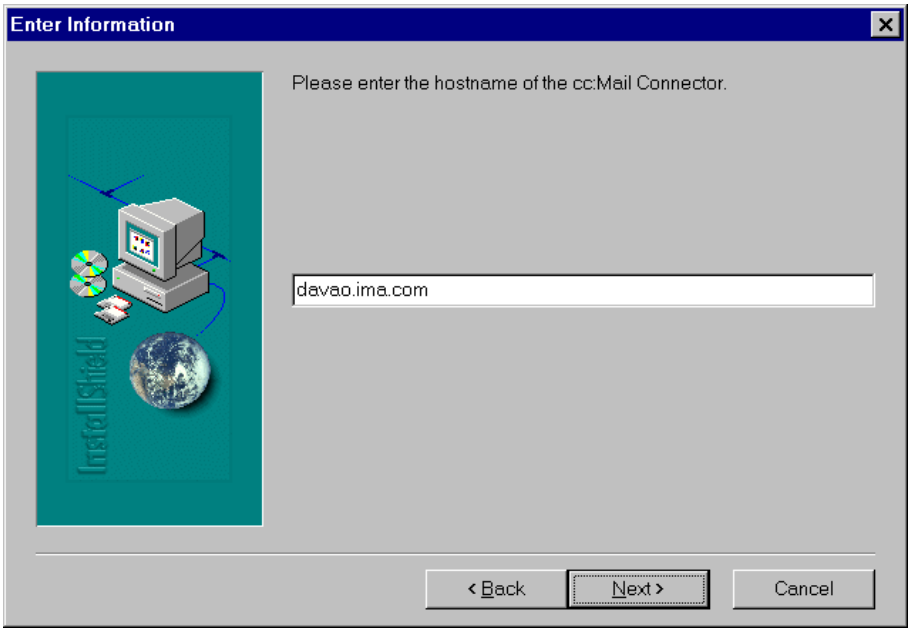

Figure 5h - Screen for specifying cc:Mail Connector host name

The next screen (see Figure 5h) will ask for the host name of the machine that will run the cc:Mail Connector. You may leave this blank if you do not wish to install the cc:Mail Connector. Or you may specify “localhost” to indicate that the cc:Mail Connector resides (or will be installed) on the same machine. Click on the *Next* button to go to the next screen.

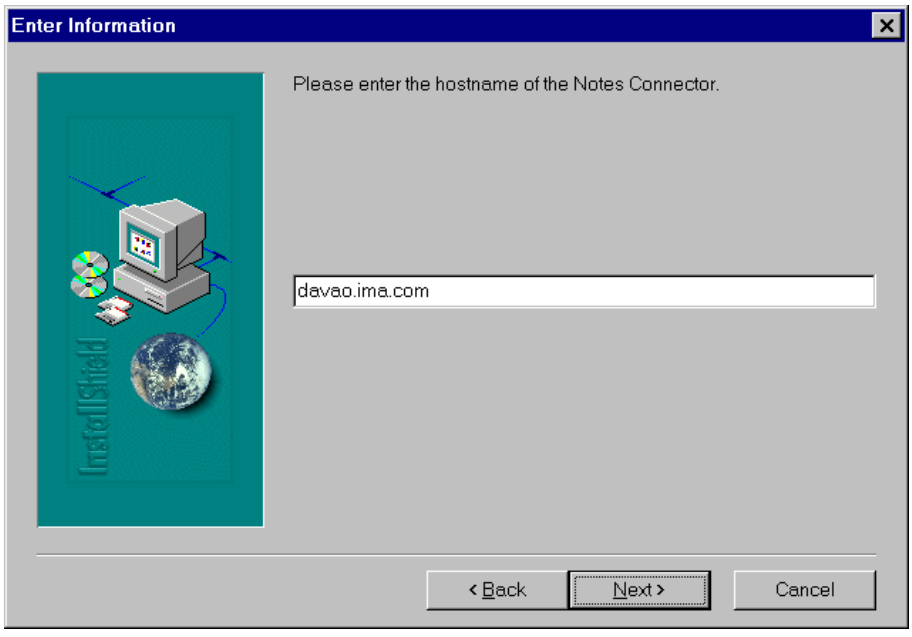

Figure 5i - Screen for specifying Notes Connector host name

The next screen (see Figure 5i) will ask for the host name of the machine that will run the Notes Connector. You may leave this blank if you do not wish to install the Notes Connector. Or you may specify “localhost” to indicate that the Notes Connector resides (or will be installed) on the same machine. Click on the *Next* button to go to the next screen.

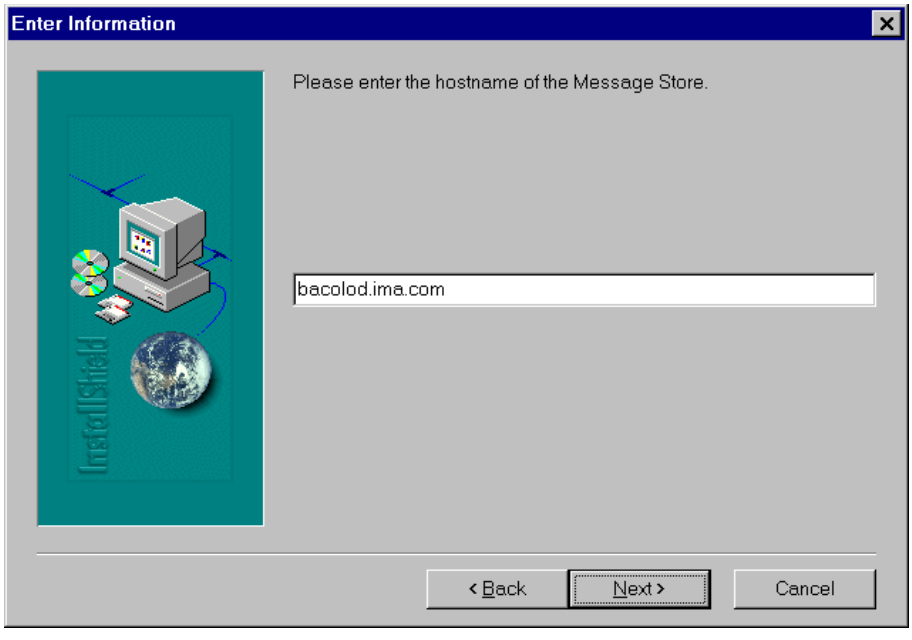

Figure 5j - Screen for specifying Message Store host name

The next screen (see Figure 5j) will ask for the host name of the machine that will run the Message Store. You may leave this blank if you do not wish to install the Message Store.

Or you may specify “localhost” to indicate that the Message Store resides (or will be installed) on the same machine. Click on the *Next* button to go to the next screen.

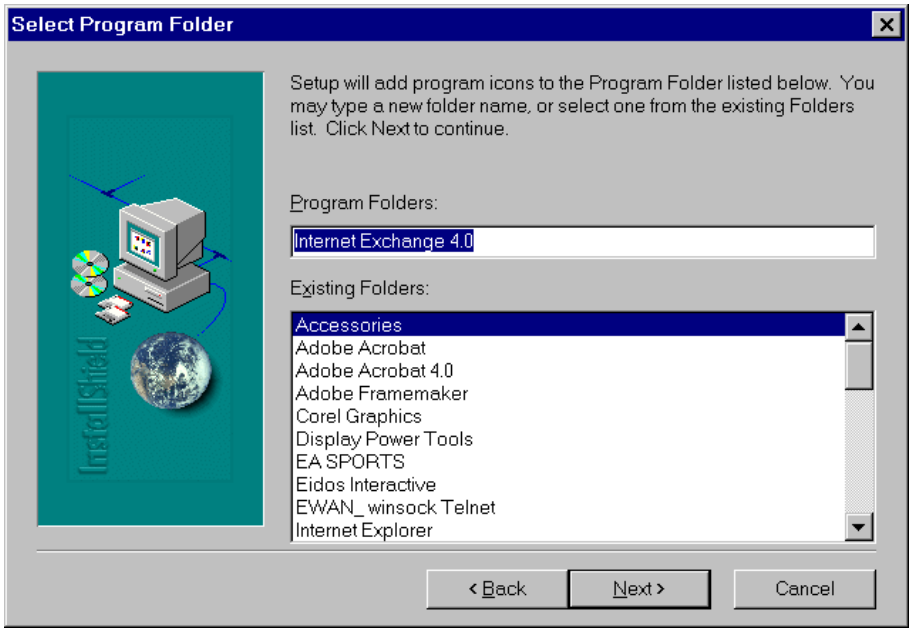

Figure 5k - Screen for creating/selecting program folder

The next dialog box (Figure 5k) allows the user to create a new program folder in the Start Programs menu or choose an existing folder in which Internet Exchange program icons will be placed. By default, a new program folder named **Internet Exchange 4** will be created in the Start Programs menu. Click on the Next button to start the installation process.

A new screen will appear after the installation (see Figure 5l). Click on the *Finish* button to complete the setup. Once you have clicked on the *Finish* button, the HTPassword screen will appear and you will be asked to supply a new password. The default user for this password is *administrator.* After entering the new password, press *Enter.* You are now ready to license the module(s) installed on the machine.

INSTALLING THE LICENSES

To install **Internet Exchange 4**, a license certificate containing the license information needed to activate the license keys is required. License keys need to be requested from an authorized license manager. After registration, a certificate containing information on the licensed modules is issued. This certificate is needed to identify and validate the user when installing the license key. Both the certificates and license keys are needed to run **Internet Exchange 4**.

License Types

There are three types of licenses for Internet Exchange: *Evaluation*, *Interim* and *Permanent*. Each of these license is described below.

Evaluation Keys

These are time-limited keys (normally 30 days) and are used with the freely available evaluation copies of Internet Exchange. Once a registration form is received from the customer, the authorized license manager generates this key and gives it to the client.

Interim Keys

These keys are also time-limited keys, except that an *interim* license can be updated to a *permanent* license at a later date. These keys are used for serialized or purchased copies of Internet Exchange.

Permanent Keys

These keys are used for the conversion of a given *interim* key into a *permanent* license, and are only applied to serialized copies of the software. Unlike *evaluation* and *interim* licenses, *permanent* licenses are based on the Internet Exchange serial number and the Fully Qualified Domain Name (FQDN) of the gateway machine. The permanent key is generated only by an authorized license manager.

Running the License Manager

The **Internet Exchange 4** licenses are installed/updated via the *License Update* pages provided by the Web Administration Interface. To install/update licenses, look for the Internet Exchange icon in the *Programs* menu. Click on the *Apache Web Server* (or you may run any Web server on yourmachine) to start the Web-based administration utilities. Then run your Web browser and type the name of your host in the URL field. If the machine running the Web-based Administration Tools is named *cuenca.ima.com*, type *cuenca.ima.com* in the URL field. The authentication page for the main Web Administration Interface will appear (see Figure 5l). Enter the user name and the corresponding password in the pop-up dialog box then click on the *OK* button.

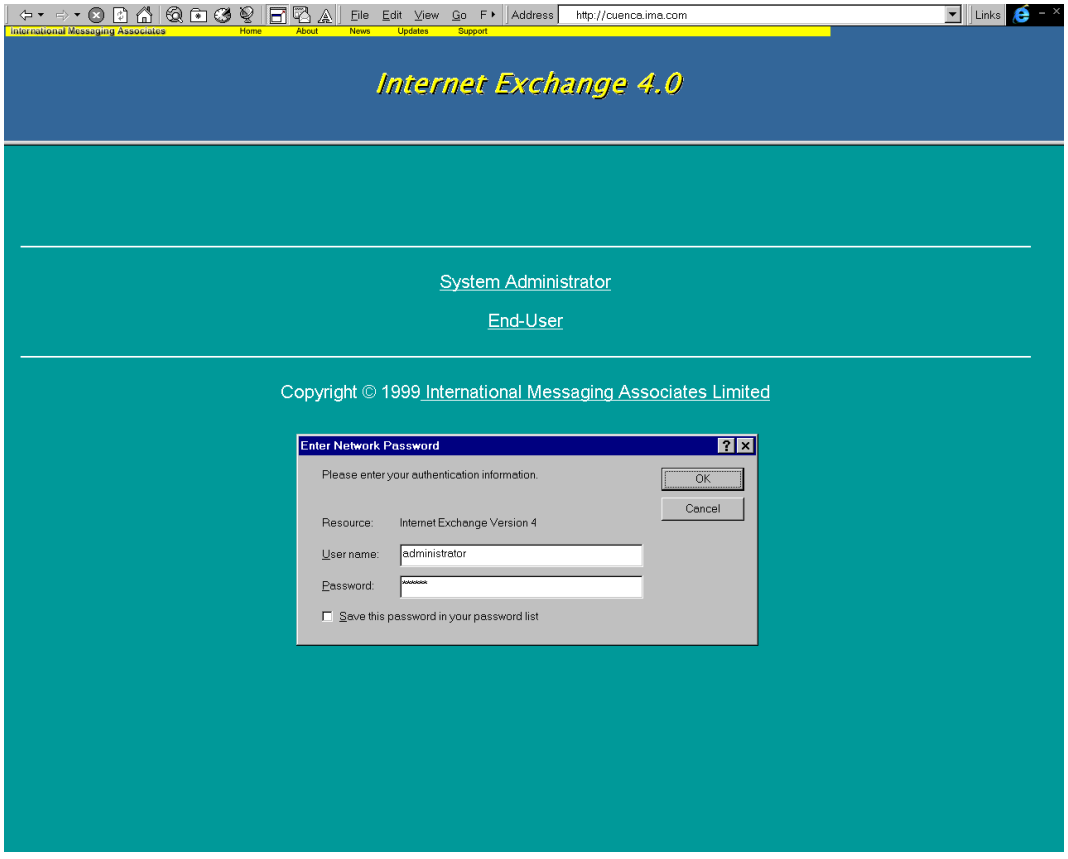

Figure 5l - Web Administration Interface Authentication Screen

If the user name and password that you entered have been verified to be correct, the main Web Administration Interface will appear (see Figures 5m.1 and 5m.2)

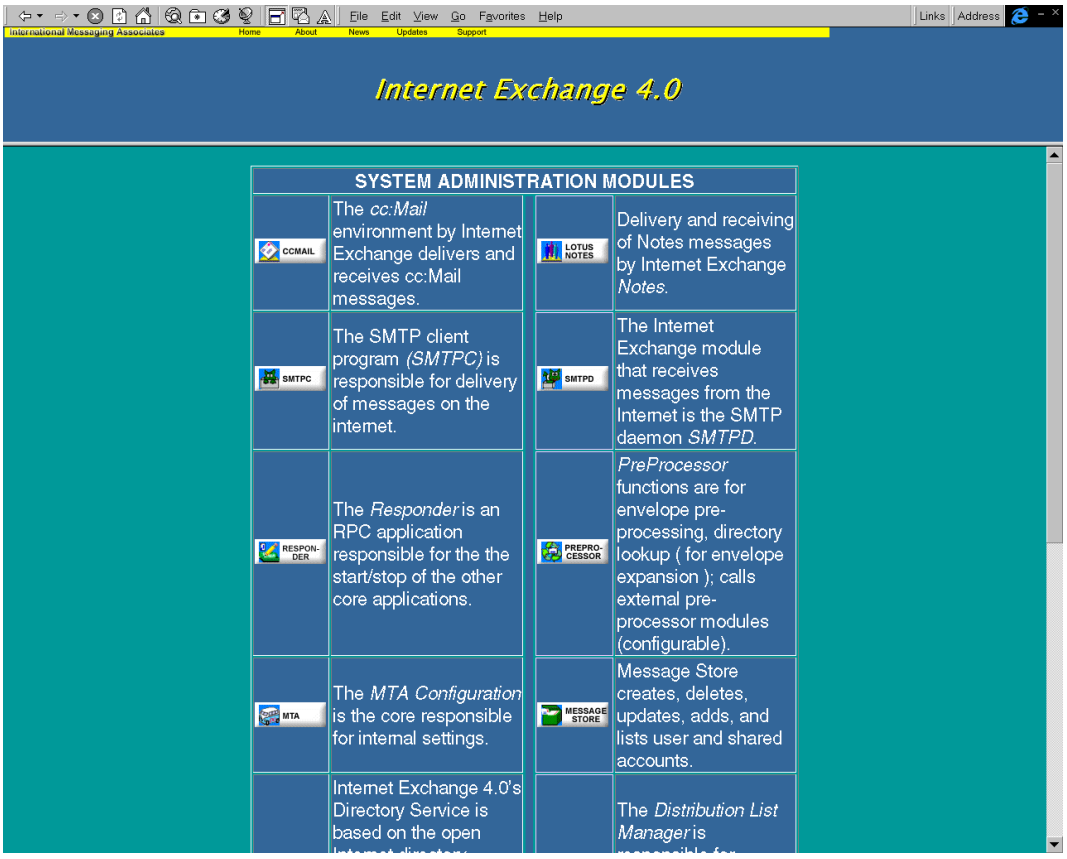

Figure 5m.1 - Main Web Administration Interface

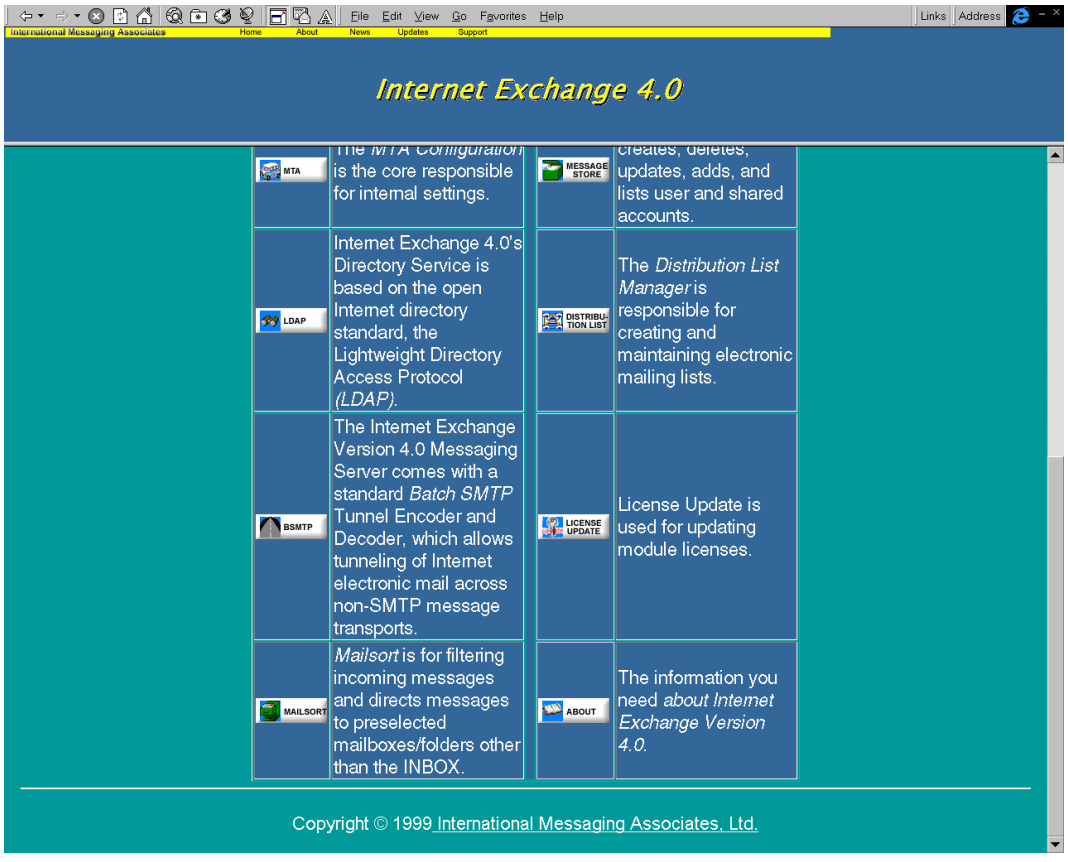

Figure 5m.2 - Main Web Administration Interface

Click on the *License Update* button to go to the main Licensing Tools page (see Figure 5n). In this page, click on the *License Manager* link. The next screen (Figure 5o) will ask for the directory where the certificates are stored and the name of the module to be licensed.

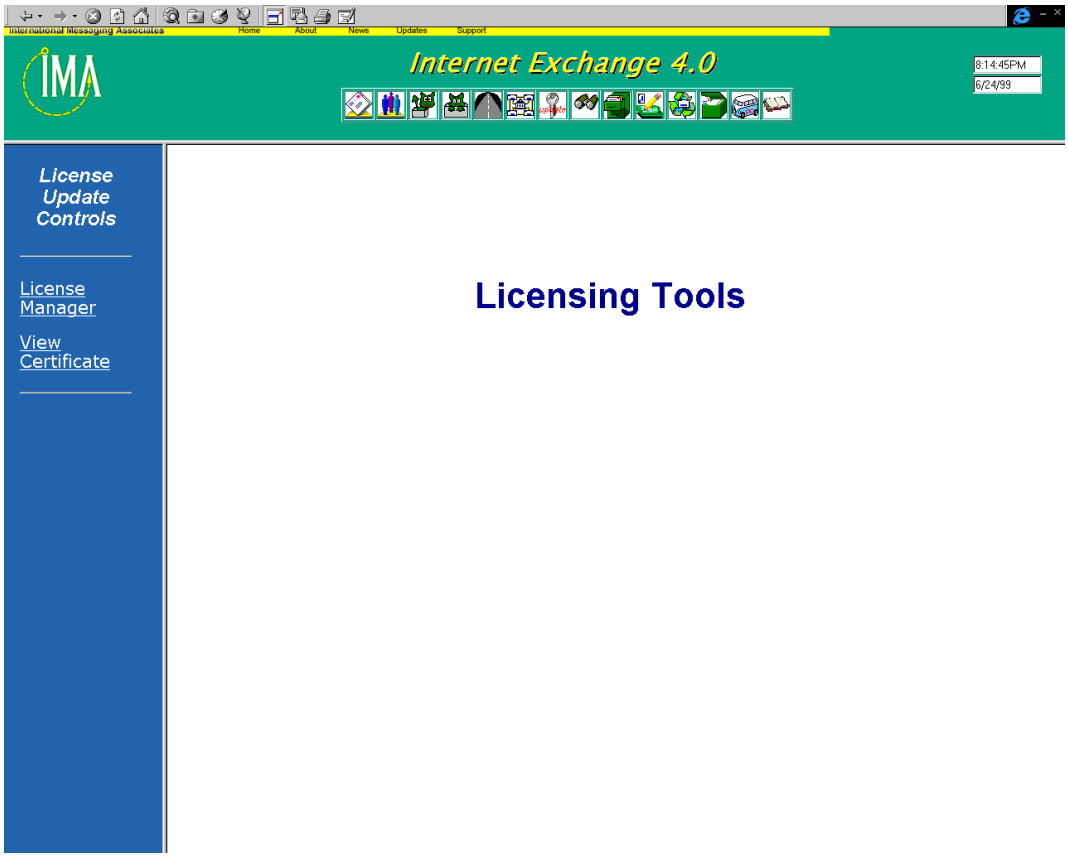

Figure 5n - Main Licensing Page

The screen above requires the *Certificate Directory* and *License Key* values.

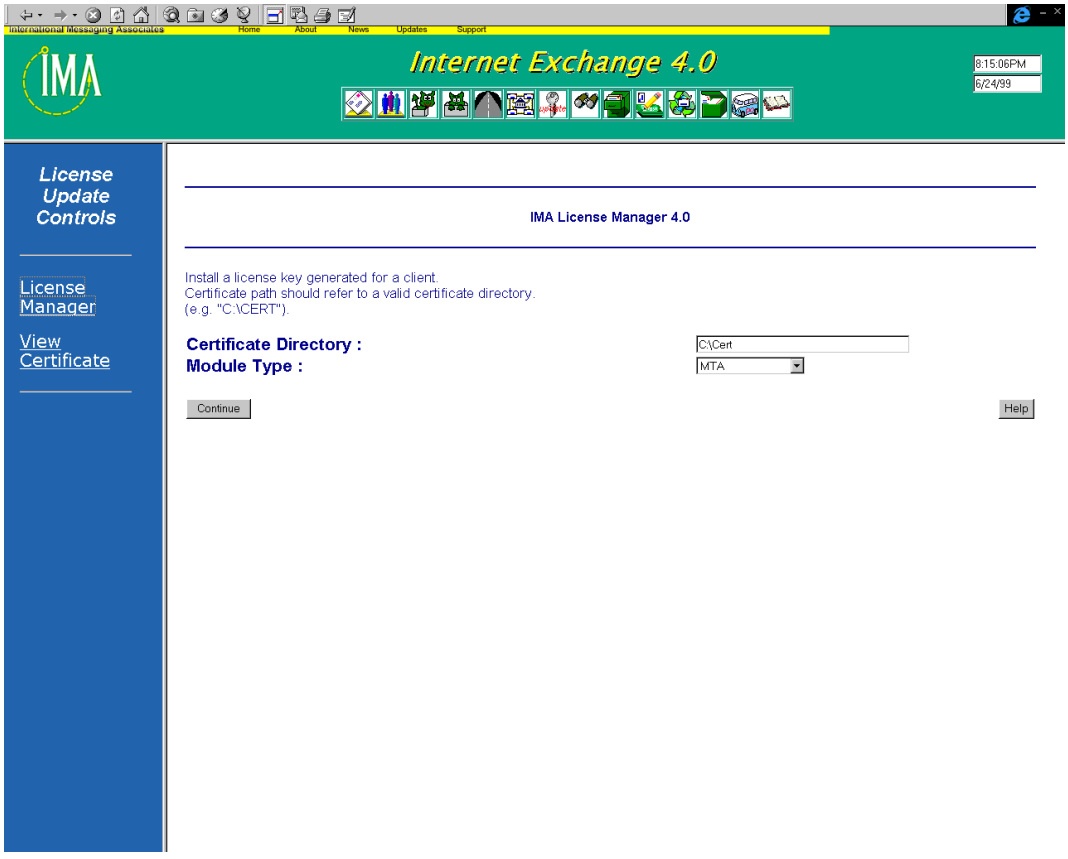

Figure 5o - License Manager

Certificate Directory

The initial directory entry displayed is based on the IEMTA.INI file entry. This entry should point to the directory containing the certificate files.

Module Type

Internet Exchange 4 modules consists of the *Internet Exchange MTA*, *Internet Exchange Message Store*, *Internet Exchange cc:Mail Connector Module*, and *Internet. Exchange Notes Connector Module.*

Click on the *Continue* button to proceed with the installation of the license key(s). The License Manager then verifies the existence of the certificate. If this file is missing, the licensing process will terminate. If the certificate is found, its contents are extracted and displayed in an HTML form in a new screen (Figure 5p).

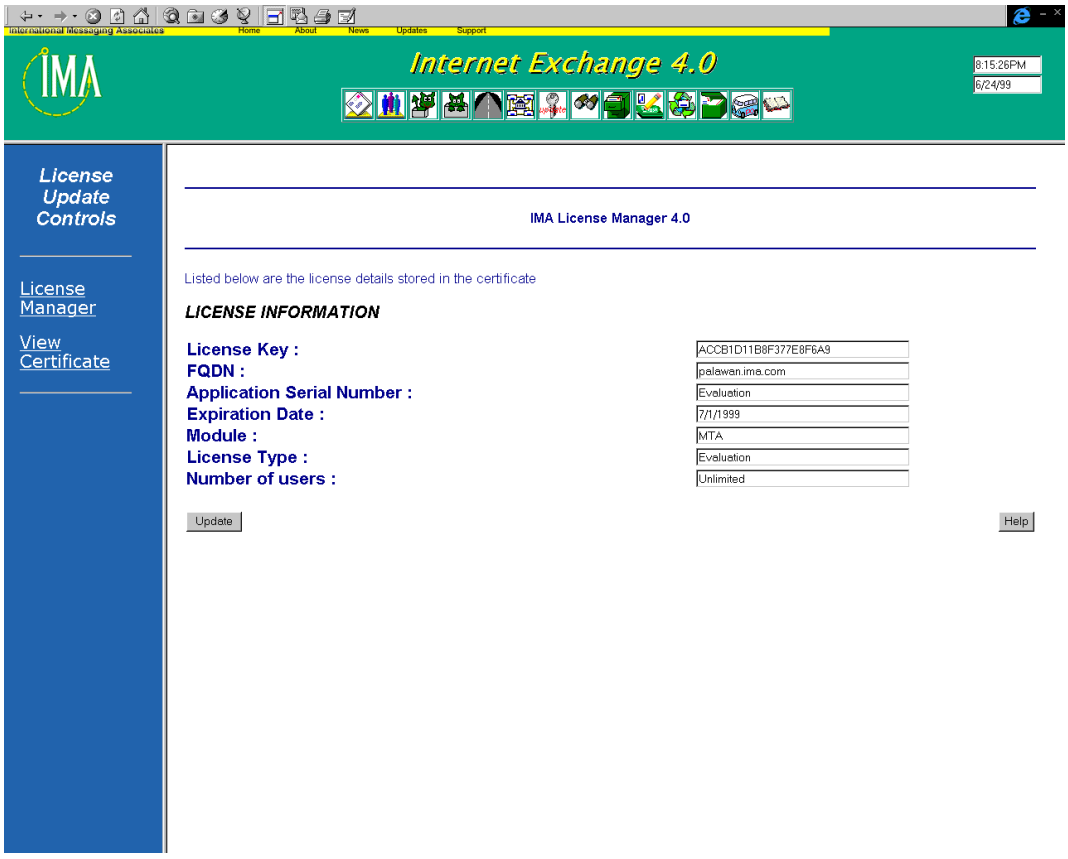

Figure 5p - License Information Page

This screen displays the license details extracted from the certificate, except for the *FQDN* field, which is based on the IEMTA.INI file.

License Key
The license key stored in the certificate.

FQDN
The Fully Qualified Domain Name based on the INI file entries, GatewayHostName, and GatewayDomain entries.

Application Serial Number
The application serial number for the module being licensed.

Expiration Date
Displays the date of validity for the certificate/license key. Date format should be *mm/dd/yyyy*.

Module
Displays the specific module tag that will use the license certificate.

License Type
Displays the type of license that will be issued for the client. License types include *Evalu-*

ation, Interim and *Permanent*.

Number of Users

Displays the number of allowed users.

Click on the *Update* button to continue the licensing process. If the operation is successful, the IEMTA.INI file is updated to reflect the license key used for the module.

An option to view the contents of the certificate can be selected in the left frame of the lup-date pages. Select *View Certificate* option. A screen showing the licensing information for the licensed modules will appear.

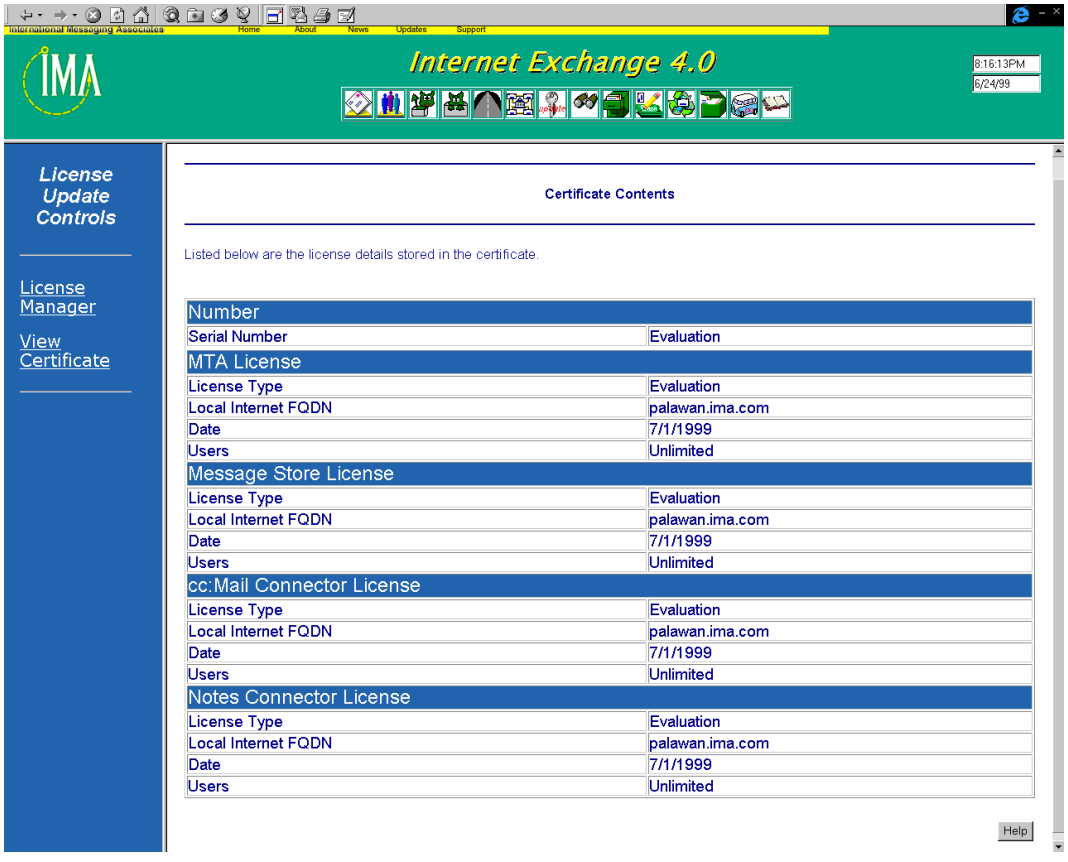

Figure 5q - View Certificate Page

PART 3

Operation and Administration

Internet Exchange Messaging Server

WEB ADMINISTRATION INTERFACE

Internet Exchange 4 uses a Web-based configuration and administration utility for managing the operations of the Messaging Server.

Once the web server has been installed and is running, bring up your web browser and point your URL to the URL of the web server installed by the **Internet Exchange 4** installation utility. The Web Administration Interface Authentication Page will appear (see Figure 6a). Click on the *System Administrator* link. A dialog box will appear asking for the user name and password.

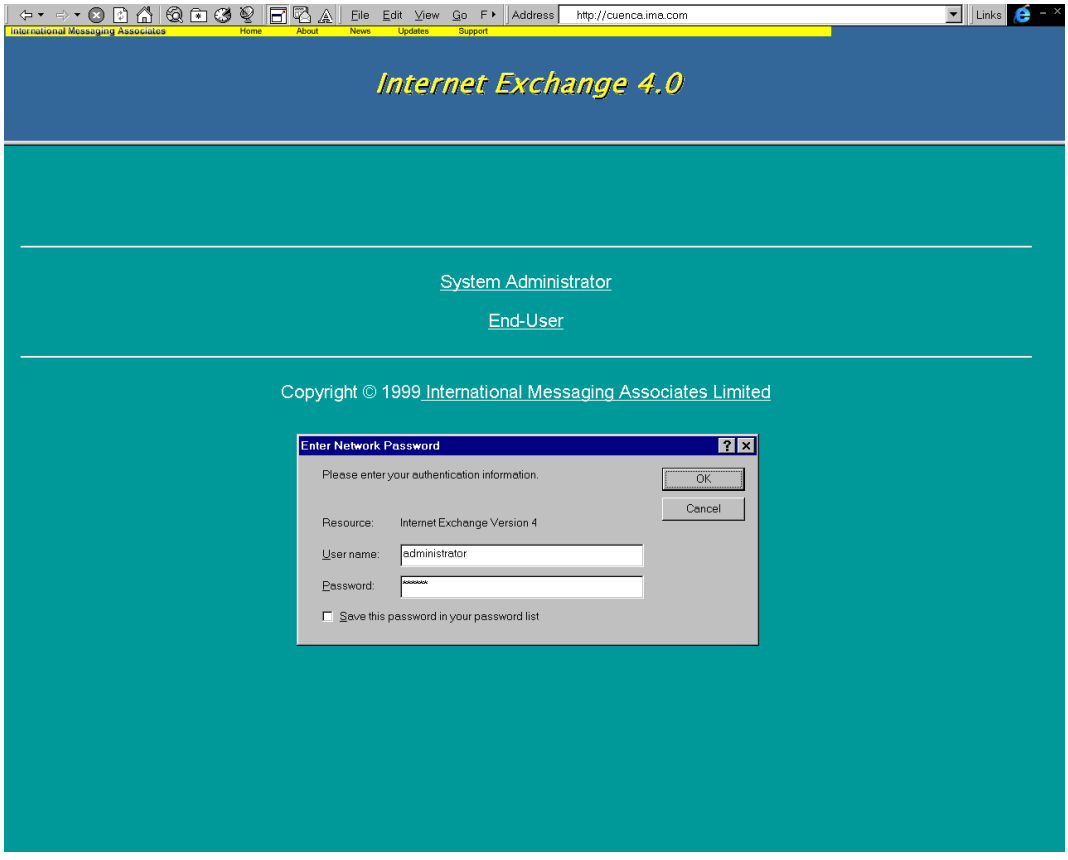

Figure 6a - Authentication Page

Enter the user name of the system administrator and the corresponding password in the text boxes provided. Then click on the *OK* button. If the user name and password are verified to be correct, the main Web Administration Interface will appear (see Figures 6b.1 and 6b.2).

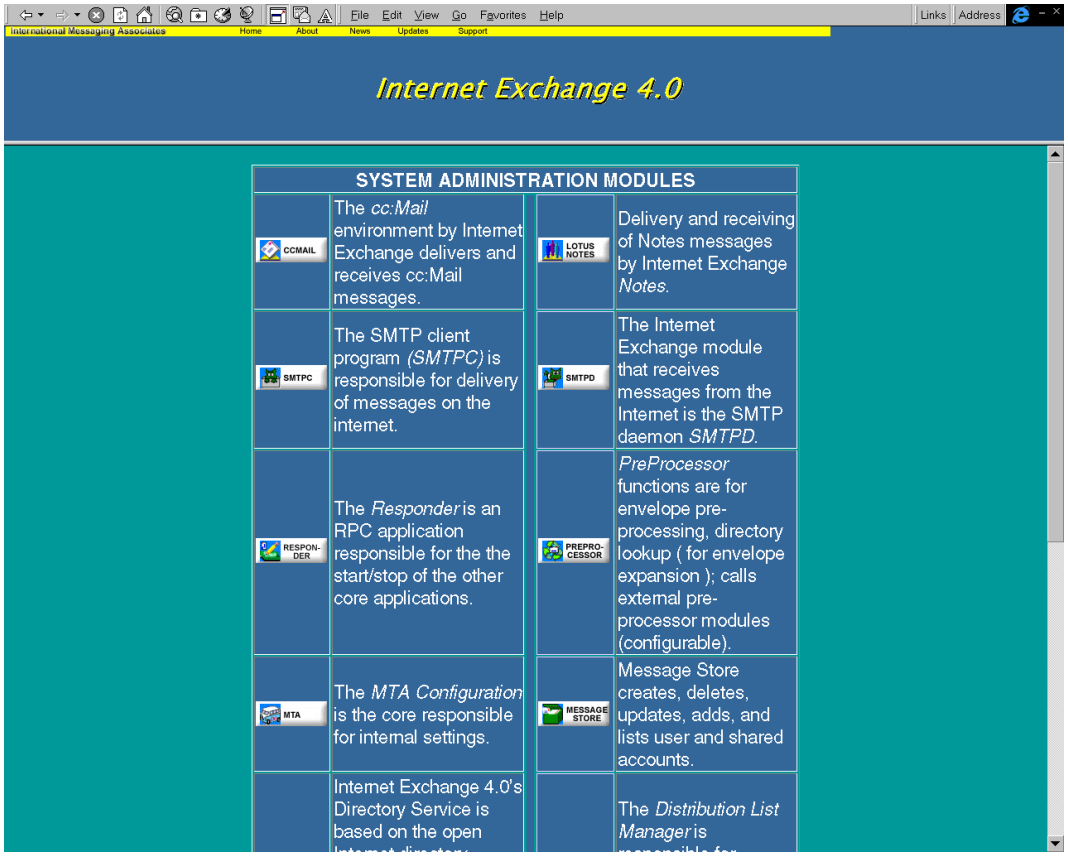

Figure 6b.1 - Main Web Administration Interface

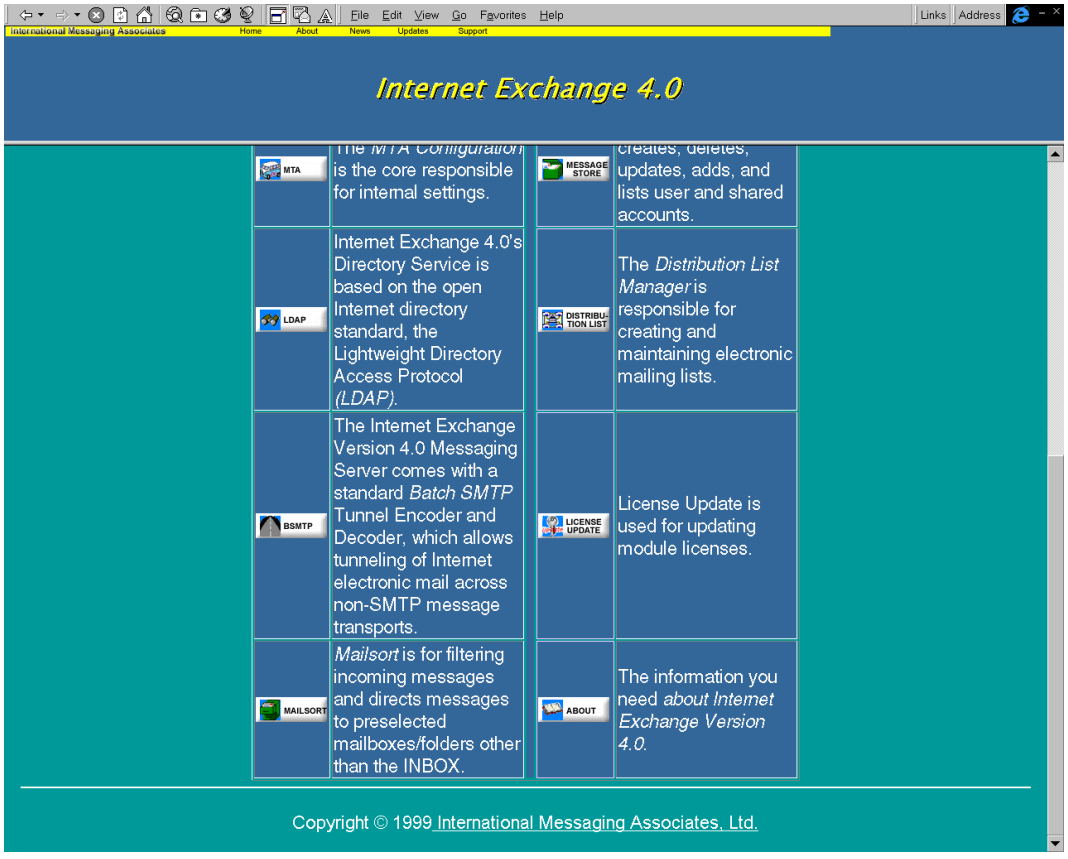

Figure 6b.2 - Main Web Administration Interface

SMTP DAEMON

The SMTP Daemon (SMTPD) is the module responsible for receiving messages from the Internet. It is a server process that continuously runs on the machine.

To configure SMTPD, click on the *SMTPD* button on the Main Administrator Web Administration Interface. A screen for configuring the SMTPD module's various options will appear (Figure 6c).

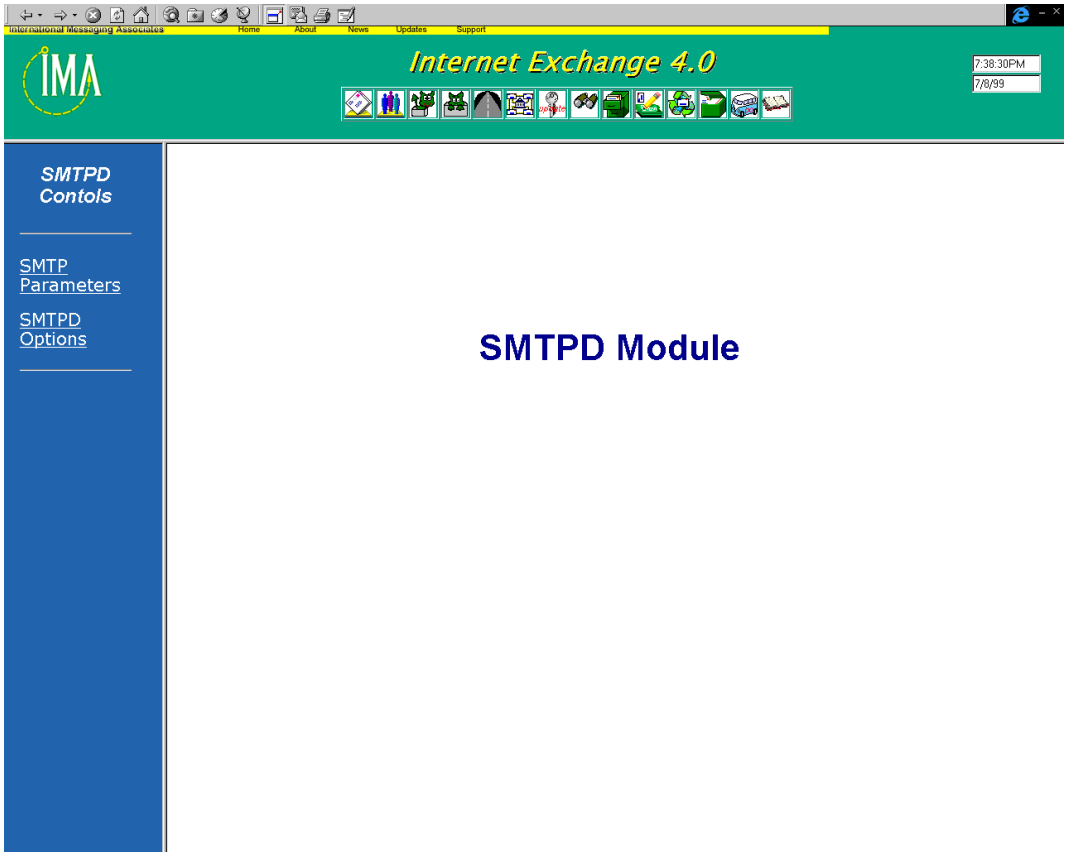

Figure 6c - Main SMTP Configuration Page

Common SMTP/ESMTP Parameters

The SMTP Daemon (SMTPD) and SMTP Client (SMTPC) Modules share a number of common parameters (i.e. SMTP Ports, ESMTP Support, etc.). To configure the different SMTP parameters , click on the *SMTP Parameters* link at the left side of the screen. A new screen will appear (Figure 6d). Enter the desired values in the text boxes provided and activate the *ESMTP Support* options desired by checking the appropriate boxes. Click the *Submit* button to implement the settings.

SMTP Ports

SMTPC port

Specifies the TCP port number to be used by SMTPC, which delivers messages on the Internet. This is useful when running Internet Exchange behind a firewall or any other non-standard setup. The default value is 25.

SMTPD port
Specifies the TCP port number to be used by SMTPD. This is useful when running Internet Exchange behind a firewall or any other non-standard setup. The default value is 25.

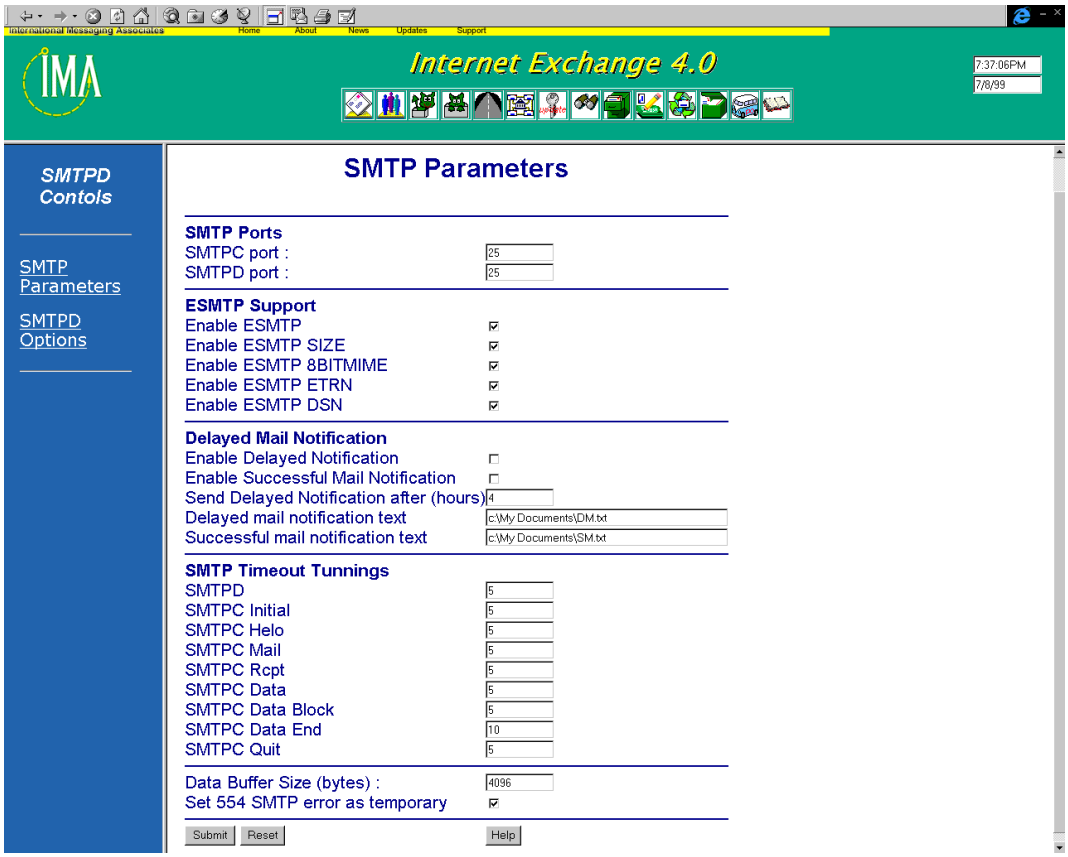

Figure 6d - Common SMTP Parameters

ESMTP Support

Enable ESMTP SIZE
Activates the *EnableESMTPSIZE* feature so that SMTPD/SMTPC can use the SIZE extension service. By default, *EnableESMTPSIZE* is enabled. If this option is enabled, SMTPD will advertise the EHLO keyword SIZE in response to the EHLO command. The gateway administrator can configure the maximum inbound message size for each peer domain as well as the default maximum size under the Peer Configuration section. The optional parameter for the keyword SIZE, which is used to specify the fixed maximum size, can be determined from the Peer Configuration by taking the maximum value of the size limit for all the peer domains.

Enable ESMTP 8BITMIME
When enabled, SMTPD announces Internet Exchange's support for 8BITMIME. By default, this option is enabled.

Enable ESMTP ETRN
Prompts SMTPD to announce its support for ETRN and accept ETRN requests. Once an ETRN request is received, SMTPD signals the SMTPC module to start a new queue pro-

cessor for the requested ETRN host. By default, this option is enabled.

Enable ESMTP DSN
Prompts SMTPD to announce its support for DSN (Delivery Status Notification) and accept DSN requests during MAIL FROM and/or RCPT TO commands. SMTPC also generates a DSN message when reporting the delivery status. By default, this option is enabled.

Delayed Mail Notification

Enable delayed mail notification
Prompts SMTPC to send a delayed notification message to the sender when ESMTP DSN is NOT enabled or a DSN request does not specify NOTIFY=NEVER. By default, this option is disabled.

Enable successful mail notification
When this option is enabled, Internet Exchange notifies the sender when a delayed message has been successfully sent. By default, this option is disabled.

Send delayed mail notification after (hours)
SMTPC sends the delayed message notification after this specified amount of time. The default value is 4 hours.

Delayed mail notification text
Specifies the path name of the file containing the message to be used to notify the user of a delayed message delivery. If no file name is specified or no file is found at the specified path, an appropriate default warning message is used.

Successful mail delivery text
Specifies the path name of the file containing the message that will be sent to the Postmaster when the gateway, after having sent at least one delayed message notification, eventually delivers a message. If none is specified, or if no file is found at that path, an appropriate default warning message is sent.

SMTP Timeout Tunings

SMTPD
Indicates the timeout value (in minutes) that SMTPD waits on an open socket. It should not need to be changed, but if unusual delays are experienced, this can be adjusted to stop SMTPD from timing out. The default value is 5 minutes.

SMTPC Initial
The period (in minutes) that SMTPC waits for the initial contact of a remote host to be completed. The default value is 5 minutes.

SMTPC Helo
The period (in minutes) that SMTPC waits for the remote system to respond to the HELO command. The default value is 5 minutes.

SMTPC Mail

The period (in minutes) that SMTPC waits for the remote system to respond to the MAIL FROM command. The default value is 5 minutes.

SMTPC Rcpt
The period (in minutes) that SMTPC waits for the remote system to respond to the RCPT TO command. The default value is 5 minutes.

SMTPC Data
The period (in minutes) that SMTPC waits for the remote system to respond to the DATA command. The default value is 5 minutes.

SMTPC Data Block
The period (in minutes) that SMTPC waits for the remote system to acknowledge an individual buffer transmission of message data. It can also be defined as the length of time wherein SMTPC waits between writes to the Winsock stack before it considers the remote system "dead". The default value is 5 minutes.

SMTPC Data End
The period (in minutes) that SMTPC waits for the remote system to respond to DATA phase wrap up represented by the dot (.) command. The default value is 5 minutes.

SMTPC Quit
The period (in minutes) that SMTPC waits for the remote system to respond to the QUIT command. The default value is 5 minutes.

Data Buffer size
The size, in bytes, of the data buffer used by the SMTP programs to read data from the Internet. If the gateway machine uses disk caching, set this option to the size of the read ahead buffer. The default value is 4096 (4K); the maximum buffer size is 32768 (32 K).

Set 554 SMTP error as temporary
RFC821 on SMTP is not clear as to whether “error 554 transaction failed during the DATA phase” should be regarded as a permanent error. Usually 5xx errors are permanent, but some SMTP servers return 554 errors as temporary errors. Internet Exchange takes the conservative approach and retries such messages later. If this option is set to NO, then such messages will be bounced instead of resent to their intended recipients. The default is value is YES.

SMTPD Options

To configure the different SMTPD options, click on the *SMTPD Options* link on the SMTPD configuration screen. A new screen for configuring SMTPD options (i.e. maximum number of SMTPD sessions, reject unqualified address, etc.) will appear (Figure 6e). Set the desired options and enter the maximum number of SMTPD sessions. Click the *Submit* button to implement the new settings.

Maximum SMTPD sessions
Specifies the maximum number of incoming simultaneous SMTP sessions allowed. Some Winsock stacks find it difficult to handle too many incoming SMTPD sessions. A value of zero indicates that there is no preset maximum value. The default value is 15.

Reject unqualified address
When enabled, SMTPD checks the recipient and sender addresses for a proper domain part, refusing to receive messages where it is absent, e.g. *user@host.com* is accepted but *user* alone is rejected. This option is useful in encouraging users to use FQDNs everytime they send mail to the Internet. By default, this option is disabled.

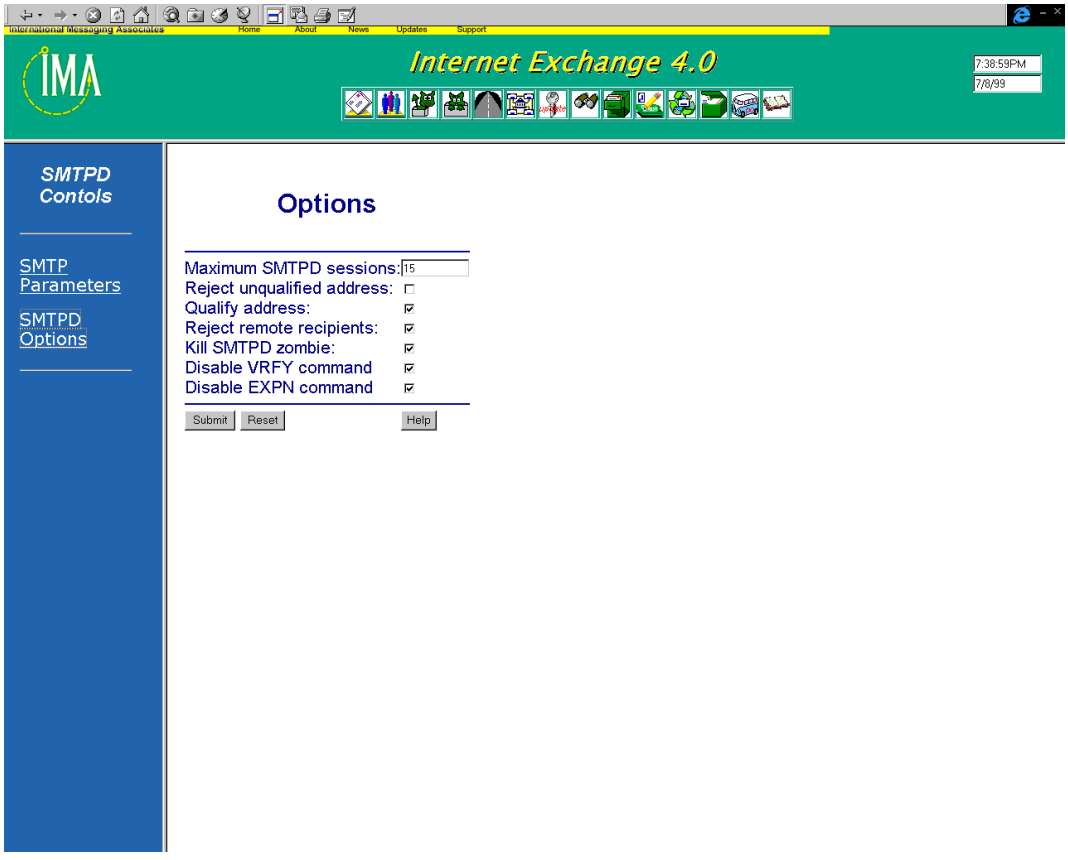

Figure 6e - SMTP Options

Qualify address
When enabled, SMTPD automatically appends the local domain part to the unqualified address. By default, this option is enabled.

Reject remote recipients
When enabled, SMTPD rejects incoming messages for remote Internet recipients. This is to prevent remote sites from trying to spoof messages by rerouting them through the gateway back to the Internet. This option is enabled by default.

Kill SMTPD zombie
When enabled, SMTPD will close the socket used by SMTPD when it last shut down prematurely. Thus, SMTPD will not get an *Address already in use* error when restarted. By default, this option is enabled.

Disable VRFY command
For security reasons, the "VRFY" (verify user) command is sometimes considered too intrusive: through this command, a remote host may confirm whether a particular user exists in a certain post office. Disabling the "VRFY" command causes SMTPD to respond with *252 command disabled* when a remote SMTP client issues this command. By default, this option is disabled.

Disable EXPN command
For security reasons, the "EXPN" (expand mailing list) command is sometimes considered too intrusive: through this command, a remote host may confirm whether a certain mailing list exists in a certain post office. Disabling the "EXPN" command causes SMTPD to respond with *550 command disabled* when a remote SMTP client issues this command. This option is disabled by default.

SMTP CLIENT

The SMTPC Module is responsible for the delivery of messages on the Internet. This is carried out by SMTPC by regularly checking for messages queued in the SMTP OUT queue. When messages are found, SMTPC establishes the required number of connections with external SMTP servers and transfers the messages to the appropriate Internet mail hosts.

To configure SMTPC, go to the main Web Administration Interface and click on the *SMTPC* button. A screen for configuring the SMTPC Module's various features will appear (Figure 6f).

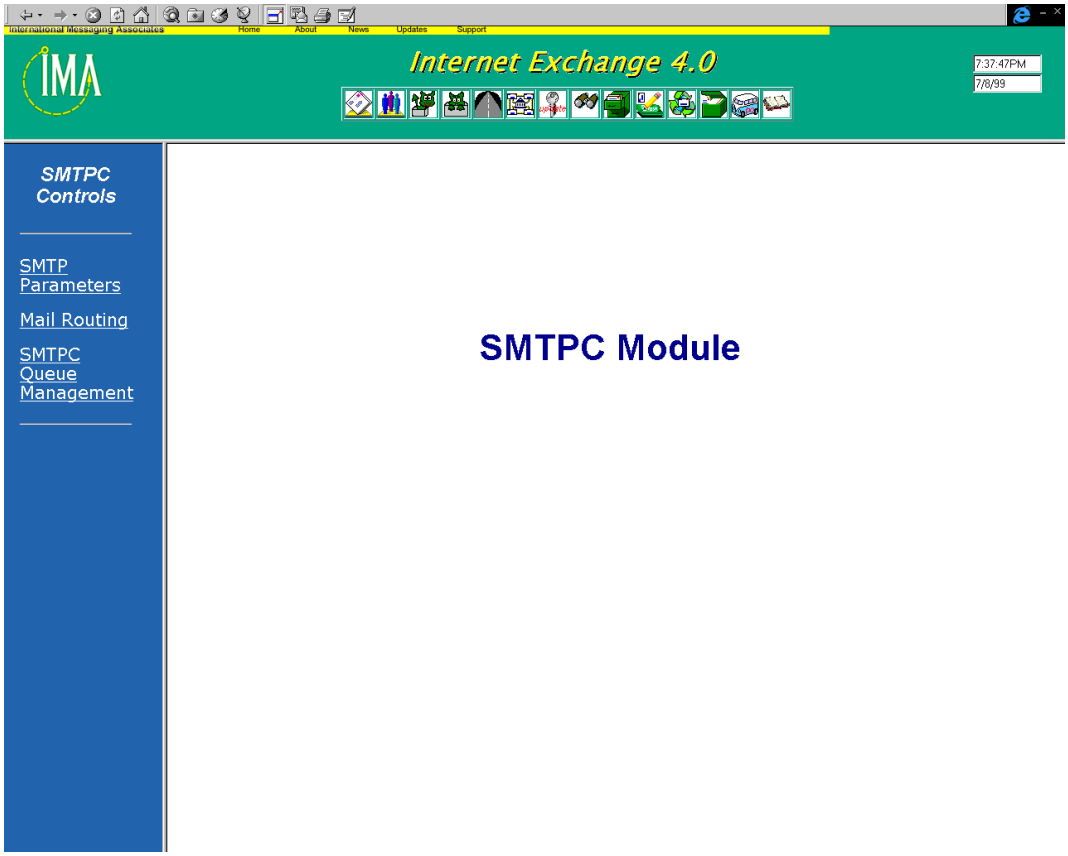

Figure 6f - Main SMTPC Configuration Screen

Common SMTP/ESMTP parameters

SMTPC shares a number of common parameters with SMTPD. To configure the different SMTP parameters (i.e. SMTP Ports, ESMTP Support, etc.), click on the *SMTP Parameters* link at the left side of the SMTPC configuration screen. The screen for configuring the common SMTP parameters will (see Figure 6d). Enter the desired values in the text boxes provided and activate the *ESMTP Support* features desired by checking the appropriate boxes. Click the *Submit* button to implement the settings.

NOTE: For a detailed explanation of the common SMTP parameters, please refer to the SMTPD section on page 6-4).

Mail Routing

To configure routing options, click on the Mail Routing link on the SMTPC configuration screen. The screen for configuring various mail routing options will appear (Figure 6g).

Select the name resolution approach desired and the enter the host table filename in the textbox provided. To configure the various DNS parameters (i.e. Maximum number of DNS Caching records, DNS retries, etc.), enter the desired values in the textboxes provided. Click the *Submit* button to implement the settings.

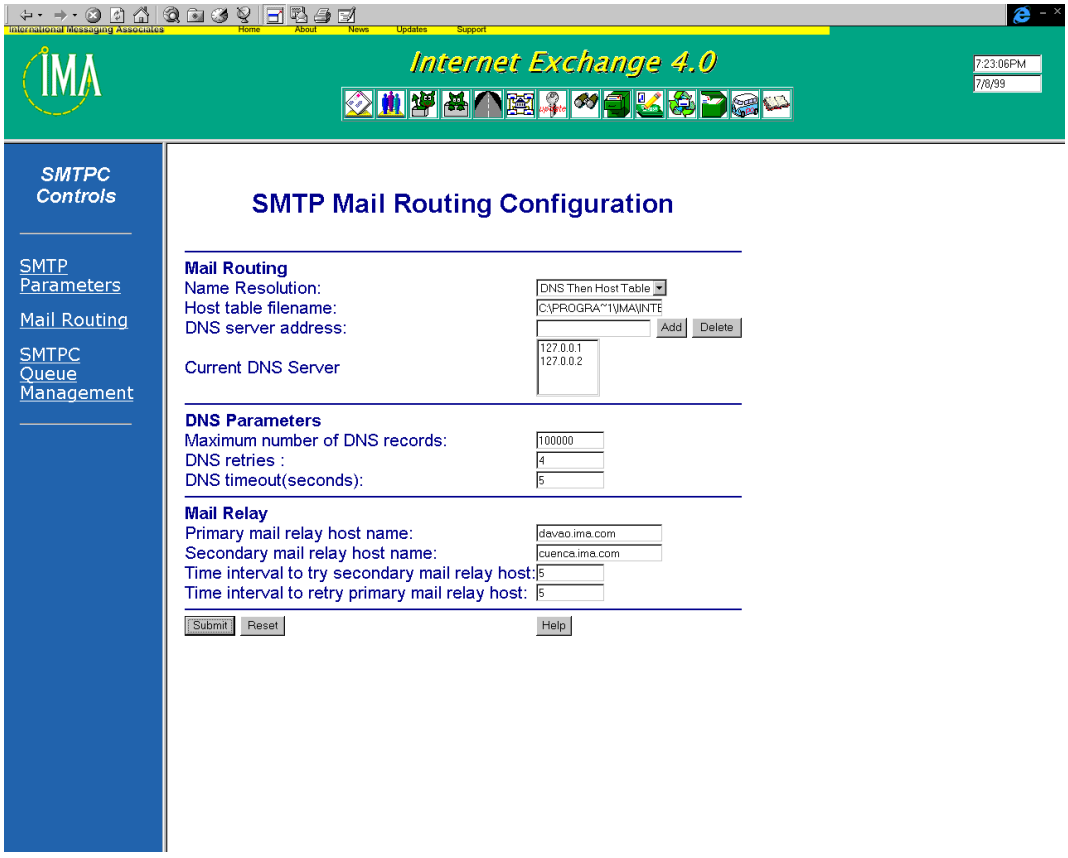

Figure 6g - Mail Routing Options

Mail Routing Parameters

Name resolution

There are several name resolution options, namely: DNS only, DNSThenHostTable, HostTableThenDNS, HostTableOnly, and MailRelayHost. Any combination of DNS or host table lookup can be used regardless of the order. When the Mail relay host only routing option is disabled, it is recommended that DNS be used if possible, as this usually results in the most reliable routing and greatest throughput.

Host table filename

Stores the location of the Internet host table for address resolution. Even if the DNS is used for name resolution, it is necessary to configure a host table that contains at least the name and address for the gateway machine as well for the default mail relay host. This

will allow Internet Exchange to send the message to the default mail relay host for further routing in case the gateway encounters problems when communicating with the name server(s).

DNS server address
SMTPC contacts the list of configured DNS servers to send messages. Each address must be of the form a.b.c.d, where each number is between 0 and 255. SMTPC can be configured to contact a list of DNS servers and/or consult the local host table when resolving hostnames.
Operation without access to DNS servers can be achieved only when the outgoing mail is routed through a mail relay host. In this case, the name and address of the mail relay host(s) must appear in the local host table.

DNS Parameters

Maximum number of DNS records
Specifies the maximum number of DNS records cached in the database on the local disk. The DNS cache greatly improves the throughput of Internet Exchange, particularly when the DNS server(s) are not on a local LAN. The default value is 1,000,000 which balances throughput against greater disk space used for the cache. A value of zero disables DNS caching.

DNS retries
Specifies the number of times a DNS query is retried after a the operation has timed out. The default value is 4.

DNS timeout (seconds)
Specifies the length of time in seconds before a DNS request timeout is registered. The default value is 5 seconds.

Mail Relay

Primary mail relay host name
A mail relay host is another host capable of forwarding mail to third parties. Most Internet hosts have this capability, but before using them it is polite to ask for permission from the local administrator. The mail relay hosts used for delivery of outbound traffic are not necessarily the same ones used as MX forwarders for incoming traffic, although in some configurations they may coincide. Internet Exchange gives the administrator the ability to define a number of strategies to deliver mail, some of which involve using a mail relay host as primary or last-resource mail router.

If SMTPC is unable to resolve a hostname by either DNS or host table lookup, it routes messages to the primary mail relay host for forwarding. This option is also used if routing is configured to mail relay host only.

Secondary mail relay host name
In Internet Exchange, it is also possible to define a secondary mail relay host to be used when the primary relay host is, for whatever reasons, unavailable. If this option is enabled, a secondary mail relay is configured for use when the primary mail relay host is unavailable.

Time interval to try secondary mail relay host
The length of time in minutes that the primary mail relay host is unavailable, after which it is considered off-line and the message is routed to the secondary mail relay, if the latter is enabled.

Time interval to retry primary mail relay host
The number of minutes before the gateway attempts to revert to the primary mail relay host after the previous attempt has failed.

SMTPC Queue Management

To configure the different queue handling options (i.e. SMTPC queue directory, maximum number of Pending Queue processors, maximum number of Deferred Queue processors, message priority, etc.), click on the *SMTPC Queue Management* link on the Main SMTPC Configuration screen. A new screen for configuring queue management options will appear (Figure 6h). Enter the directory for the SMTPC queue and the desired values for the various queue management parameters in the appropriate textboxes. Click on the *Submit* button to implement the new settings.

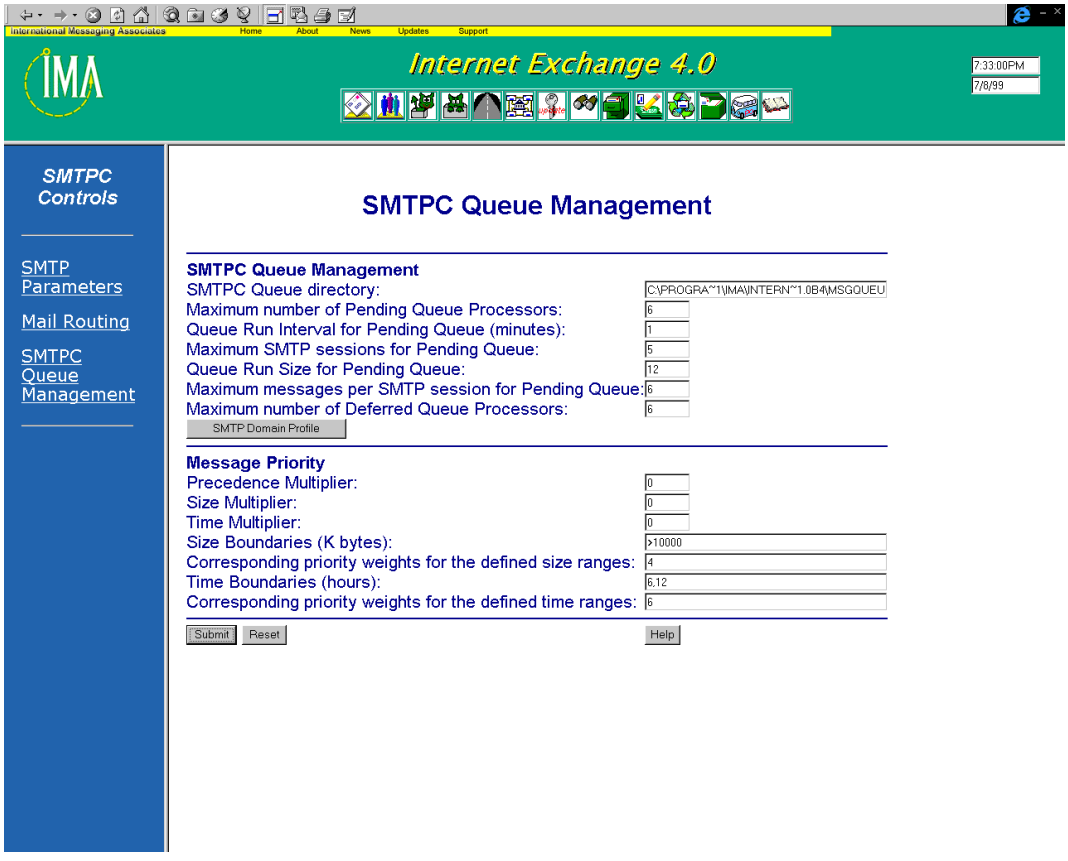

Figure 6h - Configuring SMTPC Queue Management

SMTPC Queue Management

SMTPC Queue Directory

The queue directory that SMTPC will use to store outgoing messages.

Maximum number of Pending Queue Processors

The maximum number of Pending Queue Processors that will run concurrently. Pending Queue Processors are responsible for processing messages in the Pending Queue. Each queue processor handles messages independently. The default value is 6.

Queue Run Interval for Pending Queue (minutes)

How long the Pending Queue Processor should check for pending messages in minutes. If pending messages exist, they will be processed immediately. The default value is 1.

Maximum SMTP sessions for Pending Queue
Each Pending Queue Processor is capable of establishing multiple concurrent SMTP sessions. This option specifies the maximum number of SMTP session for each processor. The default value is 5.

Queue Run Size for Pending Queue
At each queue run, each Pending Queue Processor will process messages simultaneously. The queue run size specifies the number of message for each queue run. The default value is 12.

Maximum messages per SMTP session for Pending Queue
The highest number of messages that can be sent using a single SMTP connection. When this number is increased, more messages can be sent to a remote SMTP server on each connection. The default value is 6.

Maximum number of Deferred Queue Processors
The maximum number of Deferred Queue Processors that will run concurrently. Each Deferred Queue Processor is responsible for processing deferred messages for a particular deferred SMTP domain. The default value is 6.

Maximum messages per SMTP session for Pending Queue
The highest number of messages that can be sent using a single SMTP connection. When this number is increased, more messages can be sent to a remote SMTP server on each connection. The default value is 6.

Maximum number of Deferred Queue Processors
The maximum number of Deferred Queue Processors that will run concurrently. Each Deferred Queue Processor is responsible for processing the deferred messages for a particular deferred SMTP domain. The default value is 6.

SMTP Domain Profile

It is not unusual for Internet Exchange to communicate with Internet hosts that have different capabilities, particularly with regard to email formats. The Peers configuration screen allows such information to be recorded and used in preparing outgoing messages for the Internet. This ensures that messages sent to the Internet are successfully decoded by recipients. The information is stored as a list of peers for which certain capabilities apply. These capabilities apply to a specific domain and all its sub-domains, unless a more specific capability exists within the database (PEER.BTR).

To configure the peer domains, click on the *SMTP Domain Profile* button on the SMTPC Queue Management screen. A screen for creating, editing, and deleting peer domains will appear (see Figure 6i). The front end of the interface lists all the domain names stored in the database. A special entry called "default" is added when the database file is being initialized by the system. This entry cannot be removed from the database.

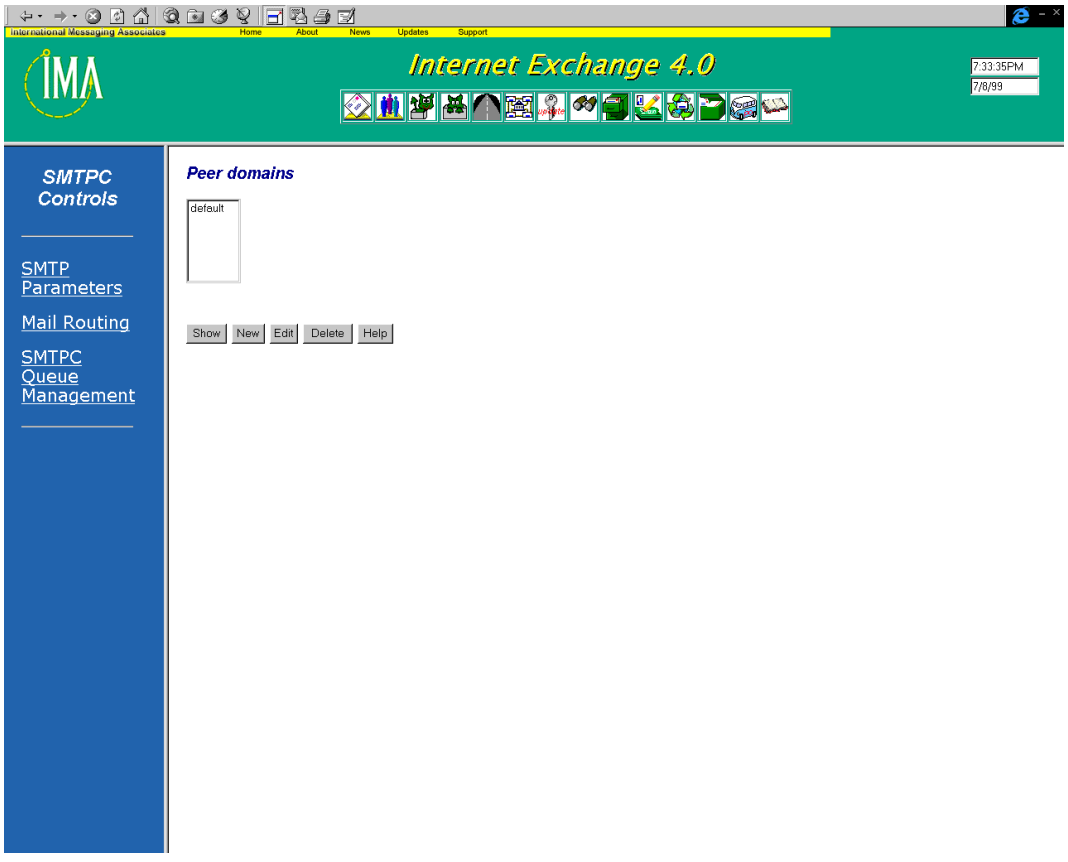

Figure 6i - Configure Peer Domains

To show an existing peer domain, select an entry from the list box. Click on the *Show* button and a web-based interface (see Figures 6j.1 and 6j.2) for modifying the peer domain's various options will appear.

To edit an existing peer domain, select an entry from the list box. Click on the *Edit* button to modify the peer domain's various attributes.

To create a new peer domain, click on the *New* button and the web-based interface for creating a peer domain and configuring its various options (i.e. Domain Name, SMTP Connection, SMTPC Profile, Native Attachment Encoding, etc.) will appear.

To remove an existing peer domain, select an entry from the list box. Click on the *Delete* button on the screen.

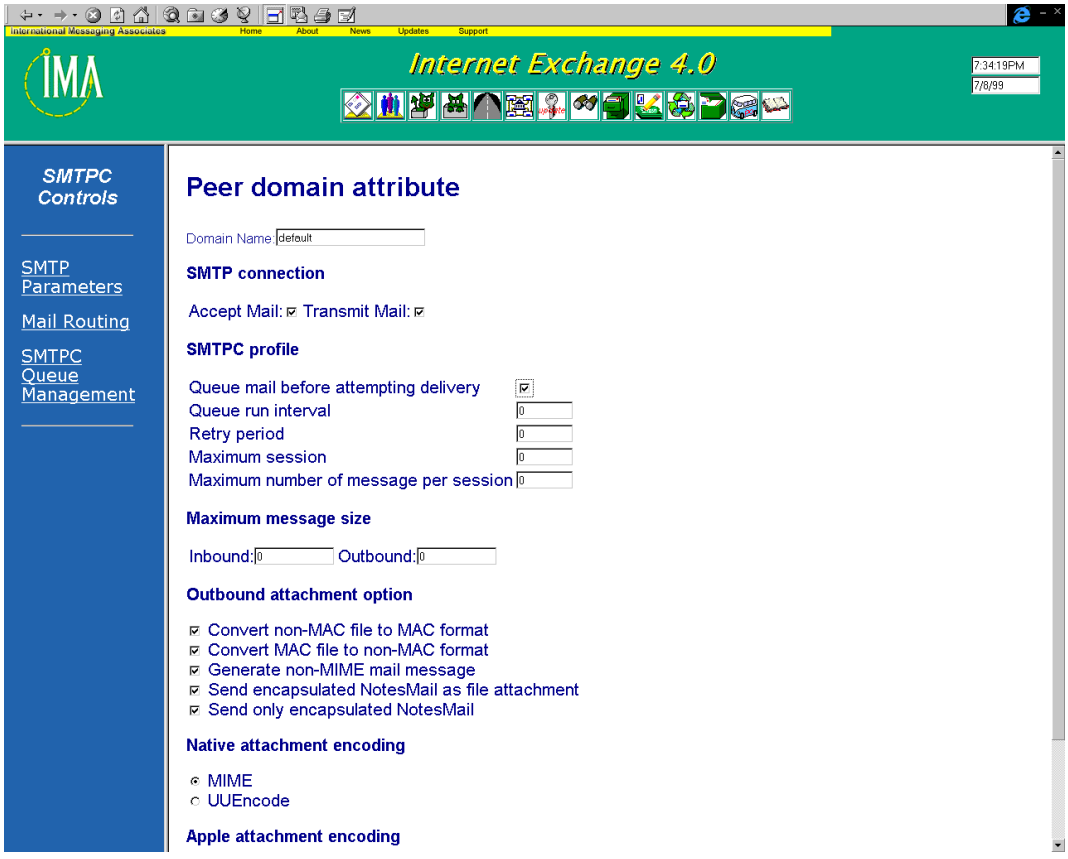

Figure 6j.1 - Configuring Peer Domain Attributes

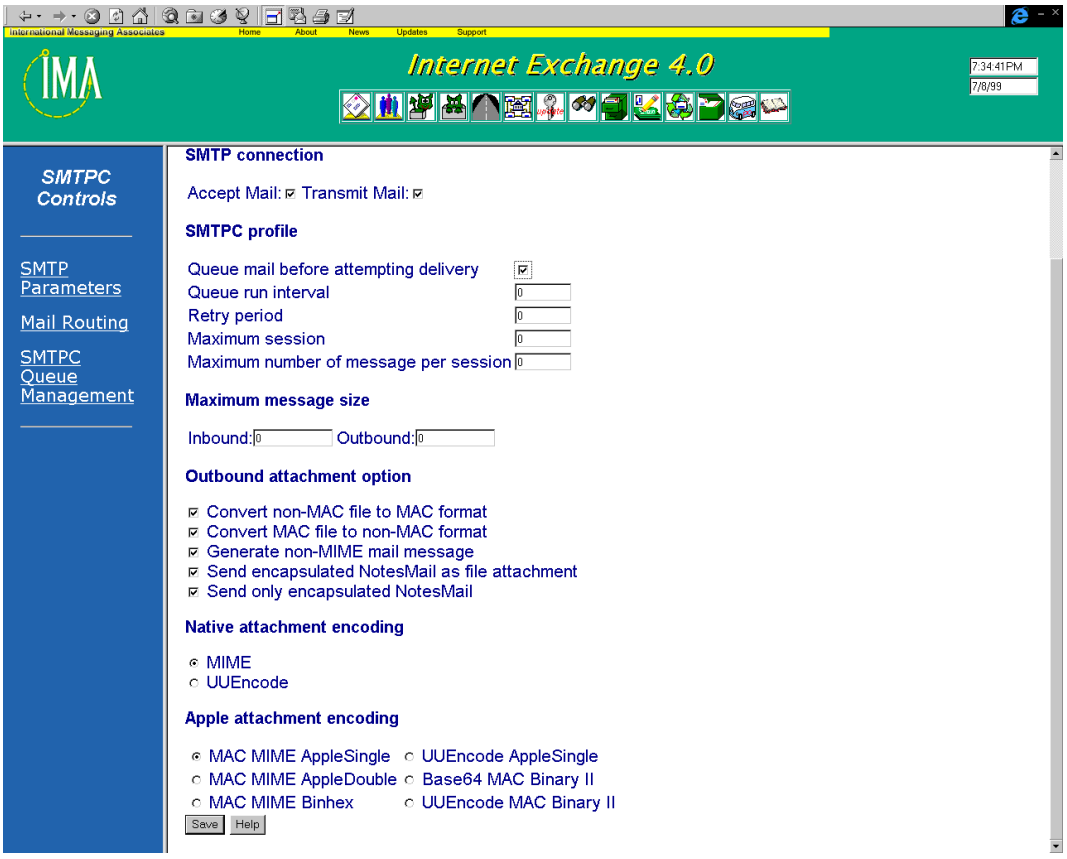

Figure 6j.2 - Configuring Peer Domain Attributes

Peer Domain Attribute

Domain Name

The domain or subdomain that has the specified capabilities.

SMTP Connection

Accept Mail

If Internet Exchange is not allowed to receive mail from a remote host, SMTPD rejects a HELO/EHLO command from that host with the following response:

550 host sales.xyz.org is not authorized to connect to iegate.jade.net

The default value is YES.

Transmit Mail

If Internet Exchange is not allowed to transmit mail to a remote site, CCOUT/NOTE-SOUT bounces any message destined for that host back to the original cc:Mail/Notes sender. The default value is YES.

SMTPC Profile

Queue mail before attempting delivery

When this option is set to ON, all outgoing messages for this domain will be queued first, i.e. placed in the SMTPC deferred queue, and they will then be processed together at the

queue run for this domain. This will make use of the overall system resource more efficiently.

For the dial-up connected ETRN hosts/domains, it is suggested to queue mail first before any delivery attempt, until an ETRN request is received. When this option is OFF, all outgoing messages will be attempted first and will be queued if the attempt fails. This is suitable for those domains that require immediately delivery. The default is OFF.

Queue run interval (in minutes)
How long the SMTPC should actively start a new Deferred Queue Processor to process the deferred messages for this domain. For those ETRN hosts, it is suggested to have a longer queue run interval, as the queue run for the ETRN host will be triggered by the ETRN command once the ETRN host is connected. The default is 15 minutes.

Retry period (in hours)
How long SMTPC should keep retrying the deferred messages for this domain. When it expires, SMTPC will bounce the messages to the sender. The default is 72 hours.

Maximum sessions
The maximum number of simultaneous outbound SMTP connections can be established for this domain. The default is 5.

Maximum number of messages per session
The highest number of messages that can be sent using a single SMTP connection. When this number is increased, more messages can be sent to a remote SMTP server on each connection. The default is 6.

Maximum Message Size

The largest message size, in bytes, that can be sent to and received from the selected domain. The smallest size allowed is 8,192 bytes (8K). A value of zero indicates no limitations.

Inbound
If the Inbound limitation is exceeded, SMTPD will reject the mail during SMTP session. The default is 0 (i.e. unlimited).

Outbound
If the Outbound limitation is exceeded, SMTPC bounces the message back to the original sender. The default is 0, i.e, unlimited.

NOTE: The following settings are used by the cc:Mail and Lotus Notes Connector modules.

Outbound Attachment Option

Convert non-MAC file to MAC format
When enabled, converts all non-Apple attachments to Apple format by adding a header and an empty resource fork and encoding the attachments using the Apple encoding method specified below. This option is useful when Internet Exchange is communicating

primarily with a network of Macintosh computers.

Convert MAC file to non-MAC format
When enabled, strips all Apple attachments of their headers and resource fork, allowing non-Macintosh sites to access the information easily.

Generate non-MIME mail message
When activated, ensures that no MIME messages are generated for this peer. This is useful when communicating with older email systems that do not understand MIME. In this case, either UUENCODE or BinHex 4.0 is used to encode binary attachments; if the peer does not contain any Macintosh recipients, it is advised to select UUENCODE encoding.

Send encapsulated NotesMail as file attachment
Attaches the native Lotus Notes.NSF to the message as well as the message text and attachments (if any). This is only useful if the recipient is also a Lotus Notes user. If the remote Internet recipients are also using Internet Exchange Lotus Notes connectors, this option can be used to set up a "Virtual Intranet" Notes network via Internet Exchange. This option is used only by the Notes Connector.

Send only encapsulated NotesMail
Attaches the native Lotus Notes .NSF to the message only, stripping the message of the text and attachments (if any). This is only useful if the recipient is also a Lotus Notes user. If the remote Internet recipients are also using Internet Exchange Lotus Notes connector, this option can be used to set up a "Virtual Intranet" Notes network via Internet Exchange. This option is used only by the Notes Connector.

Native Attachment Encoding

MIME
Specifies that non-Apple attachments are to be encoded using the MIME standard.

UUENCODE
Specifies that non-Apple attachments are to be encoded using the older UUENCODE format. The Generate non-MIME message option determines whether MIME headers should be generated for messages or not.

Apple Attachment Encoding

MAC MIME AppleSingle
Specifies that outgoing Macintosh attachments are to be encoded using the MacMime AppleSingle standard.

MAC MIME AppleDouble
Specifies that outgoing Macintosh attachments are to be encoded using the MacMime AppleDouble standard.

MAC MIME Binhex
Specifies that outgoing Macintosh attachments are to be encoded using the BinHex 4.0 standard. The Generate non-MIME message option determines whether MIME headers should be generated for messages or not.

UUEncode AppleSingle
Specifies that outgoing Macintosh attachments are to be encoded using the AppleSingle standard via UUENCODE instead of MacMime. The Generate non-MIME message option determines whether MIME headers should be generated for messages or not.

Base64 MAC Binary II
Encodes Mac Binary II attachments using the base-64 encoding scheme. This option is not used by the cc:Mail connectors. If this is selected, CCOUT uses MAC MIME AppleSingle instead.

UUEncode MAC Binary II
Encodes Mac Binary II attachments with UNIX-style x-uue Content-Transfer-Encoding. This option is used only by the Notes Connector. If this is selected, CCOUT uses MAC MIME AppleSingle instead. After configuring these options, click on the Add button to add the new domain to the list of recognized domains.

Message Priority
SMTPC assigns a priority weight to each message based upon three factors:

- The predefined message precedence
- The message size
- The total deferred time (for messages in the Deferred Queue)

The message priority weight is calculated from the following formula:

*Priority weight = precedence * Mp + size * Ms + deferred_time * Md*

The lower the priority weight, the higher the priority level and thus the sooner the message will be processed.

Precedence Multiplier (Mp)
Specifies the multiplier value for the Precedence factor. It is an integer value and is used relative to the other factors, size multiplier and time multiplier. The default value is 0.

Size Multiplier (Ms)
Specifies the multiplier value for the Size factor. It is an integer value and is used relative to the other factors, precedence multiplier and time multiplier. The default value is 0.

Time Multiplier (Md)
It specifies the multiplier value for the Time factor. It is an integer value and is used relative to the other factors, size multiplier and precedence multiplier. The default value is 0.

Size Boundaries (K bytes)
Size boundaries are used to classify messages into different ranges based on size. Different weight will then be assigned for the defined ranges. The weights are used for calculating the total priority weight.

e.g. 10, 1000, 10000
The boundaries defines 4 ranges of sizes, sizes less than 10K (<10), sizes between 10K

and 1,000K, sizes between 1,000K and 10,000K, and sizes larger than 10,000K (>10,000).

Corresponding priority weights for the defined size ranges. e.g. 0,2,4,10
The defined weights are assigned to the corresponding size range defined by the Size Boundaries. This will assign the weights to the corresponding size range defined above.

Assign 0 to (<10) range,
Assign 2 to (10,1000) range,
Assign 4 to (1000,10000) range.
Assign 10 to (>10000) range

Time Boundaries (hours) e.g. 1,6,12:
Time boundaries are used to classify messages into different ranges based on the deferred time. Different weight will then be assigned for the defined ranges. The weights are used for calculating the total priority weight.

The boundaries defines 4 ranges of deferred time: deferred time shorter than 1 hour (<1), deferred time between 1 hour and 6 hours (1,6), deferred time between 6 hour and 12 hours (6,12), and deferred time longer than 12 hours (>12).

Corresponding priority weights for the defined time ranges e.g. 1,4,6,12
The defined weights are assigned to the corresponding time range defined by the Time Boundaries. This will assign the weights to the corresponding time ranges defined above.

Assign 1 to (<1) range
Assign 4 to (1,6) range
Assign 6 to (6,12) range
Assign 20 to (>12) range

POP3/BATCH SMTP MODULE

The Batch SMTP Tunnel Encoder provides a mechanism for the tunneling of messages for an entire organization or predefined Internet addresses, while preserving the original envelope information for each message. When messages of this type arrives at a single POP3 account, they are picked up by the POP3 Batch SMTP Decoder, which decodes and then submits them to the Internet Exchange MTA with the original envelope recipients retained. This allows the messages to be further routed until they are received by the originally intended recipients.

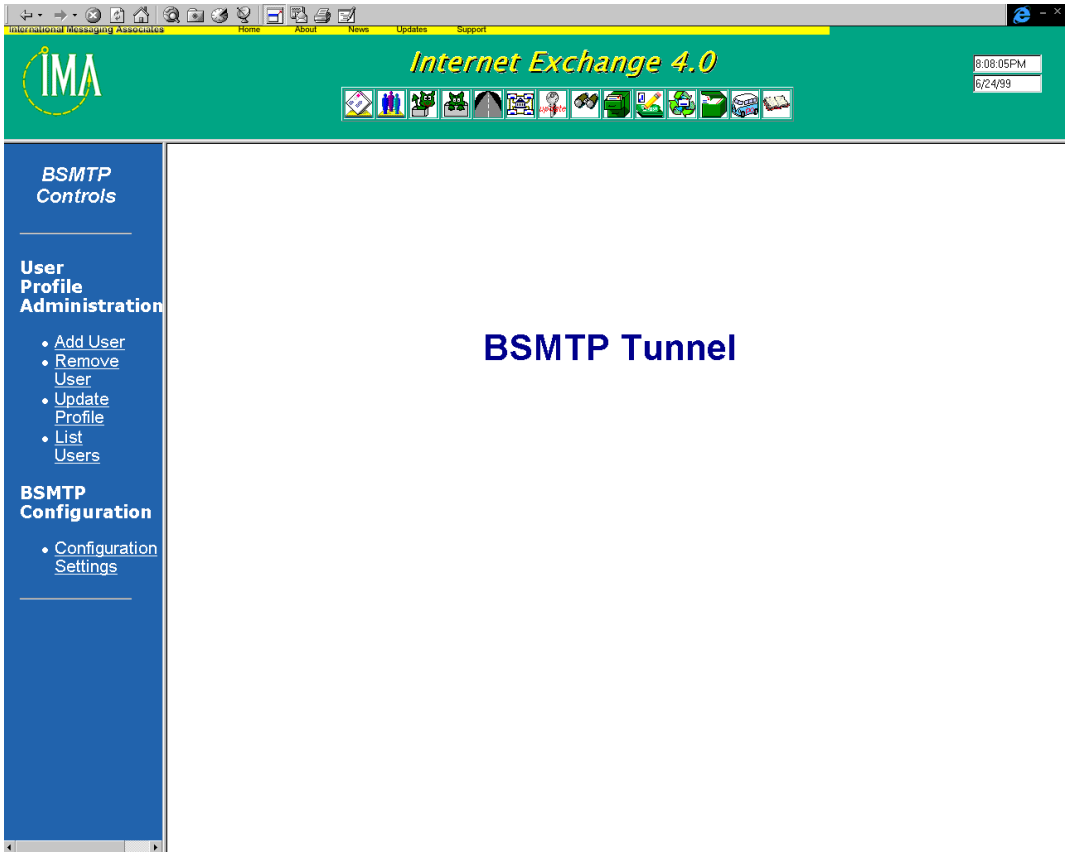

Figure 6k - Main BSMTP Tunnel Configuration Page

To configure the features of the Batch SMTP Module, go to the main Web Administration Interface and click on the *BSMTP* button. The BSMTP Tunnel screen will then appear (see Figure 6k).

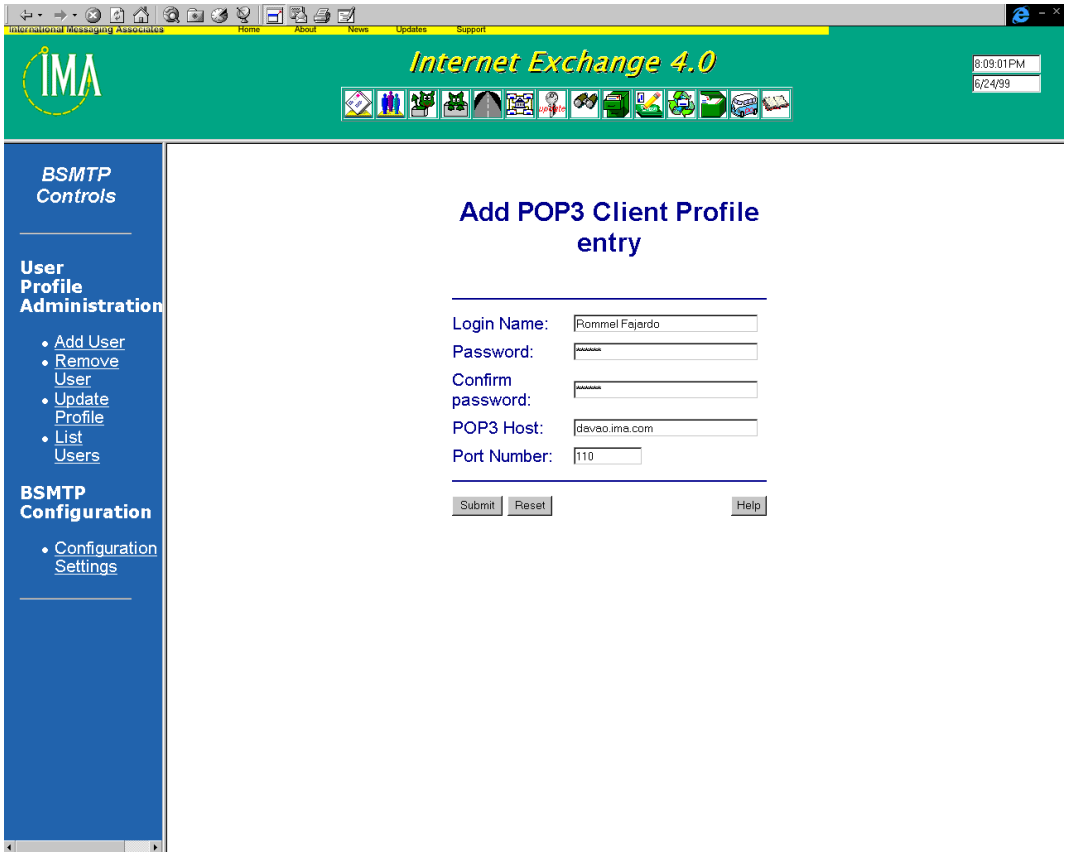

Figure 6l - Add POP3 Client

User Profile Administration

Adding POP3 Client Profiles

The Add POP3 Client screen enables the system administrator to add new remote POP3 servers that may be accessed by Internet Exchange. To add a new remote POP3 server, click on the *Add User* link on the BSMTP Tunnel screen. The Add POP3 Client screen will appear (see Figure 6l).

To add a new user profile, enter the user name for an existing account on that server. Enter user password. For security purposes, the password should be entered twice. Enter the host name and port number of the remote POP3 server to be accessed.

Click the *Submit* button to add the name of the remote POP3 server to the Batch SMTP database. Click the *Reset* button to clear all text boxes.

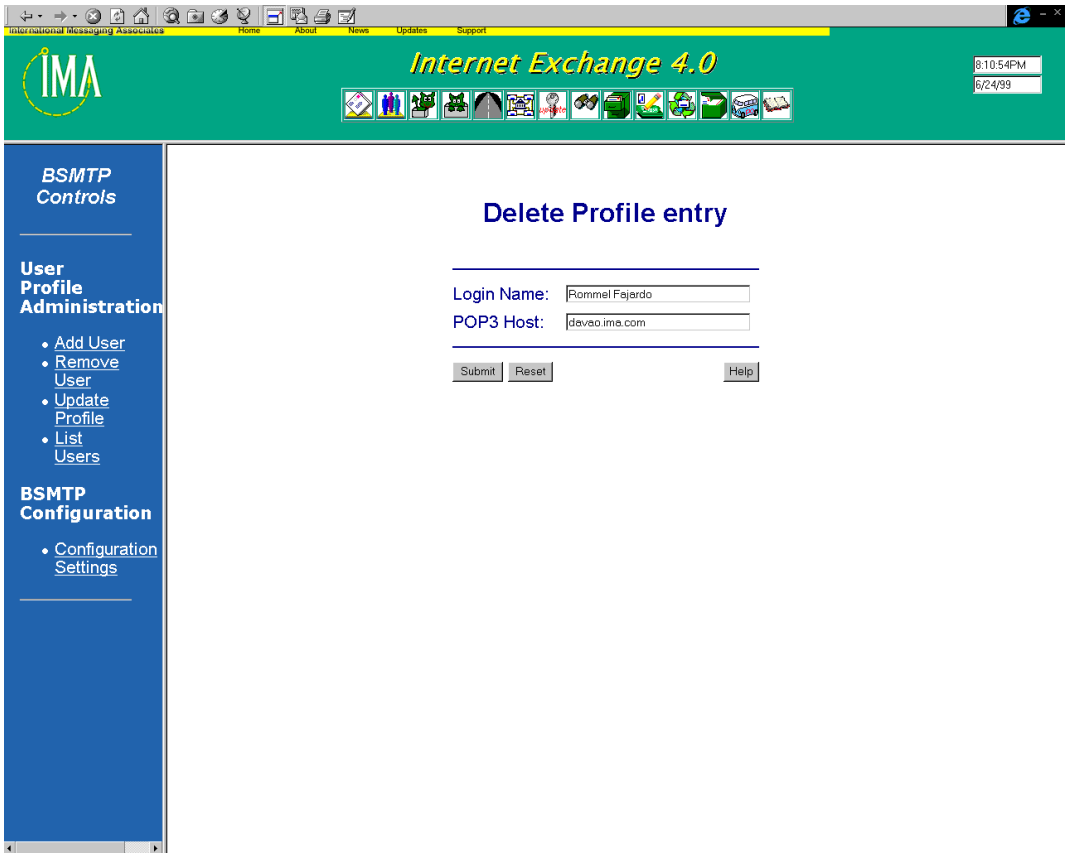

Figure 6m - Remove users

Removing POP3 Client Profiles

The Delete Profile screen (see Figure 6m) enables the system administrator to remove remote POP3 servers from the list of such servers that may be accessed by Internet Exchange. To remove a user profile, enter the user name for an existing account on that server. Enter the host name of the remote POP3 server.

Click the *Submit* button to remove the remote POP3 server from the Batch SMTP database. Click the *Reset* button to clear all text boxes.

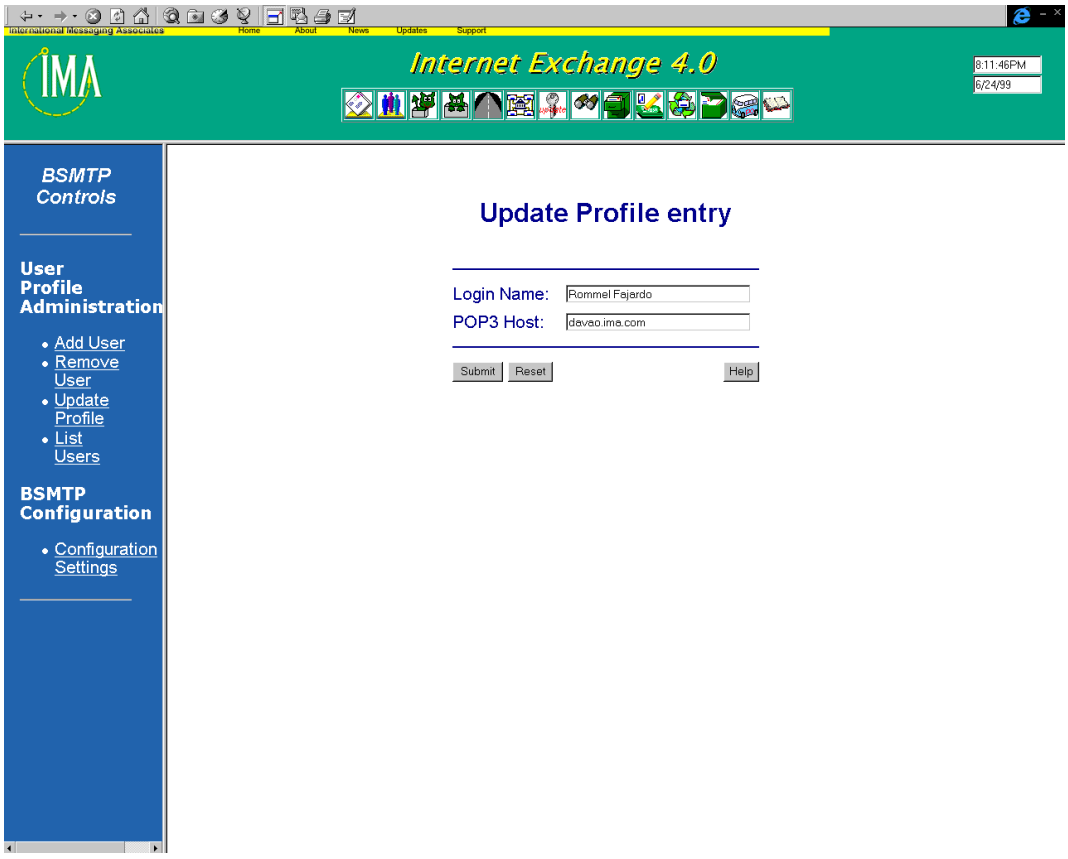

Figure 6n - Update user profiles

Updating POP3 Client Profiles

The Update Profile screen (see Figure 6n) enables the system administrator to search for a particular client profile using a login name and POP3 host combination. To update a POP3 client profile, enter the user name and the host name of the POP3 server that maintains the account.

Click the *Submit* button to bring up the Add POP3 Client Profile window (see Figure 6o), which is used for changing specific client information, such as passwords and port numbers. Click the *Reset* button to clear all text boxes.

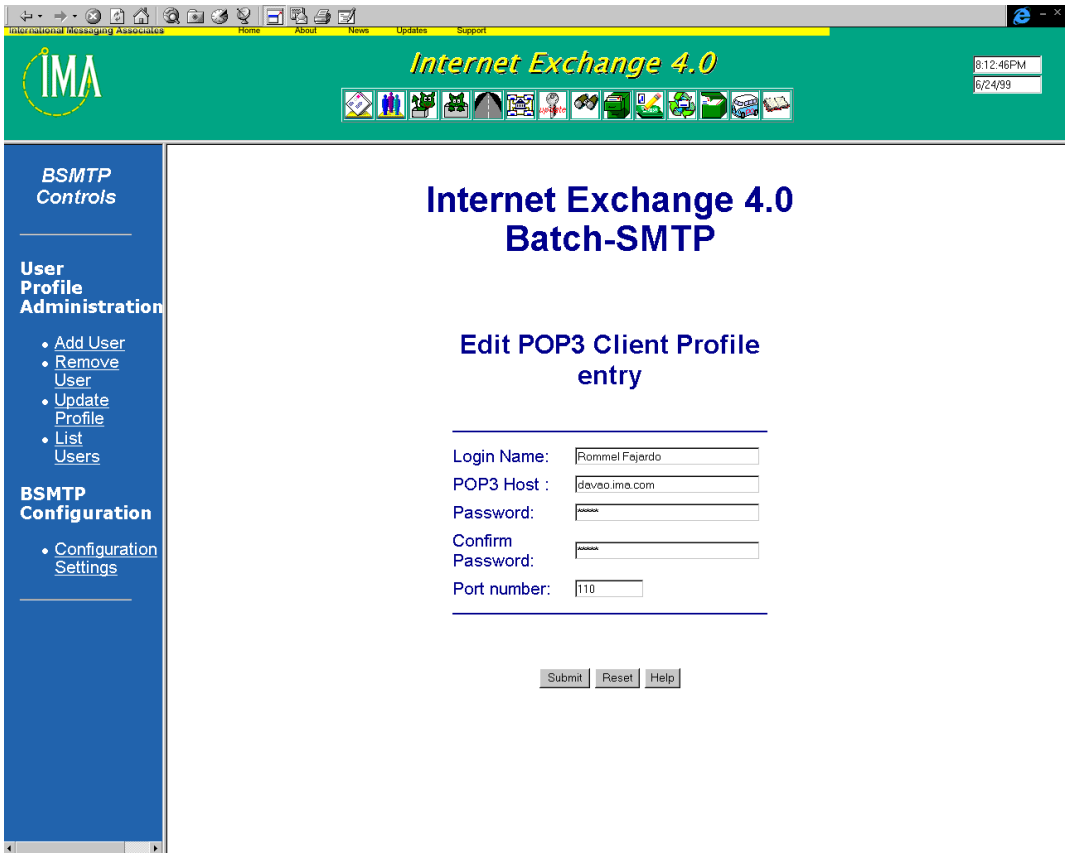

Figure 6o - Edit POP3 Client Profile

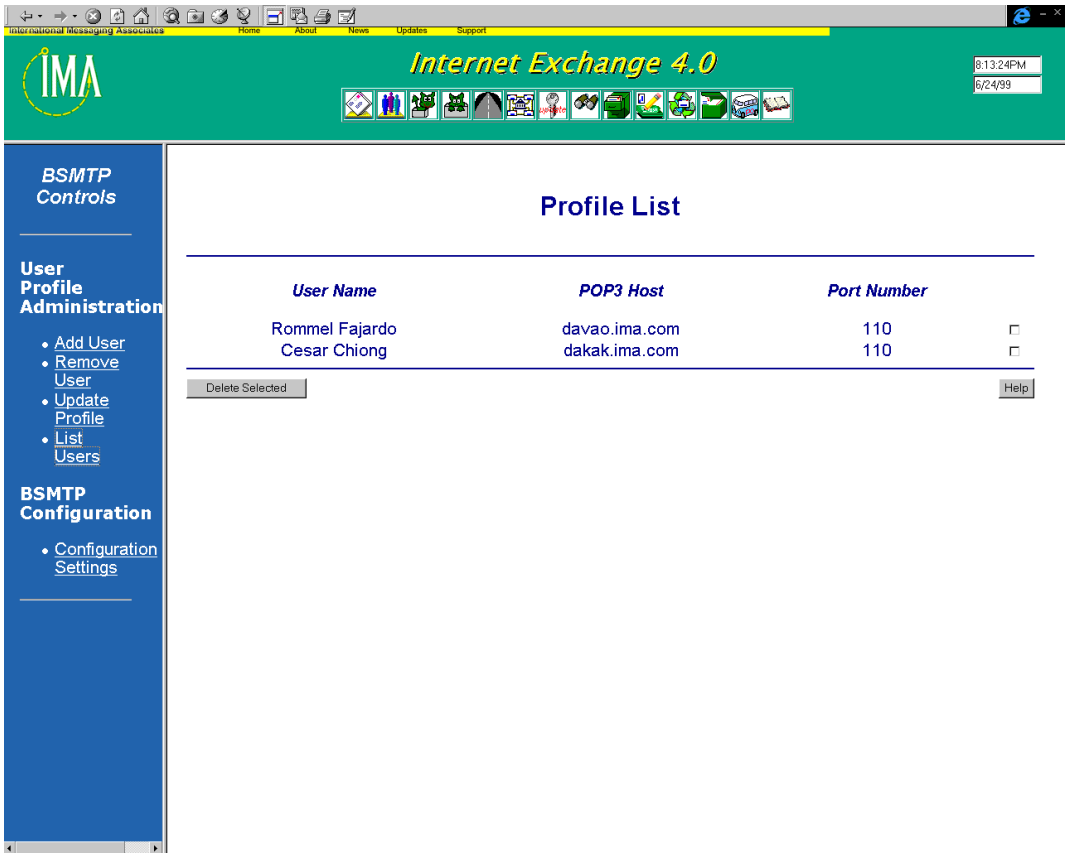

Figure 6p - List user profiles

Listing POP3 Client Profiles

The *List Users* screen (see Figure 6p) enables the system administrator to view the list of current user profiles. This screen also provides an interface to search and delete user profiles. To delete a profile, select the particular profile by enabling the check box beside the profile entry. Multiple selections of profiles to be deleted are allowed. Click on the *Delete Selected* button to remove the profiles.

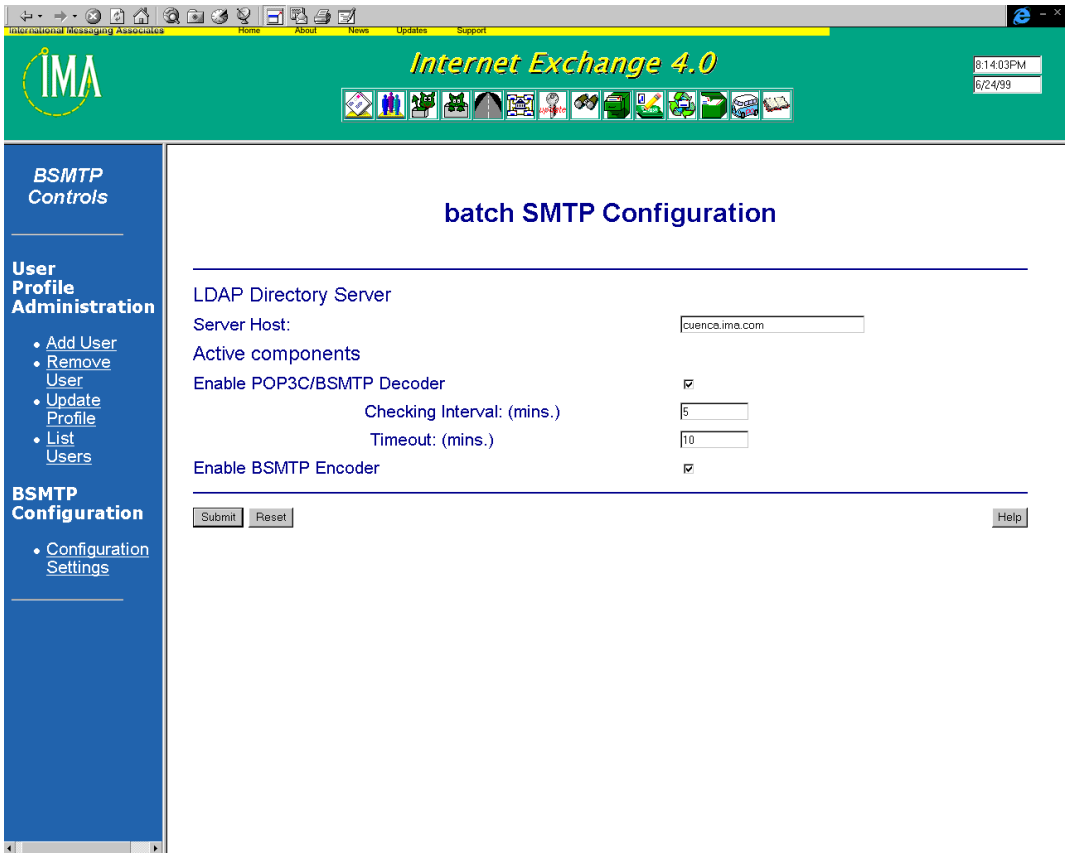

Figure 6q - Configure BSMTP

BSMTP Configuration

The BSMTP Configuration settings determines the parameters required by the BSMTP module to start both basic components: the BSMTP Processor/Decoder and BSMTP Generator/Encoder.

LDAP Directory Server

Server Host

Defines the hostname of the machine where the LDAP server is located.

Active Components

Enable POP3C/BSMTP Decoder

Set this option to enable the POP3C/BSMTP decoder.

Checking Interval

The checking interval value will determine in minutes how long the POP3C will wait before checking a POP3 server for available messages.

Timeout

Determines the timeout value in minutes of how long the POP3 Client will wait for the

POP3 servers respond for every POP3 commands.

Enable BSMTP Encoder
Activate this option to enable the BMSTP Encoder.

BSMTP Encoder

The BSMTP encoder consists of a tunneling mechanism that wraps or encapsulate messages it retrieved from the Message Queue into Application/Batch-SMTP messages. These messages are then re-injected into the Message Queue for forwarding to a specified account.

Domain Forwarding

The BSMTP Encoder incorporates the MTA forward domain table information that provides the necessary domain/mail address mapping for multiple output channel functionality. BSMTP can add, modify, and delete a table entry through a separate BSMTP Domain Forwarding Web configuration.

An example entry of the Forward Domain Table is shown below:

Domain	**Channel**	**Relay**
*.smallcorp.com	BSMTP	Bsmtp@ima.com
Othernet.org	BSMTP	Bsmtp@bignet.net

In the above example, all messages destined for smallcorp.com and its subdomains will be forwarded to bsmtp@ima.com, while messages destined for othernet.org will be routed to Bsmtp@bignet.net.

PREPROCESSOR MODULE

Internet Exchange 4's Preprocessor Unit is an integrated subsystem of the Message Transfer Agent (MTA), which features a highly scalable architecture. Each of the Preprocessor programs are plug-in modules which can be run on separate machines, ensuring efficient utilization of computing resources and maximum throughput.

To configure the features of the Preprocessor Unit, go to the main Web Administration Interface and click on the *Preprocessor* button. The Main Preprocessor Configuration screen will then appear (see Figure 6r).

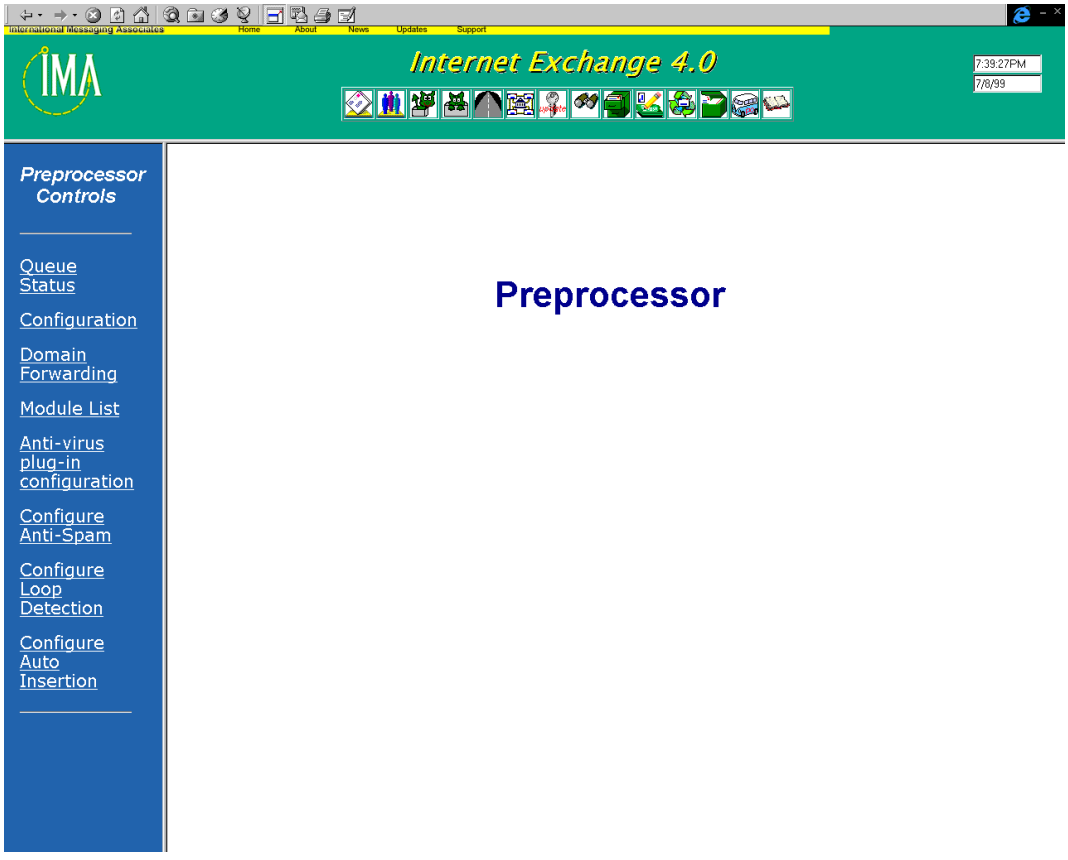

Figure 6r - Main Preprocessor Configuration Page

Queue Status

This is used to view current queue status. The following parameters are shown in tabular format (see Figure 6s):

Queue Name

The system has a number of named queues, which are listed in the file 'queue.cfg'. These queues are created when the system is installed. The system comes reconfigured with a number of queues, divided into input queues and output queues. All entries in these queues are listed in the status page as one entry.

SMTPD	Messages posted from the SMTP Daemon
BSMTPIN	Messages posted from BSMTP
CCOUT	Messages posted from the cc:Mail Connector
DLOUT	Messages posted from the Distribution List Manager
DSN	Special queue for messages using ESMTP
NOTESOUT	Messages posted from the Notes Connector

Table 6a - Input queues handled by the Preprocessor

BSMTPOUT	Messages destined for the BSMTP encoder for encapsulation
CCMAIL	Messages destined for the cc:Mail PO
DL	Messages destined for the distribution lists
LOCAL	Messages destined for users with IMAP4/ POP3 accounts
NOTES	Messages destined for the Notes Server
SMTPC	Messages destined for other Internet mail servers

Table 6b - Output queues handled by the Preprocessor

Flags
Indicates the flags set for the queue. Some connectors require a special compatibility mode. This only affects cc:Mail and Notes.

Messages
Indicates the number of messages in the queue at this moment. It is a real time figure for the messages flowing through the system.

Activity
Indicates messages processed over a given time. This number is an indication only, and a higher number means more messages per time period.

Queue Directory Disk status

Displays the total disk size and the current number of free bytes.

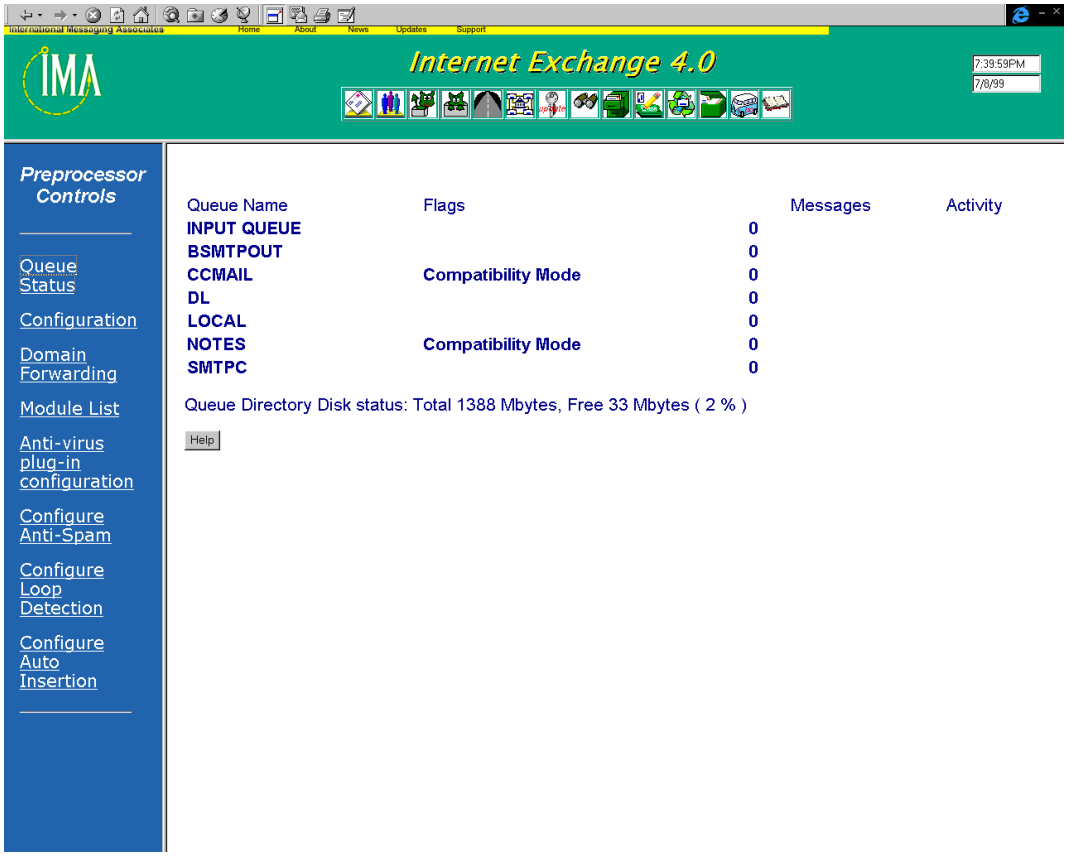

Figure 6s - View queue status

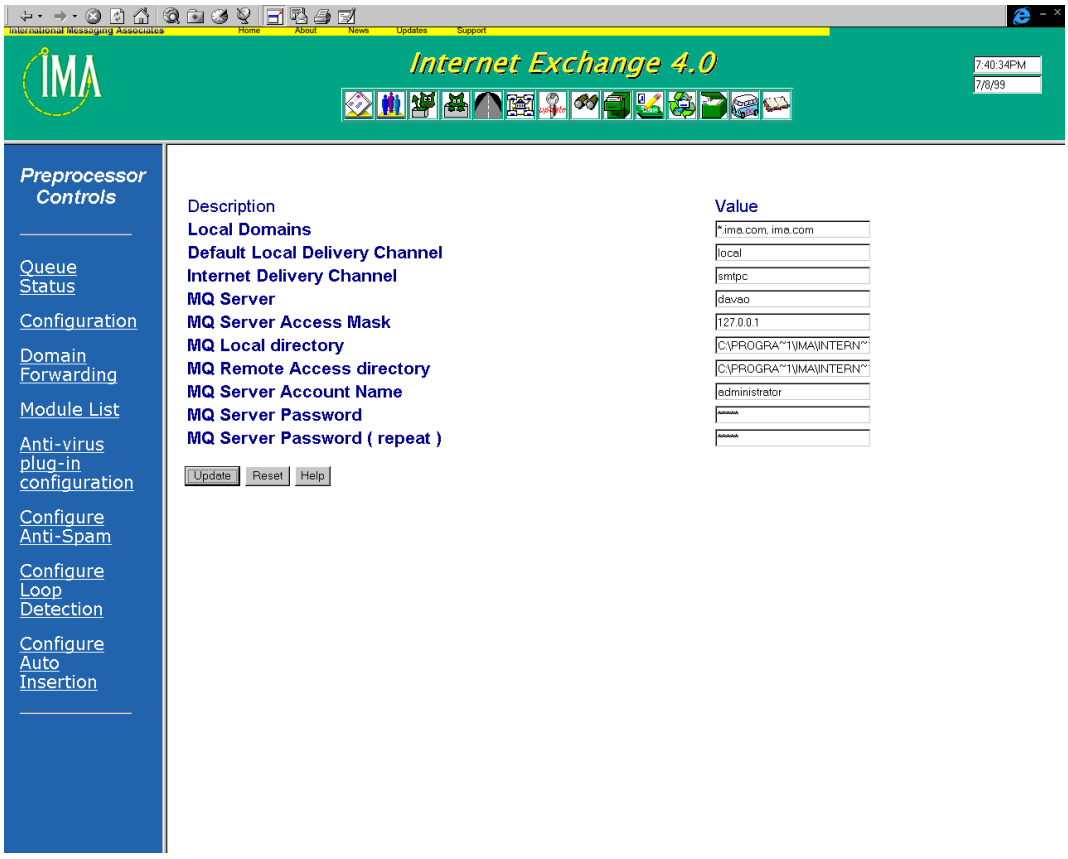

Figure 6t - Configure Preprocessor Parameters

Configuring Preprocessor

Click on the *Configuration* link to configure the Preprocessor module. You may then edit the following parameters:

Local Domains
Specifies the local host domain.

Default Local Delivery Channel
Specifies the delivery channel used locally by the Preprocessor module.

Internet Delivery Channel
Specifies the delivery channel used for the Internet.

MQ Server
Specifies the MQ server name.

MQ Server Access Mask
Specifies the MQ server access mask.

MQ Local directory
Specifies the local directory accessed by MQ.

MQ Remote Access directory
Specifies the remote directory accessed by the MQ server.

MQ Server Account Name
Specifies the account name used for the MQ server.

MQ Server Password
Specifies the MQ server password for authorization.

MQ Server Password (repeat)
Specifies the MQ server password for authorization.

Click on the *Update* button to change the current settings.

Domain Forwarding

The forward domain table information provides the necessary domain/mail address mapping for multiple output channel functionality. This table allows the adding, modification and deletion of a table entry through a Web-based interface. An example entry of the Forward Domain Table is shown below.

Domain	**Channel**	**Relay**
*.smallcorp.com	BSMTP	Bsmtp@ima.com
Othernet.org	BSMTP	Bsmtp@jade.net

In the above example, all messages destined for smallcorp.com and its subdomains will be forwarded to bsmtp@ima.com, while messages destined for othernet.org will be routed to Bsmtp@jade.net.

To go to the Domain Forwarding Configuration Page, click on the *Domain Forwarding* link. A new screen will appear (see Figure 6u).

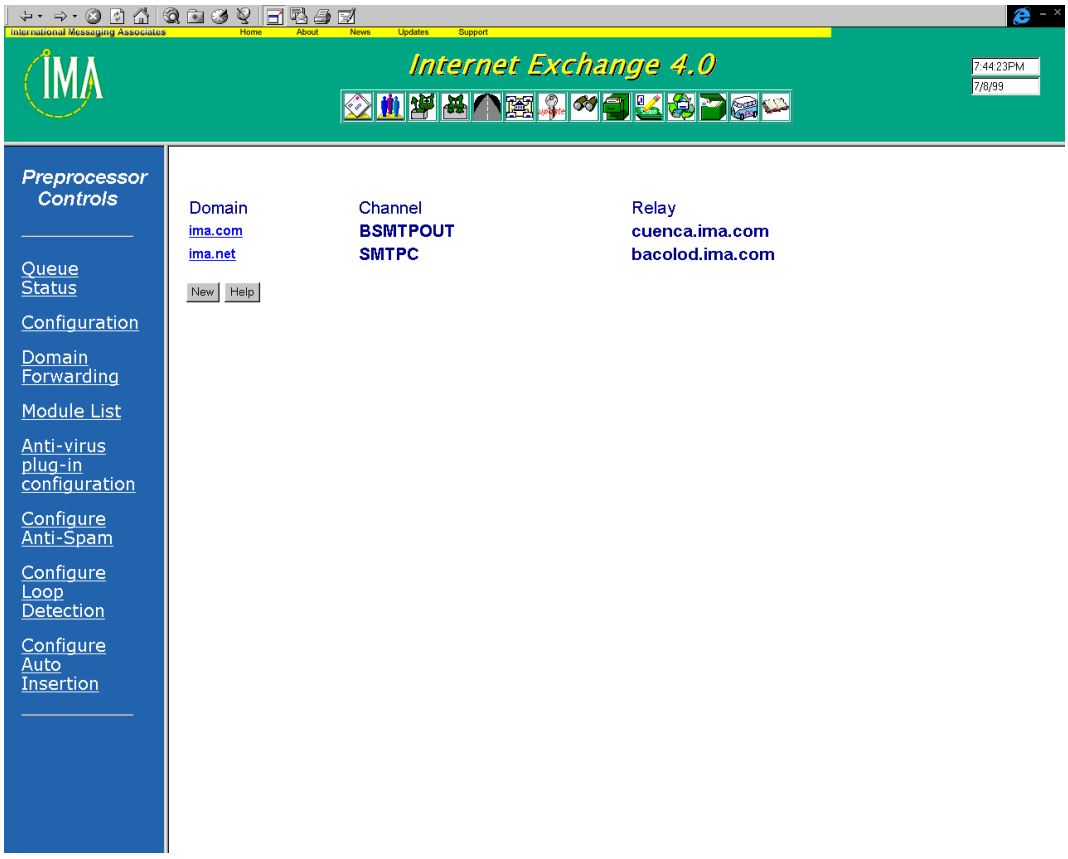

Figure 6u - Create new domains

Click on the *New* button to add a new domain mapping. In the next screen (see Figure 6v), enter the values for the following parameters:

Domain name

The second part of a fully qualified domain name (FQDN). The format should be similar to "domain.com". For example, the domain name of the FQDN *davao.ima.com* is *ima.com*.

Queue Selection

The system has a number of named queues, which are listed in the file 'queue.cfg'. These queues are created when the system is installed. The system comes reconfigured with a number of queues, divided into input queues and output queues. All entries in these queues are listed in the status page as one entry.

- BSMTPOUT - Messages to be sent to the BSMTP encoder for encapsulation
- CCMAIL - Messages for delivery to cc:Mail post office.
- DL - Messages destined for one of the Distribution lists.
- LOCAL - Messages for users with a local IMAP / POP3 account.
- NOTES - Messages destined for a Notes Server.

- SMTPC - Messages destined for other Internet Mail Servers.

Relay Host

Specifies the name of the host server. If SMTPC is unable to resolve a hostname by either DNS or host table lookup, it routes messages to this host for forwarding.

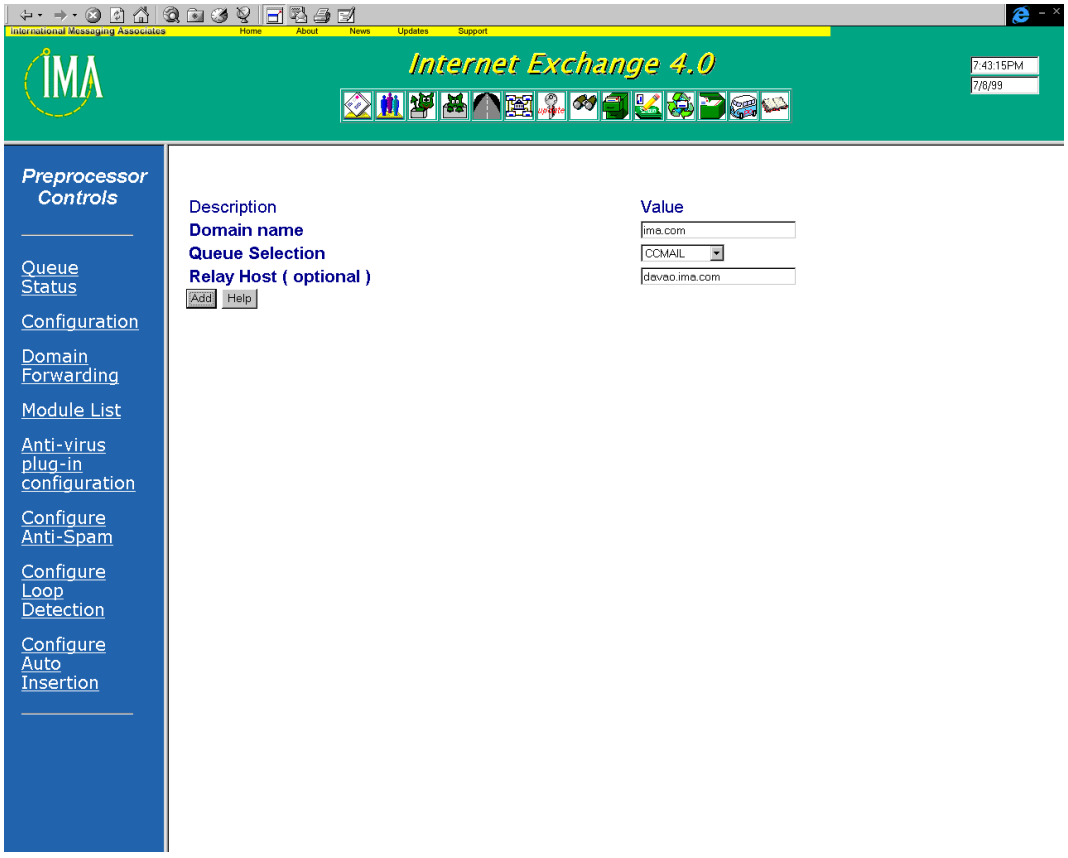

Figure 6v - Add domain mappings

Click the *Add* button to add the new entry to the domain table.

Module List

Click on the *Module List* link on the Main Preprocessor Configuration page to display the list of modules being run by the Preprocessor Unit (see Figure 6w).

Module

The module name. Each module name on the list is linked to its corresponding Channel Action Matrix. Clicking on a module name will bring you to the Channel Action Matrix for that particular module (see Figure 6x).

Filename

The full pathname of the module.

Version

This field displays the module version number.

Description

Displays a brief description of the module.

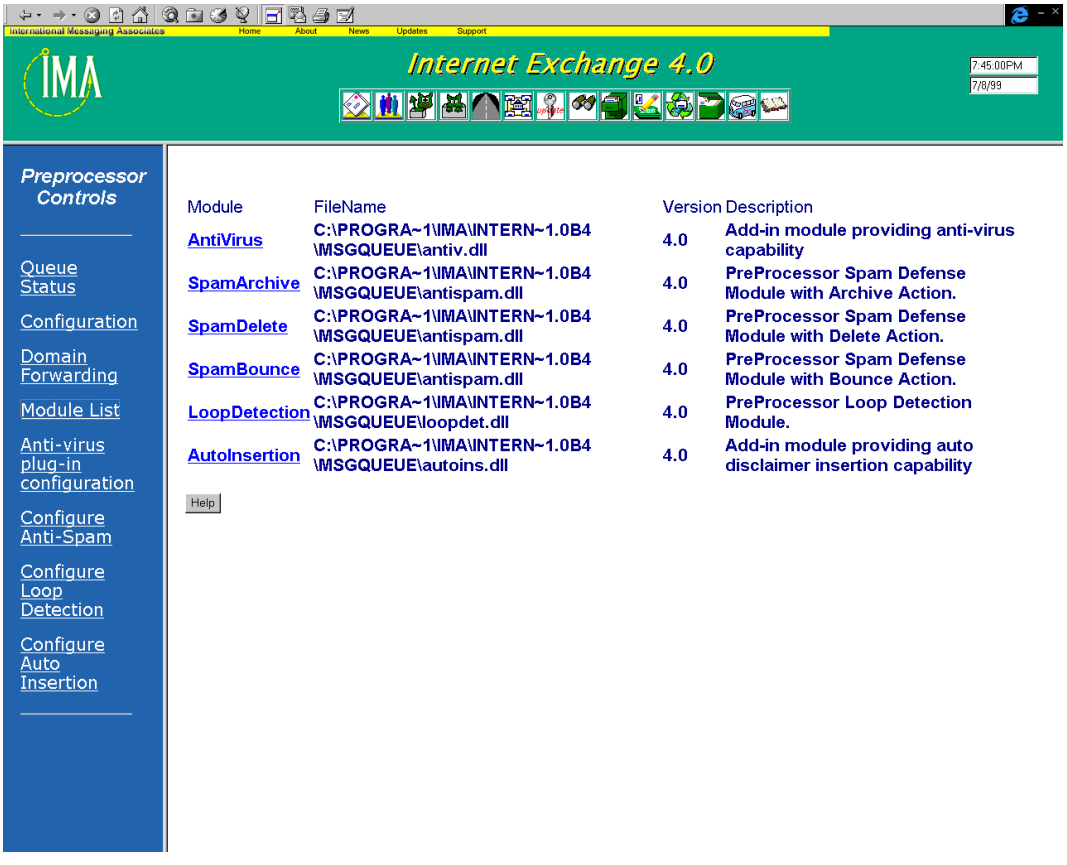

Figure 6w - Display module list

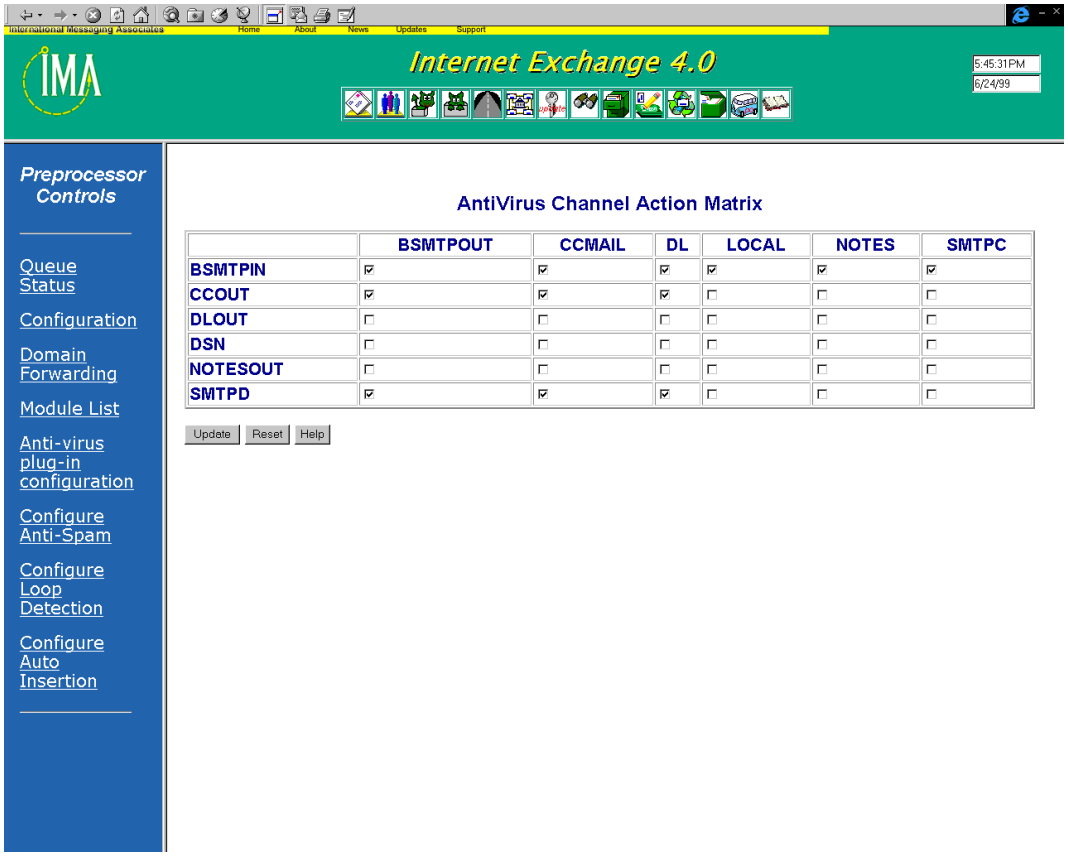

Figure 6x - Channel Action Matrix

Each module in the Preprocessor Unit includes a Channel Action Matrix for determining when to run a particular module. The figure above (Figure 6x) shows the Channel Action Matrix for the Anti-virus Module. Through the Channel Action Matrix, the system administrator is able to maximize throughput by making sure that unnecessary Preprocessor actions are avoided. For example, the system administrator may prefer not to run the Anti-virus Module for messages originating from the local cc:Mail environment and destined to users in the same environment. Or he/she may not want to run the Auto Insertion engine for messages originating from the local Notes environment. These options are easily configured using the Channel Action Matrix.

In the sample configuration shown in the figure, all messages coming from BSMTPIN and destined for BSMTPOUT, CCMAIL, DL, LOCAL, NOTES, and SMTPC are subjected to anti-virus checks by the Preprocessor Unit's Anti-virus Module. Also, based on the configuration, messages coming from CCOUT and destined for BSMTPOUT, CCMAIL, and DL are also subjected to anti-virus checks.

Anti-virus Module

Internet Exchange 4's Anti-virus Module is a 32-bit multi-threaded stand-alone pre-processing module capable of performing simultaneous virus scanning for MIME and non-MIME message attachments. Each thread created by the anti-virus engine is responsible for processing one message at a time. Whenever the Anti-virus Module receives a message, it checks the Channel Action Matrix whether it should invoke the third-party anti-virus package currently configured to run on the machine. **Internet Exchange 4's** Anti-virus Module supports the following anti-virus packages:

- *McAfee VirusScan* - this software engine supports the following platforms: DOS, Windows 95, Windows 98, and Windows NT.
- *Sophos Anti-Virus for Windows 95/98* - this application has the capability to automatically eliminate many common viruses and can easily be installed. It can be updated monthly with the latest anti-virus technology via the World Wide Web or via a CD or floppy disk.
- *Sophos for Windows NT* - this application is specifically designed for the Windows NT platform and has the same features found in Sophos Anti-Virus for Windows 95/98.
- *F-PROT Professional Anti-Virus Package* - this is specifically designed to support Windows 95/98 and Windows NT 4.0 (Server/Workstation).

The Anti-virus Module supports the following encoding methods:

- BASE64
- Quoted-Printable
- 7Bit
- 8Bit
- UUENCODE
- Binhex
- AppleSingle
- AppleDouble
- Non-MIME encoded UUENCODE/Binhex
- Embedded UUENCODE/Binhex in MIME Text item

To configure the various features of Anti-Virus Module, go to the main Preprocessor Configuration screen and click on the *Anti-Virus plug-in configuration* link. The configuration screen for the Anti-Virus module will appear (see Figure 6y).

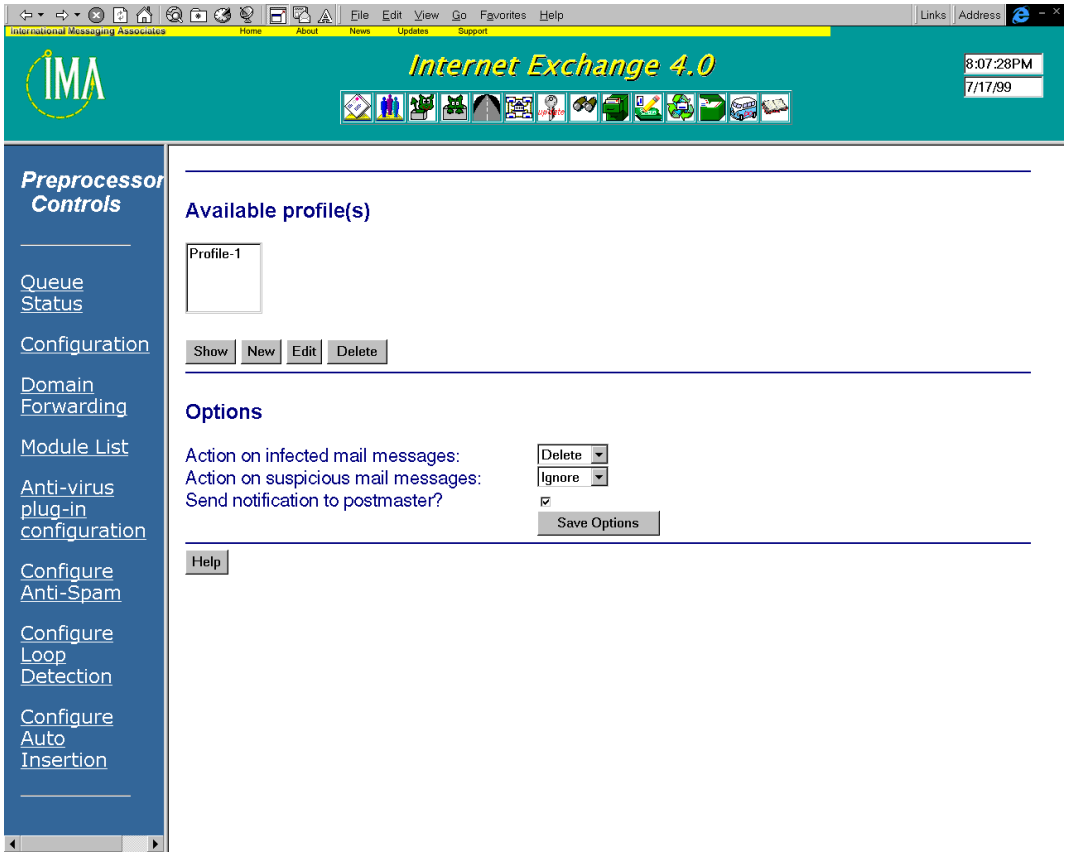

Figure 6y - Configure Anti-virus options

View Profile

To view an existing anti-virus profile, select that particular profile and click on the *Show* button. A new screen displaying the attributes of that anti-virus profile will be displayed (see Figure 6z).

Create Profile

To create a new profile, click on the *New* button. The same screen as shown in Figure 6z will appear but with blank fields.

Edit Profile

Select an existing profile and click on the Edit button. A new screen for modifying the attributes of that profile will appear (see Figure 6z)

Delete Profile

Select an existing anti-virus profile and click on the *Delete* button to remove that profile.

Options

Action on infected mail messages

Enables the system administrator to determine what to do with virus-infected messages. Such messages may either be deleted, bounced to the sender, or archived.

Action on suspicious mail messages
The Anti-virus Module can either ignore messages that are suspected to be virus-infected or bounce them back to the sender.

Send notification to Postmaster
If enabled, the Anti-virus Module will notify the Postmaster whenever messages are bounced, deleted, archived, or ignored by the Anti-virus Module as configured by the system administrator.

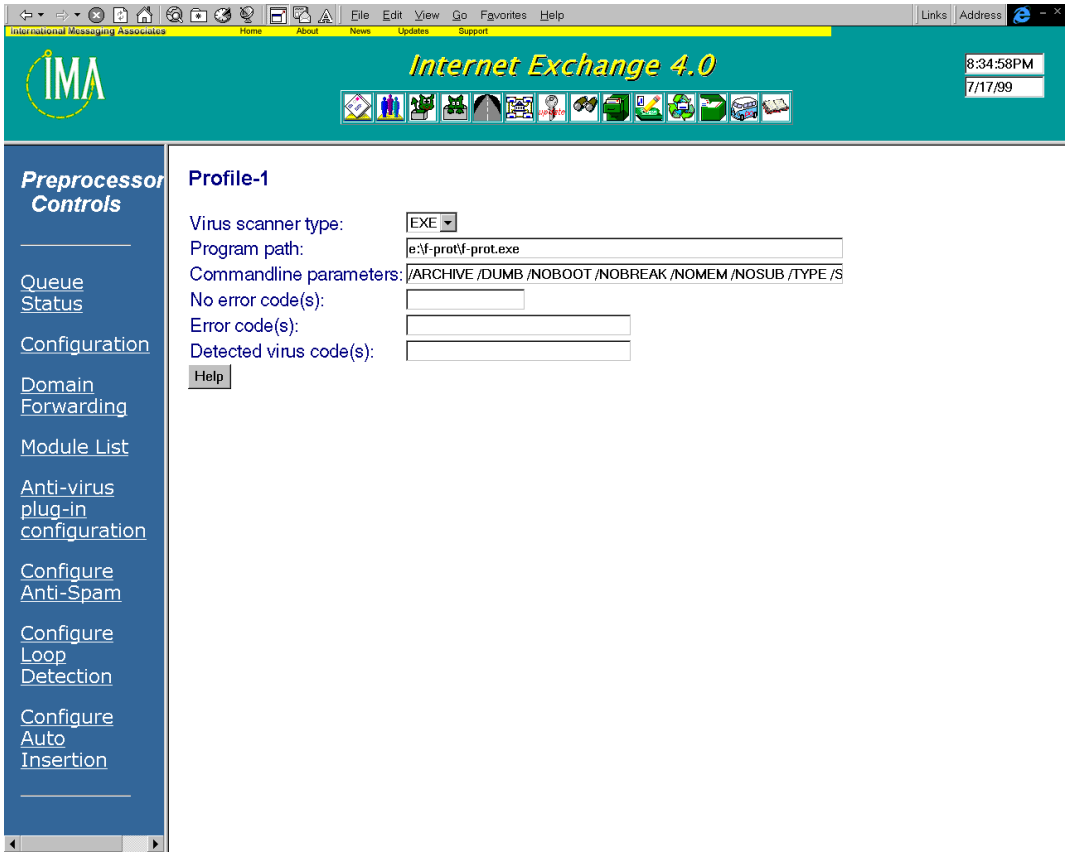

Figure 6z - Add/edit new virus scanner

Virus scanner type
The type of anti-virus software installed on the machine. The Anti-virus Module runs this software to scan messages for viruses.

Program path
The full path name of the directory/folder where the anti-virus software's executable (*.exe) file resides.

Command line parameters
The required and optional parameters that prompts the Anti-virus Module to substitute a temporary filename to that of the virus scanner (*consult your anti-virus software's manual*

for details).

No error code(s)
The DOS error codes that indicates virus-free conditions (*consult your anti-virus software's manual for details*).

Error codes
The DOS error codes that indicates scanning errors have occurred (*consult your anti-virus software's manual for details*).

Detected virus code(s)
The DOS error codes that indicates a virus has been detected (*consult your anti-virus software's manual for details*).

Example
Configure Anti-virus module to use DOS/Console based virus scanner
The F-PROT Professional Anti-Virus Package is used in the following example:

1. Make sure that the selected virus scanner is successfully installed in the system. Consult the appropriate manual for installation procedure.

2. Start the web browser.

3. Go to the main IEMS configuration page.

4. Click on the *Anti-virus plug-in configurations* link.

5. The Anti-virus screen will then appear (Figure 6z).

6. Select *EXE* for virus scanner type.

7. Put the full path name of *f-prot.exe* in the Program path entry.
8. Enter the required and optional parameters in the *Commandline parameters* field, make sure %f is at the end of the string. This prompts the Anti-virus plug-in to substitute a temporary file name to that of the virus scanner.

9. Enter DOS error code(s) that indicates no error (virus-free) condition. Consult the manual of the selected virus scanner for a list of these values.

10. Enter DOS error code(s) that indicates error condition. Consult the manual of the selected virus scanner for a list of these values.

11. Enter DOS error code(s) that indicates "virus is detected" condition. Consult the manual of the selected virus scanner for a list of these values.

12. Go to the screen for configuring anti-virus options (see Figure 6y).

12. Select *Archive*, *Delete* or *Bounce* for the *Action on infected mail messages* entry.

13. Select *Ignore* or *Bounce* for the *Action on suspicious mail messages* entry.

14. Click the *Send notification to postmaster* check box to receive notification messages.

15. Click *OK* to submit the settings to the configuration CGI program. The CGI program will validate the settings and gives error messages if:

    - Program path is not correct
    - CommandLine parameters does not contain “%f”

Anti-Spam Module

The Anti-spam module of **Internet Exchange 4** provides the administrator with options to control the reception of unsolicited and unwanted SPAM mail messages. In addition to providing control over what sites can use **Internet Exchange 4** as mail relay, the system can be defined to reject mail during the SMTP exchange from:

- Any number of host and domains
- IP 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:

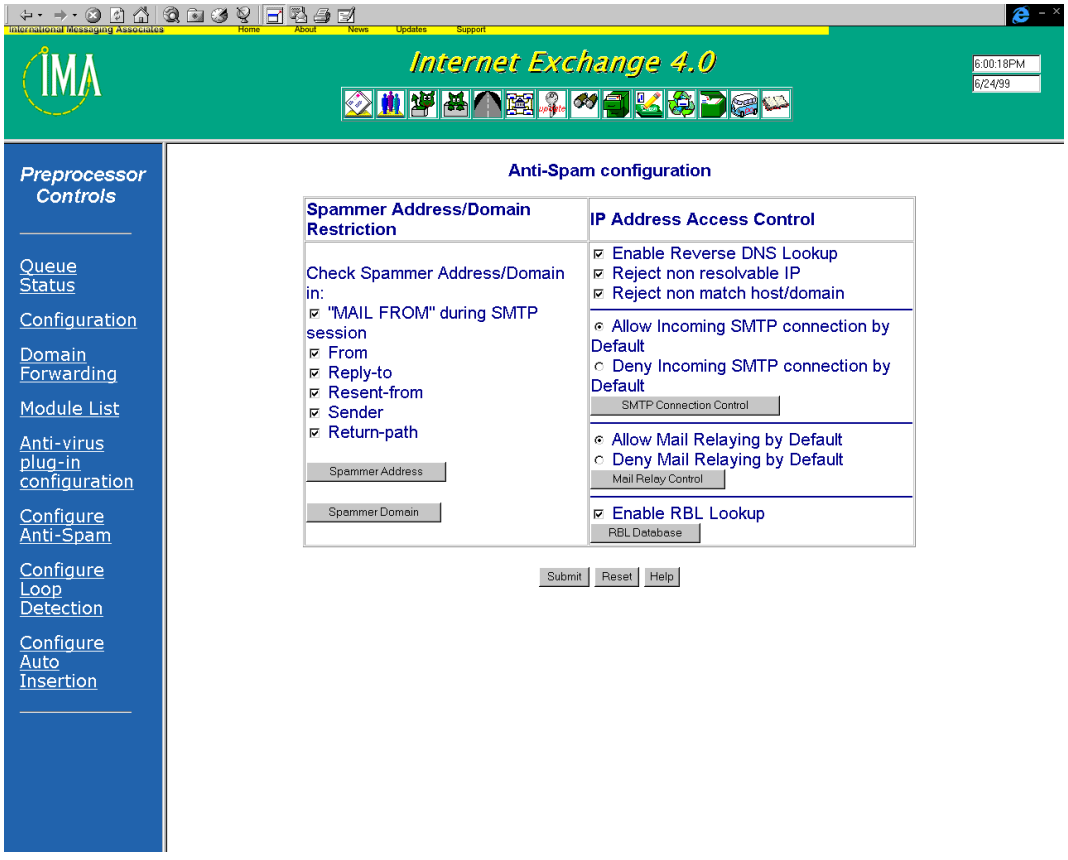

Figure 6aa - Configure Anti-Spam options

To configure AntiSpam, click on the *Configure Anti-Spam* link at the left side of the Main Preprocessor Configuration screen. The Anti-Spam configuration screen will appear (see Figure 6aa). You may then configure the various anti-spam parameters:

Blocking a Spammer's Address/Domain

Internet Exchange 4 features several Web-based interfaces for blocking a known spammer address/domain. To activate Internet Exchange's anti-spam capabilities, go to the screen shown above and select the message parameters by which a spammer's address/domain can be matched during the SMTP session. For example, if the options *MAIL FROM* and *From* are checked, the Anti-spam Module will scan for the spammer's address/domain in the *MAIL FROM* and *From* headers of the RFC822 message.

MAIL FROM" during SMTP connection
SMTPD scans any spammer address or domain during the "MAIL FROM" session. If enabled, SMTPD returns a 553 error to the remote sendmail host if a match is found.

From
Scans any spammer's address or domain in the RFC822 message "From" header.

Reply-To
Scans any spammer's address or domain in the RFC822 message "Reply-to" header.

Resent-from
Scans any spammer's address or domain in the RFC822 message "Resent-from" header.

Sender
Scans any spammer's address or domain in the RFC822 message "Sender" header.

Return-path
Scans any spammer's address or domain in the RFC822 message "Return-path" header.

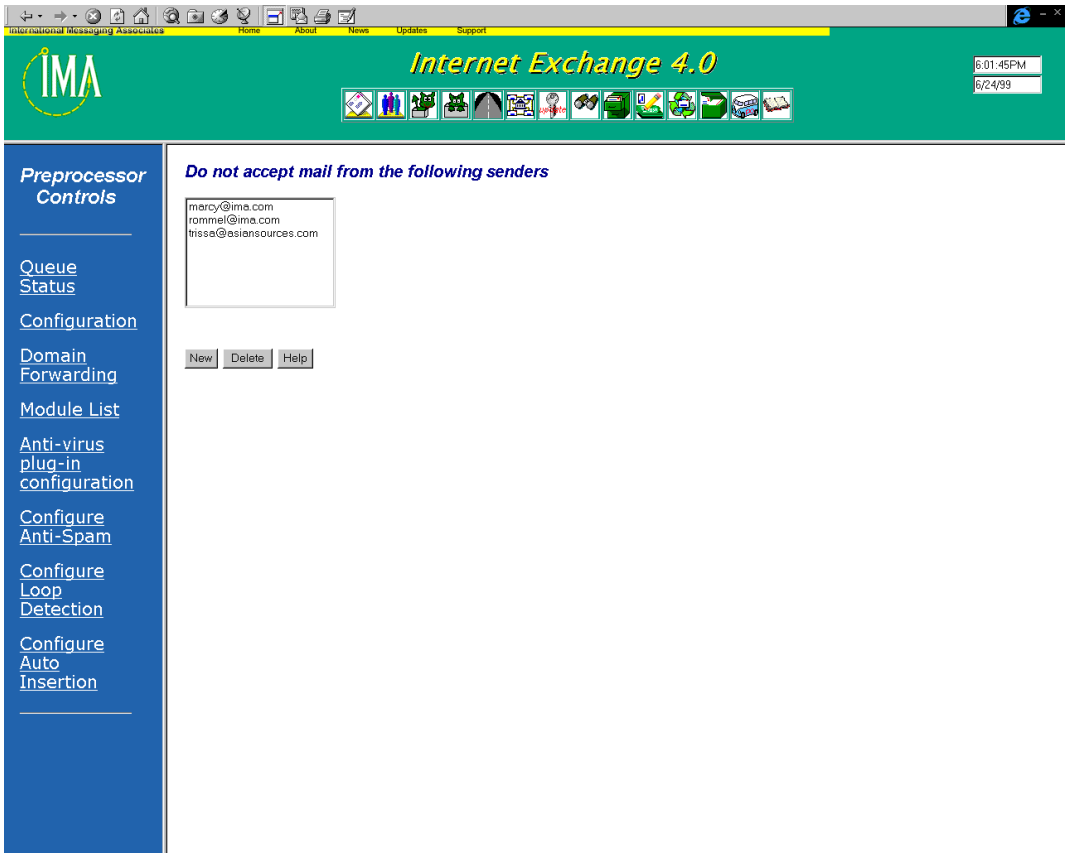

Figure 6bb - Screen showing list of banned addresses

Adding a Spammer's Address/Domain to the Banned Users List

After activating the desired options, click on the *Spammer Address* button to add the address of a known spammer to the Banned Users List. The screen shown in Figure 6aa will appear.

To add a new address to the Banned Users List, click the *New* button. A new screen for adding spammer addresses will appear (see Figure 6cc).

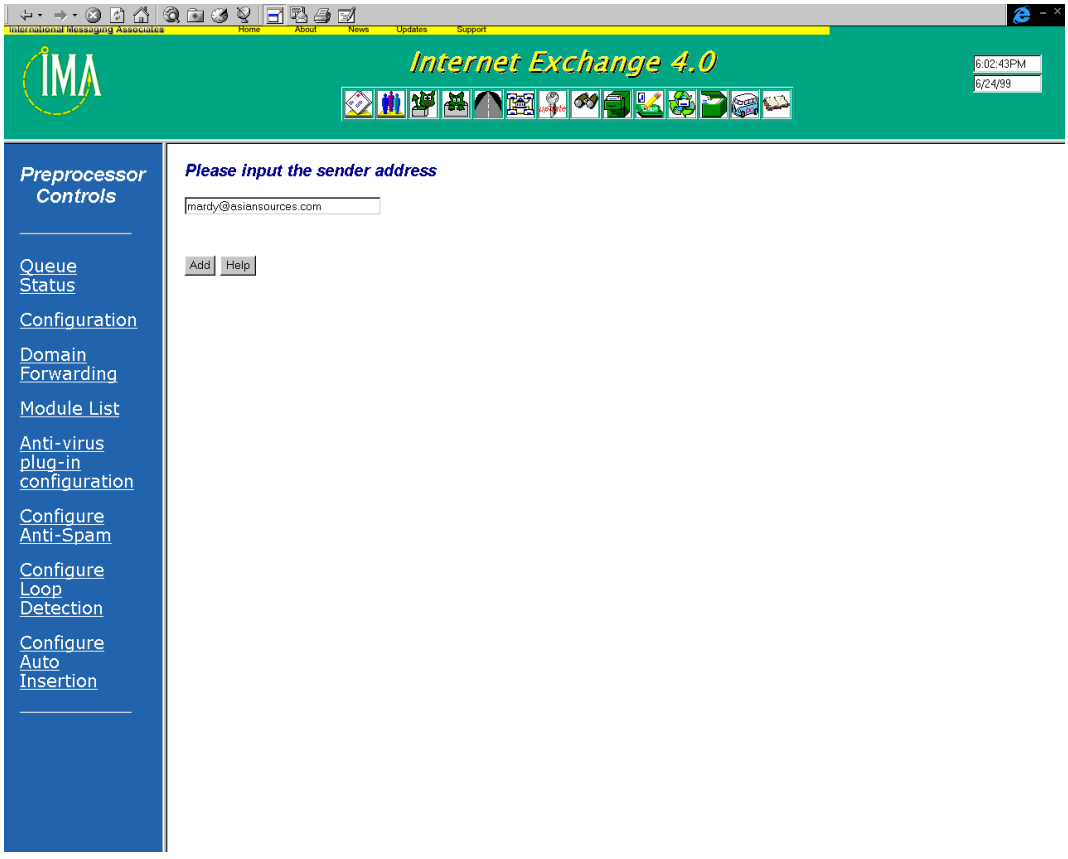

Figure 6cc - Add new address to the list of banned users

Enter the address of the spammer and click on the *Add* button. This will add the spammer's address to the list of users who are banned from accessing Internet Exchange.

To remove a sender from the banned users list, select the name of the sender (see Figure 6bb) and click on the *Delete* button.

Click on the *Spammer Domain* button to add/delete/edit peer domains. A new screen (see Figure 6dd) will appear:

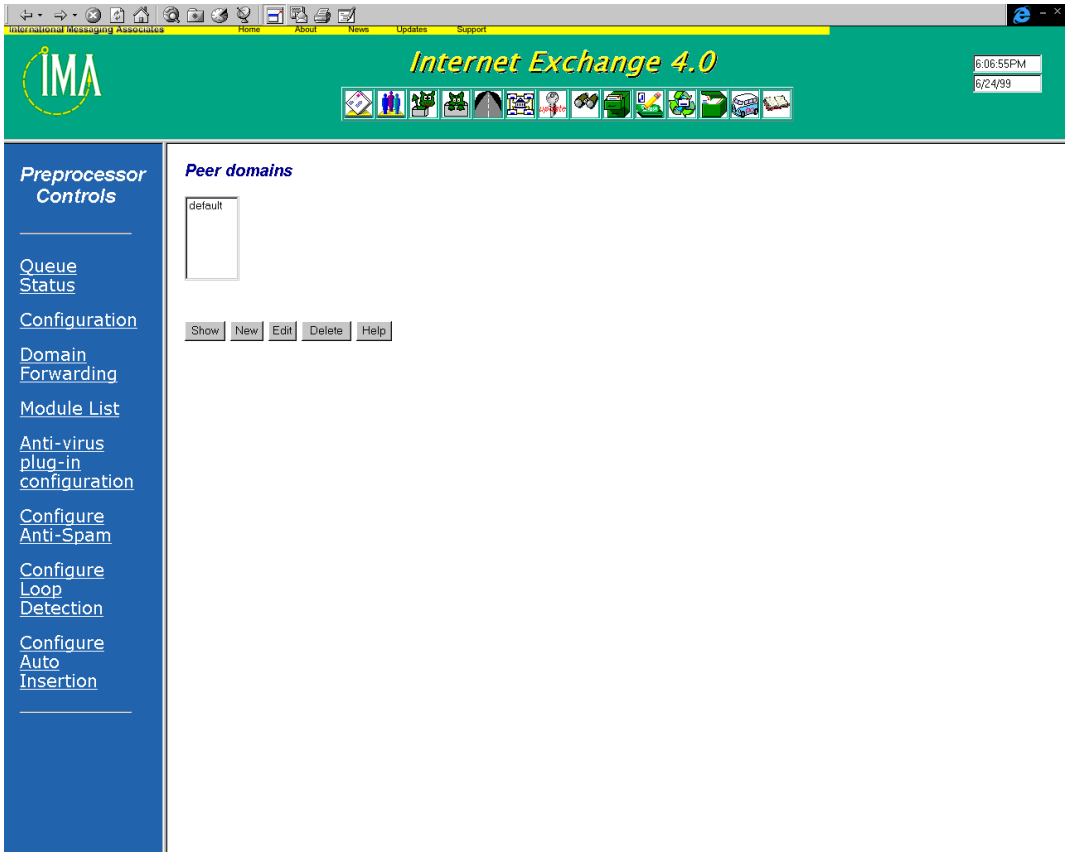

Figure 6dd - Add/delete/edit peer domains

To view an existing peer domain, select an entry from the list box. Click on the *Show* button and a new screen for modifying the peer domain's various options will appear (see Figures 6ee.1 and 6ee.2).

To edit an existing peer domain, select an entry from the list box. Click on the *Edit* button and a web-based interface (see Figures 6ee.1 and 6ee.2) for modifying the peer domain's various options will appear.

To add a new peer domain, click on the *New* button and the web-based interface (see Figures 6ee.1 and 6ee.2) for creating a peer domain and configuring its various options (i.e. Domain Name, SMTP Connection, SMTPC Profile, Native Attachment Encoding, etc.) will appear.

To remove an existing peer domain, select an entry from the list box. Click on the *Delete* button on the screen.

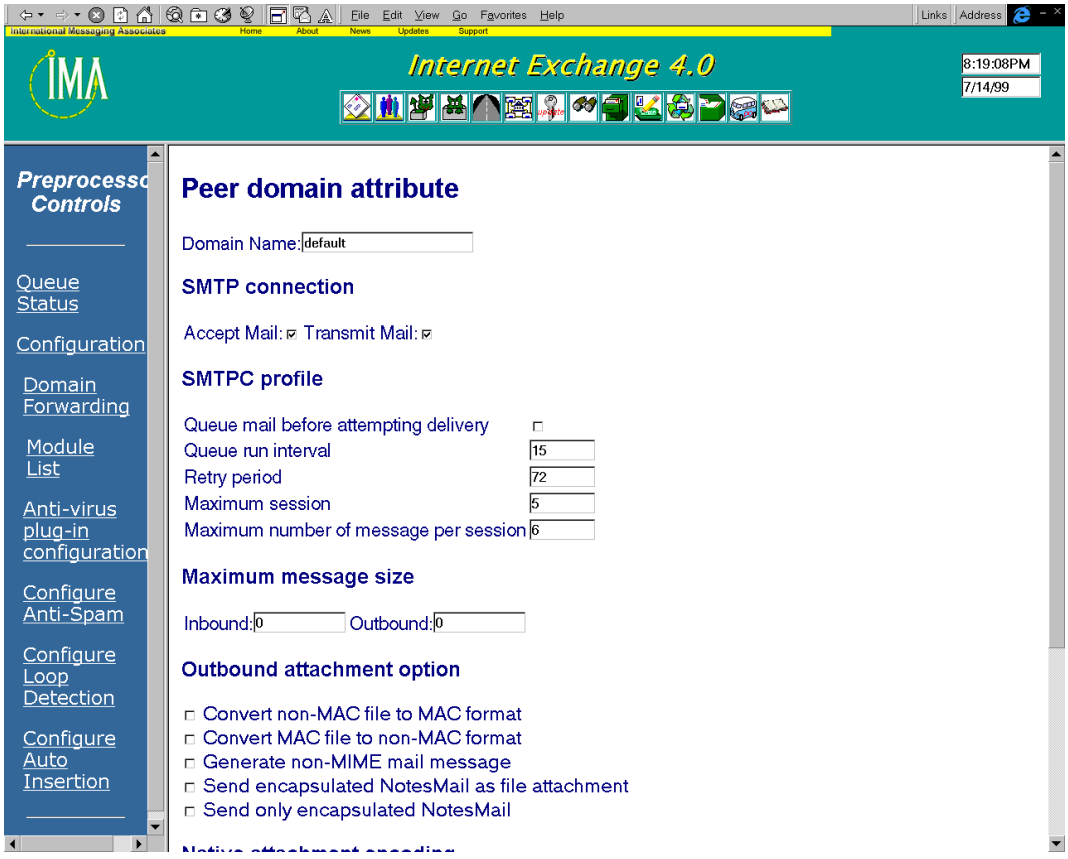

Figure 6ee.1 - Web interface for adding/editing peer domains

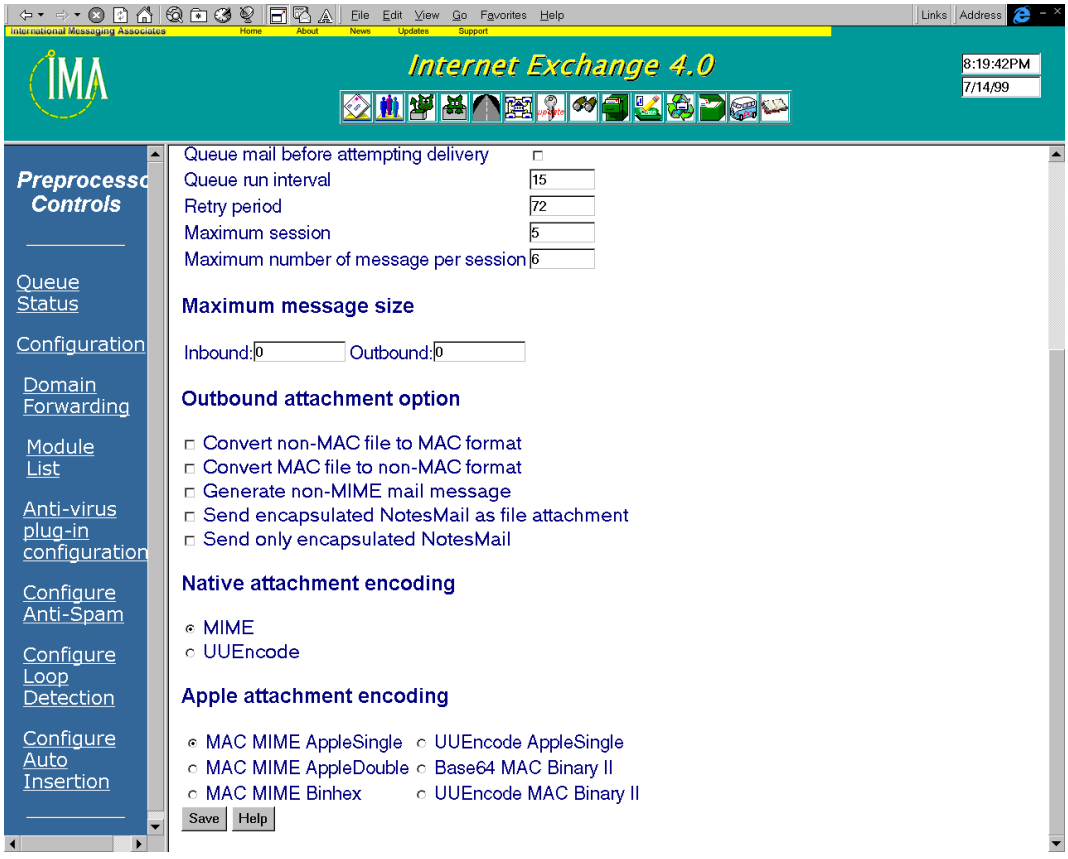

Figure 6ee.2 - Web interface for adding/editing peer domains

NOTE: For a detailed explanation of the different parameters on Figure 6dd.1 and 6dd.2, please refer to the section on SMTP Domain Profile on page 6-17.

IP Address Access Control

The following parameters can be configured by in the main Anti-spam configuration screen (see Figure 6aa).

Enable Reverse DNS lookup
By activating this option, reverse DNS lookup is enabled. During the HELO/EHLO session, the SMTP client identifies itself to the SMTP server (SMTPD) through the HELO/EHLO parameter. The SMTP server verifies if the domain name corresponds to the IP address of the SMTP client host by performing Reverse DNS lookup. RFC1123 states that the SMTP server must not reject any SMTP connection even if SMTP client's HELO/EHLO command fails verification. However, this restriction may lead to SPAM messages being generated from spoof sites.By default, this option is disabled.

Reject Non Resolvable IP
When enabled, SMTPD rejects the connection if the incoming IP address is non-resolvable, which means that the DNS server/mail relay host cannot resolve the IP address. By default, this option is disabled.

Reject Non Match Host/Domain
When enabled, SMTPD matches the resolved domain name with the one declared by SMTP client. If the two do not match, connection is denied.

Allow/Deny Incoming SMTP connection by default
If this option is selected, SMTPD accepts every IP address except for those mentioned in the Deny IP address list. On the other hand, if “Deny Incoming SMTP connection by default is selected, every IP address except for those mentioned in the Allow IP addresses list is rejected. By default, this option is set to *Allow Incoming SMTP connection.*

Allow/Deny Mail Relaying by default
If this option is selected, SMTPD allows mail relaying for all IP addresses except for those mentioned in the Deny IP address list. On the other hand, by selecting Deny Mail Relaying by default, every IP address except for those mentioned in the Allow IP addresses list is prohibited for mail relaying. By default, this option is set to *Allow Mail Relaying.*

NOTE: It is strongly recommended that mail relaying on Internet connected servers be disabled in order to protect the site from unauthorized use by spammers.

SMTP Connection Control

To enable this option, click on the *SMTP Connection Control* button on the main Anti-spam configuration screen. A new page for viewing denied IP addresses will appear (see Figure 6ff).

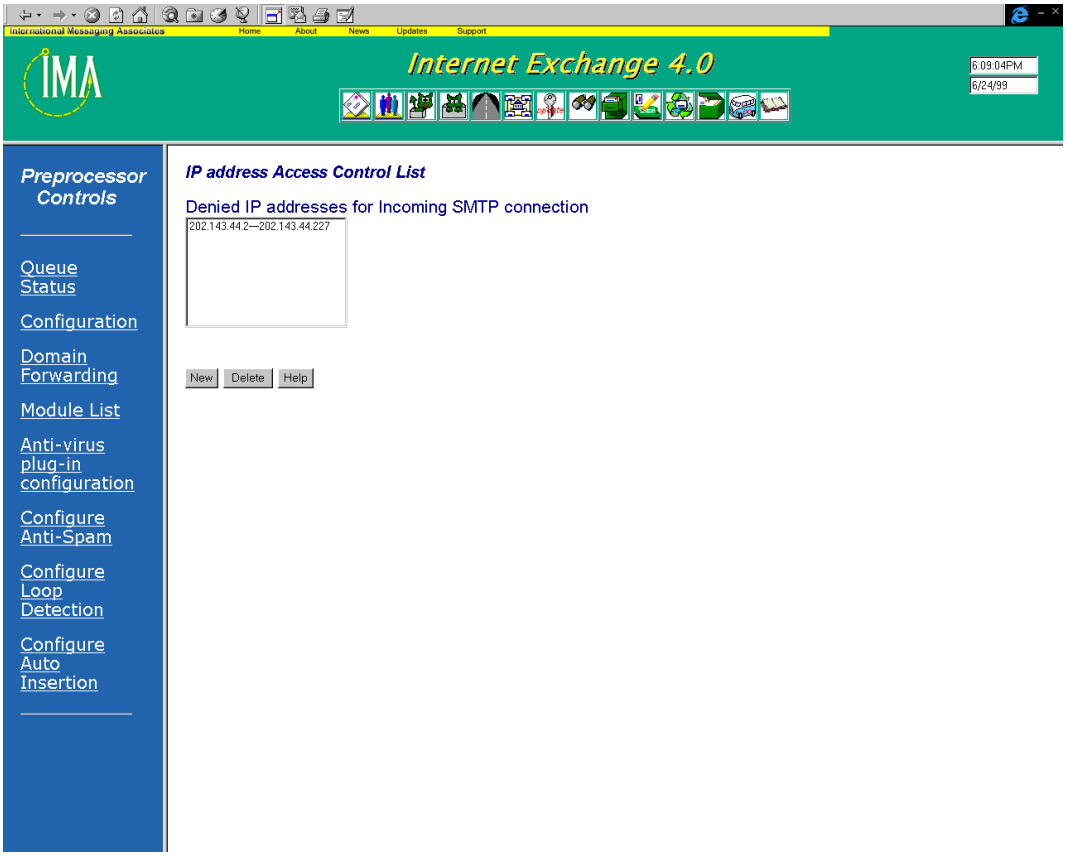

Figure 6ff - View banned IP addresses

To remove an existing entry, select that particular entry and click on the *Delete* button.

Click the *New* button to add a new IP address range. A new screen will be displayed (Figure 6gg). Enter the IP address range and click on the *Add* button to include that particular IP range in the list of banned IP addresses.

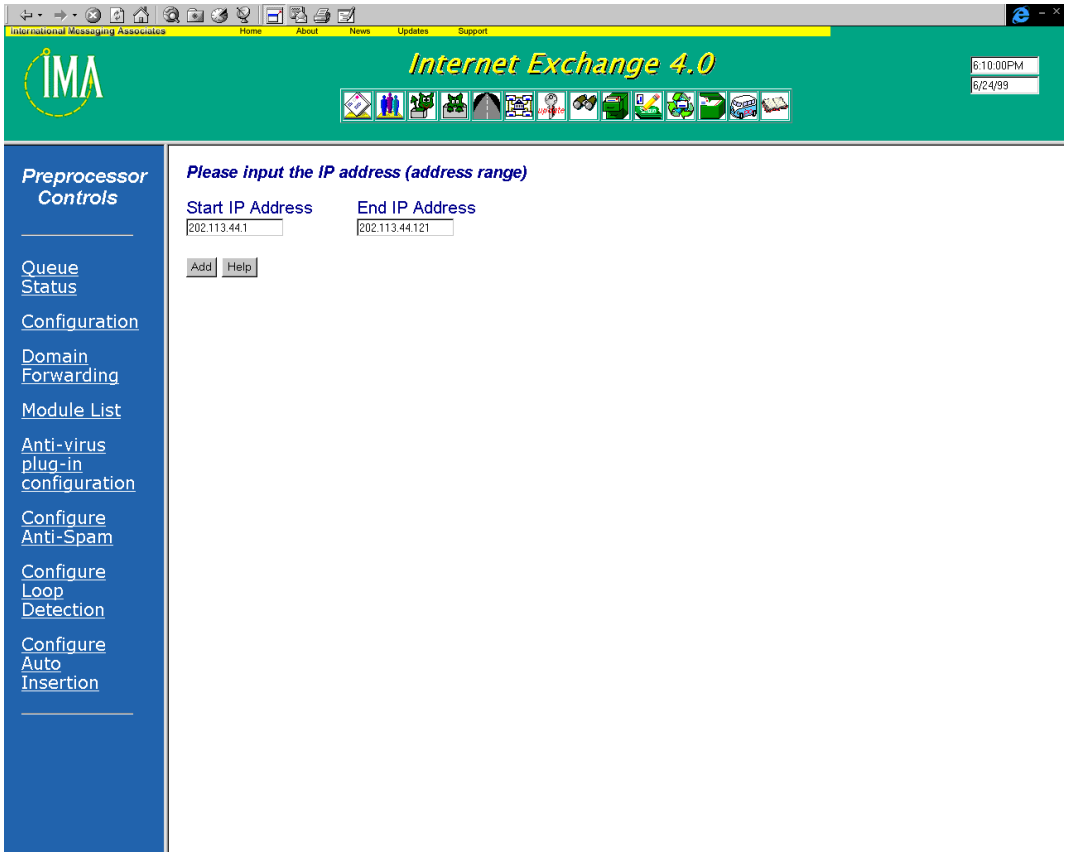

Figure 6gg - Add banned IP addresses

Enable RBL Lookup

An RBL (Real-time Blackhole List) is a blacklist of Internet IP addresses that are known to send Spam mails, be friendly to Spammer, or be totally open to mail relaying. They utilize DNS to distribute the blacklist IP database, as via DNS a record lookup of the incoming IP address under a particular DNS zone. If this option is selected, the Anti-spam module will have additional spam mail detection capability.

Internet Exchange currently supports five RBL-style systems. They are:

- MAPS-RBL (Mail Abuse Preventions System's Real-time Blackhole List)
  For more information go to http://maps.vix.com/rbl/

- ORBS (Open Relay Behavior-modification System)
  For more information go to http://www.orbs.org

- MAPS-DUL (Mail Abuse Prevention System's Dial-up user List)
  For more information go to http://maps.vix.com/rbl/

- Internet Mail Relay Services Survey (IMRSS)
  For more information go to http://www.imrss.org/

- DynamicIP Spam Sources List
  For more information go to http://www.imrss.org/dssl

MAPS-RBL is a system that creates intentional network outages so that the transport of unwanted mass email is prevented. ORBS is a database that lists SMTP servers that have been confirmed to permit third-party relay. MAPS-DUL, on the other hand, lists dial-up and other dynamically assigned IP addresses to prevent trespassing by people and/or organizations who send unsolicited email using direct connections to their victims' mail servers without using their ISP's mail server as a relay or gateway. IMRSS is a secure confidential list of open email relay servers that may be queried dynamically from various sites via DNS.

To enable RBL support, check the *Enable RBL lookup* option in the main Anti-spam configuration screen. Then click on the *RBL Database* button. The following screen will appear:

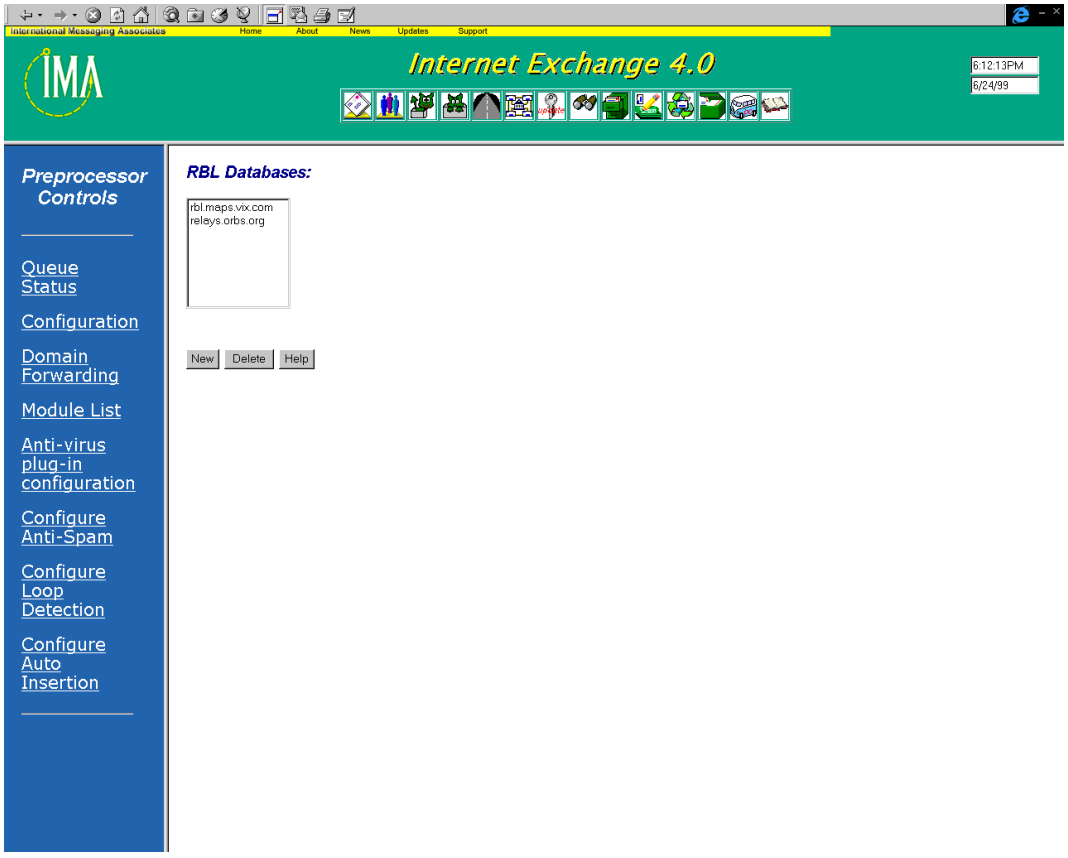

Figure 6hh - View RBL systems used by Internet Exchange 4

To add a new database to the list, click on the *New* button. A new screen will appear (see Figure 6ii).

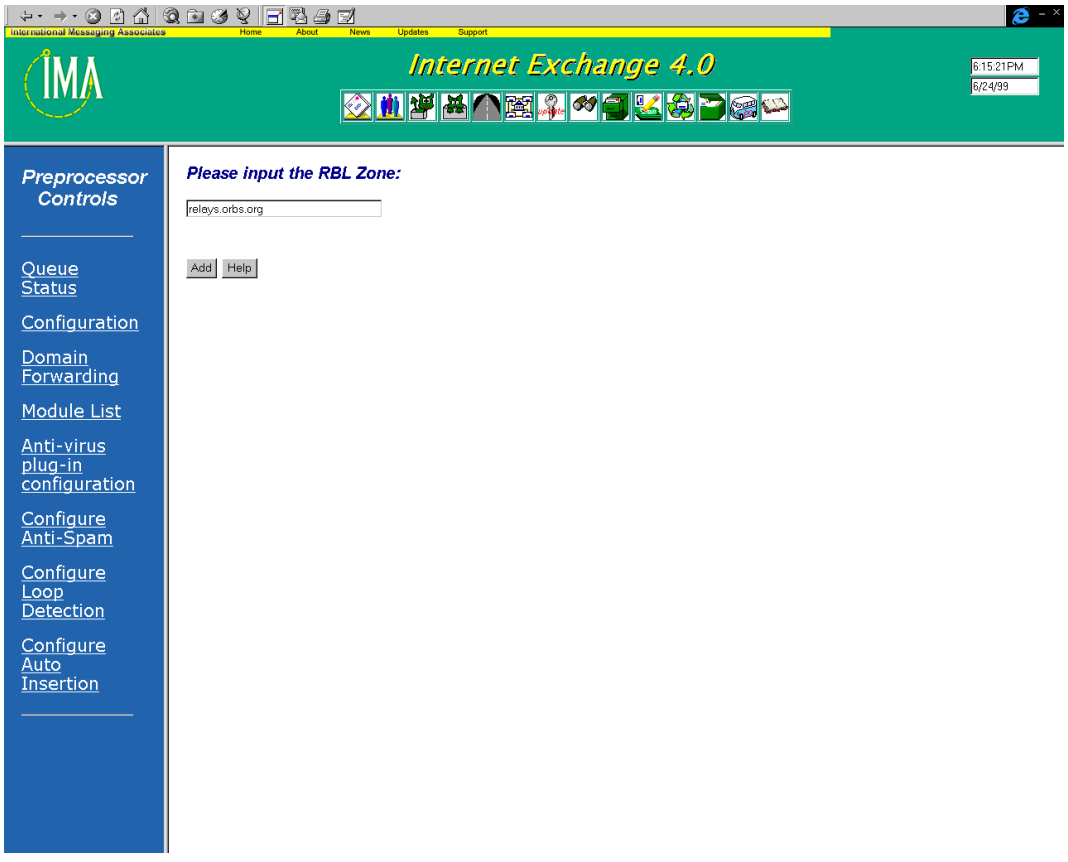

Figure 6ii - Add new RBL zones

Enter the new RBL zone to be added and click on the *Add* button. The new RBL zone will now be supported by the Anti-spam Module.

Loop Detection

Click on the Configure Loop Detection link on the main Preprocessor configuration screen. A new screen for specifying the following parameters will appear (see Figure 6jj):

Maximum trips

Specifies 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 is exceeded, the message is bounced. This option is useful in preventing message loops. The default value is 5.

Looping items to postmaster

If set, any looping messages are bounced to the local postmaster instead of being returned to the remote sender. This is often useful in preventing infinite email looping. The default is NO.

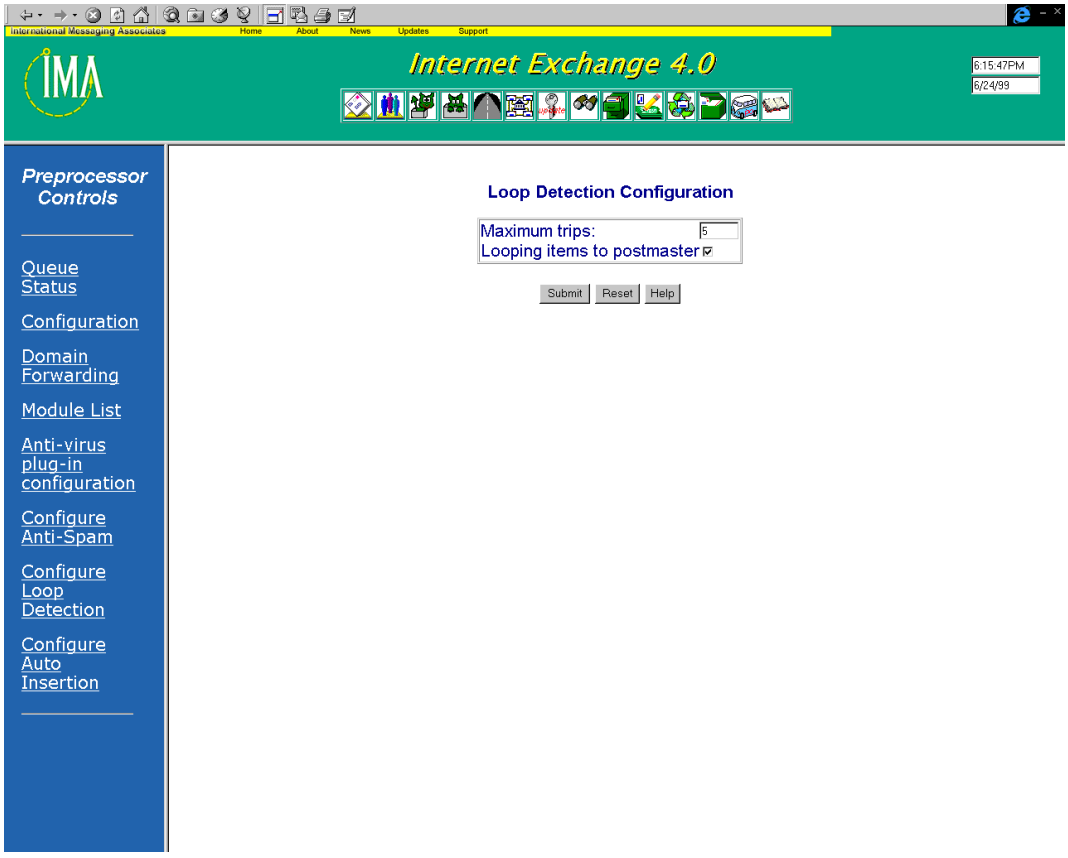

Figure 6jj - Configure Loop Detection

Click on the *Submit* button to implement the new settings.

Configuring Auto Insertion

The Auto Insertion utility is useful for adding disclaimers to outgoing messages. Using this feature, messages created by users will automatically include a disclaimer stating the confidentiality of the message and limiting the liability of the company that maintains the mail system where the message originated.

To configure the Auto Insertion engine, click on the *Configure Auto Insertion* link on the main Preprocessor configuration screen. The following screen will appear:

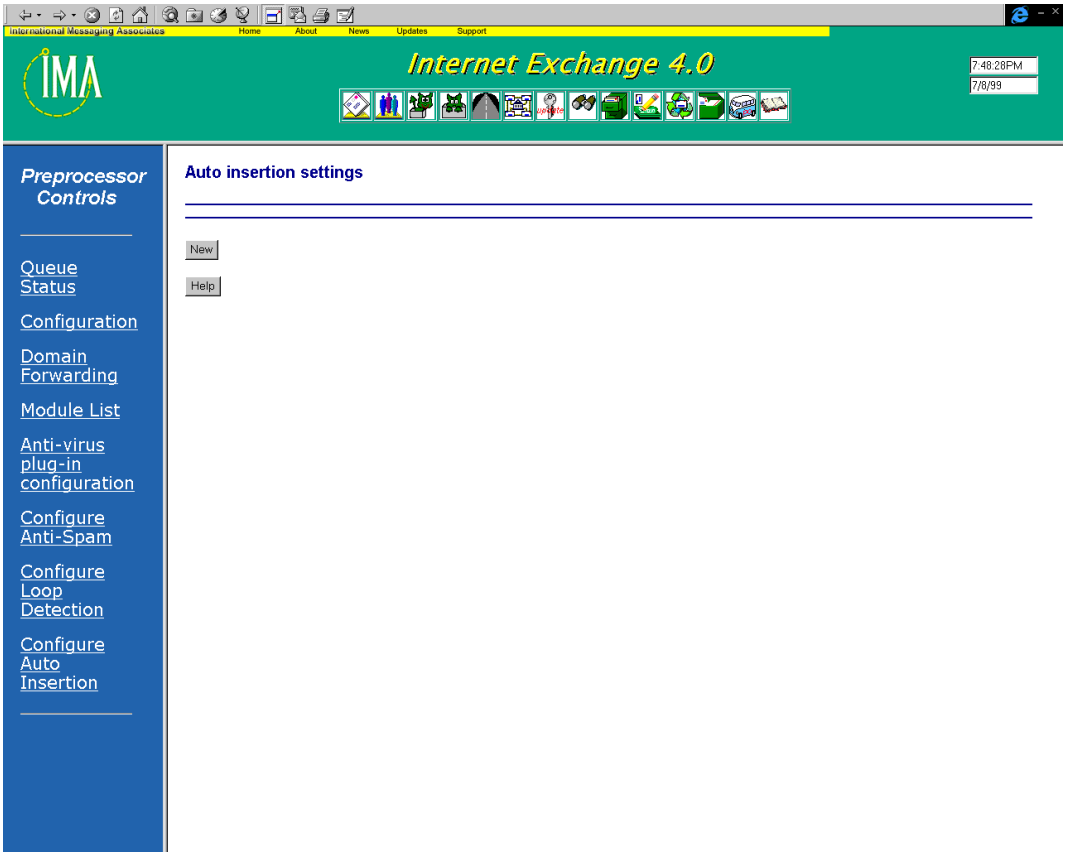

Figure 6kk - Auto insertion settings

Click on the New button to configure the Auto Insertion Engine's options. A new screen will appear (see Figure 6ll) for configuring the following parameters:

Source Channel

A disclaimer will automatically be attached by the Auto Insertion Engine to messages coming from this channel.

Text file

The path to the *.txt file that contains the disclaimer to be attached to outgoing messages.

HTML file

The system administrator is provided with the option to use an HTML file as a disclaimer for outgoing messages. This is the path to the HTML file.

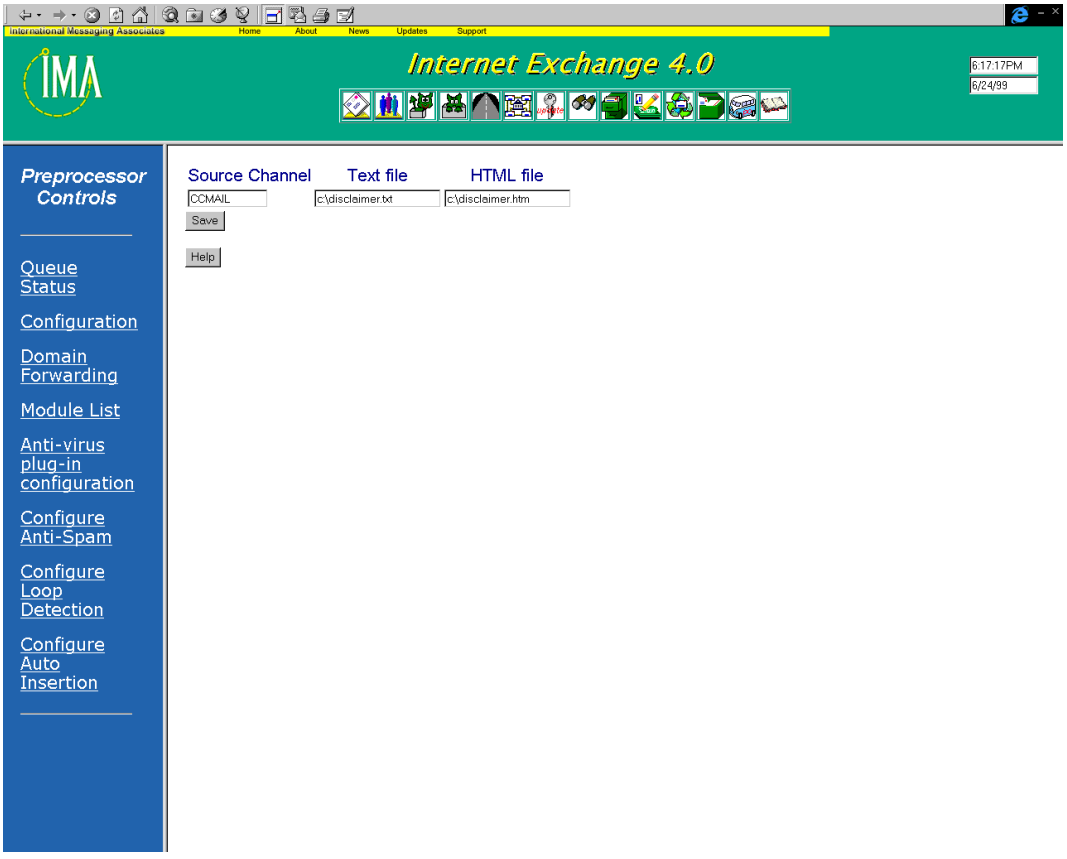

Figure 6ll - Configure Auto Insertion Engine

DISTRIBUTION LIST MANAGER

The Distribution List Manager allows messages to be sent to all of a list's subscribers simply by sending the said messages to a single address. The Distribution List Manager also enables the list maintainer to create Internet electronic mailing lists that support the following features: mail blocking, adding and removing subscribers, and setting the preferred delivery options. The Distribution List Manager provides the Web-interface necessary for creating and updating mailing lists. This assures the list maintainer the ease of maintenance.

The Distribution List Manager provides a web interface for the list maintainer (Figure 6mm). This interface enables the list maintainer to perform list operations. The list operations are:

- List of Lists
- Create New List
- Add Subscriber
- Remove Subscriber
- Delete Mailing List
- Display Members
- Modify Mailing List

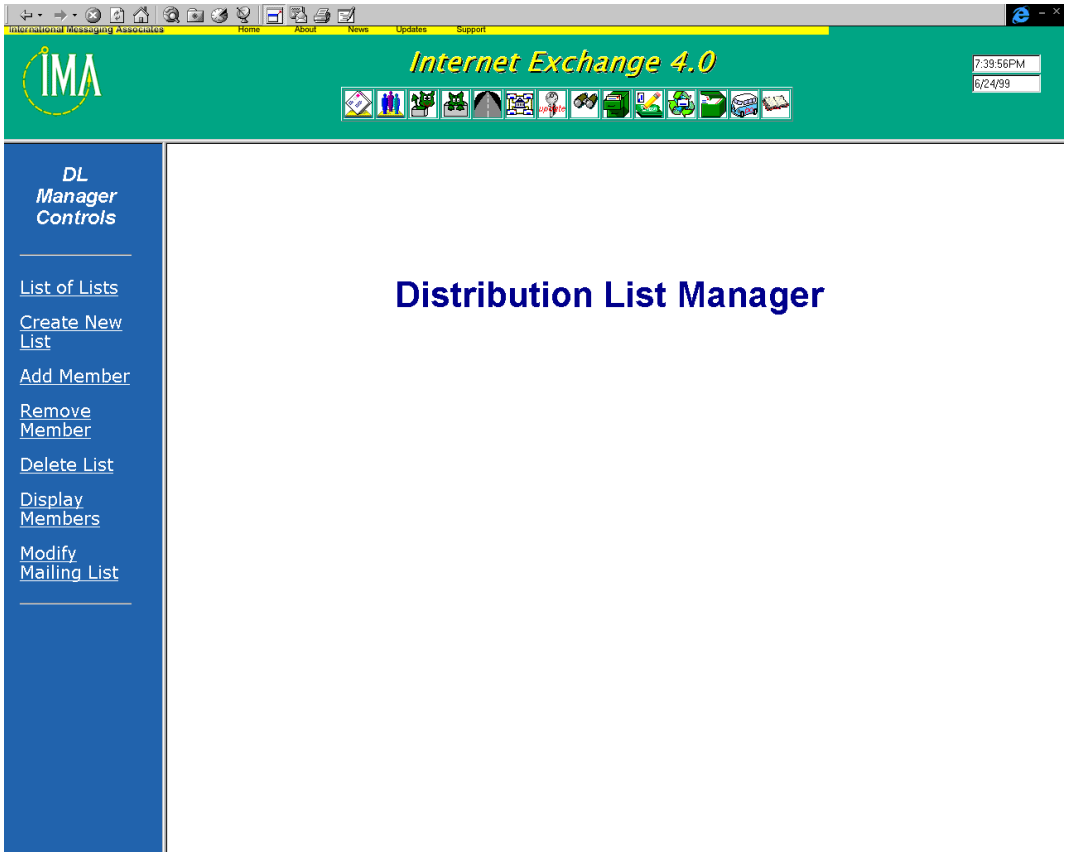

Figure 6mm - Main Distribution List Manager Configuration Page

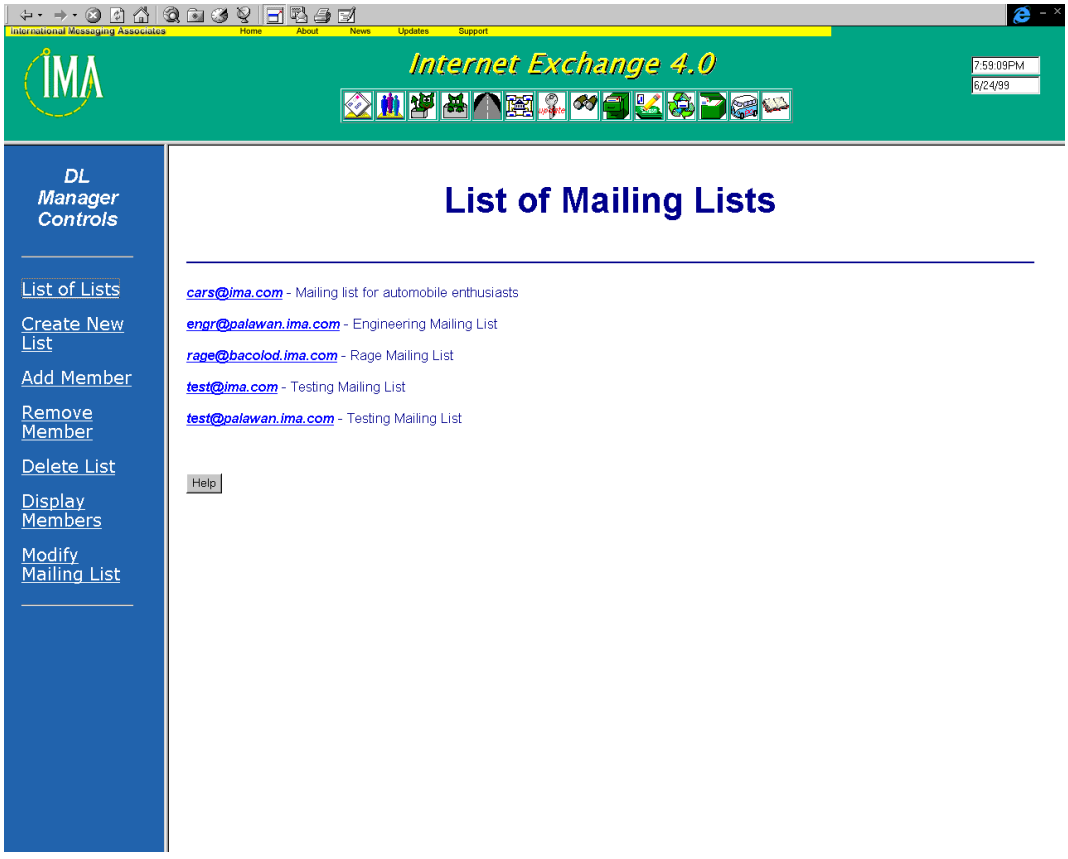

Figure 6nn - Display existing mailing lists

View all mailing lists

This option displays a list of all existing mailing lists that are serviced by the Distribution List Manager. Click the *List of Lists* link to go to the screen listing all registered mailing lists (see Figure 6nn).

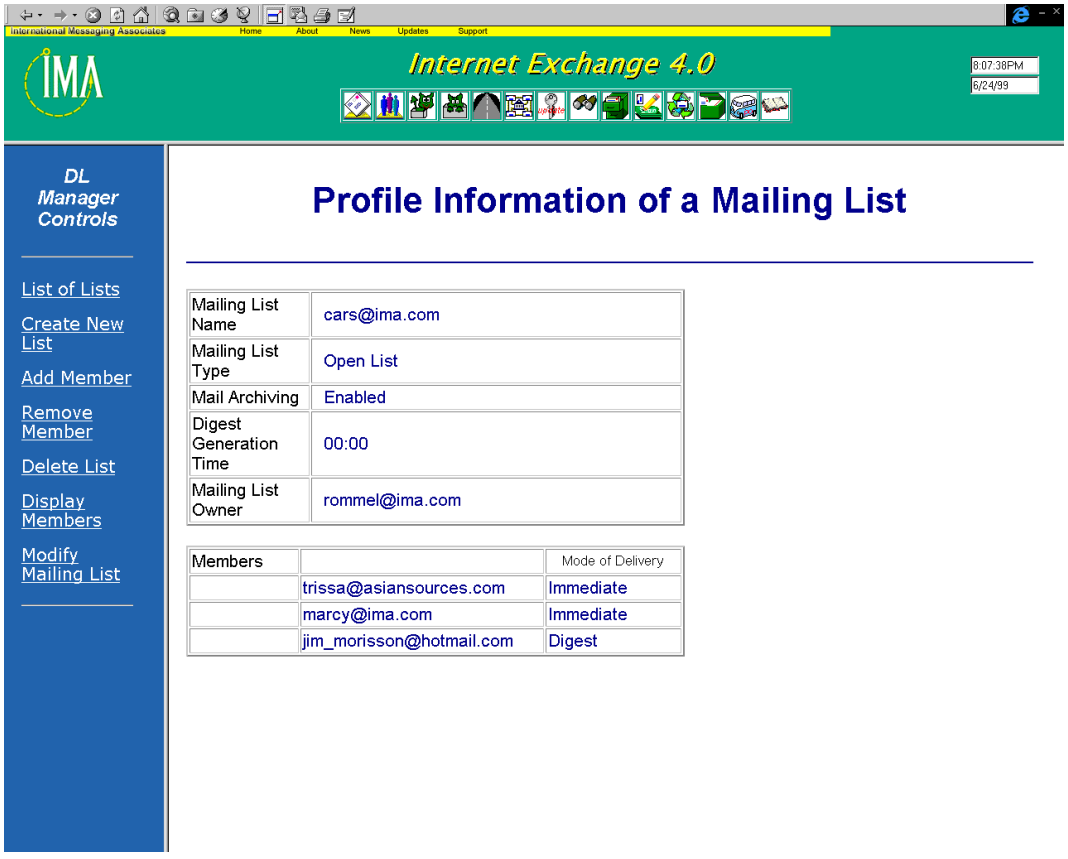

Figure 6oo - Mailing list profile

To view the profile of a mailing list, click on the mailing list name. In this example, we click on *cars@ima.com.* The profile of that particular mailing list is displayed in the next screen (see Figure 6oo).

The mailing list profile page displays the following information:

- List name
- List owner
- Type of list
- List members
- Mail archiving option
- Digest generation time

Create new mailing list

To create a new mailing list, click on the *Create New List* link. The following screen will appear:

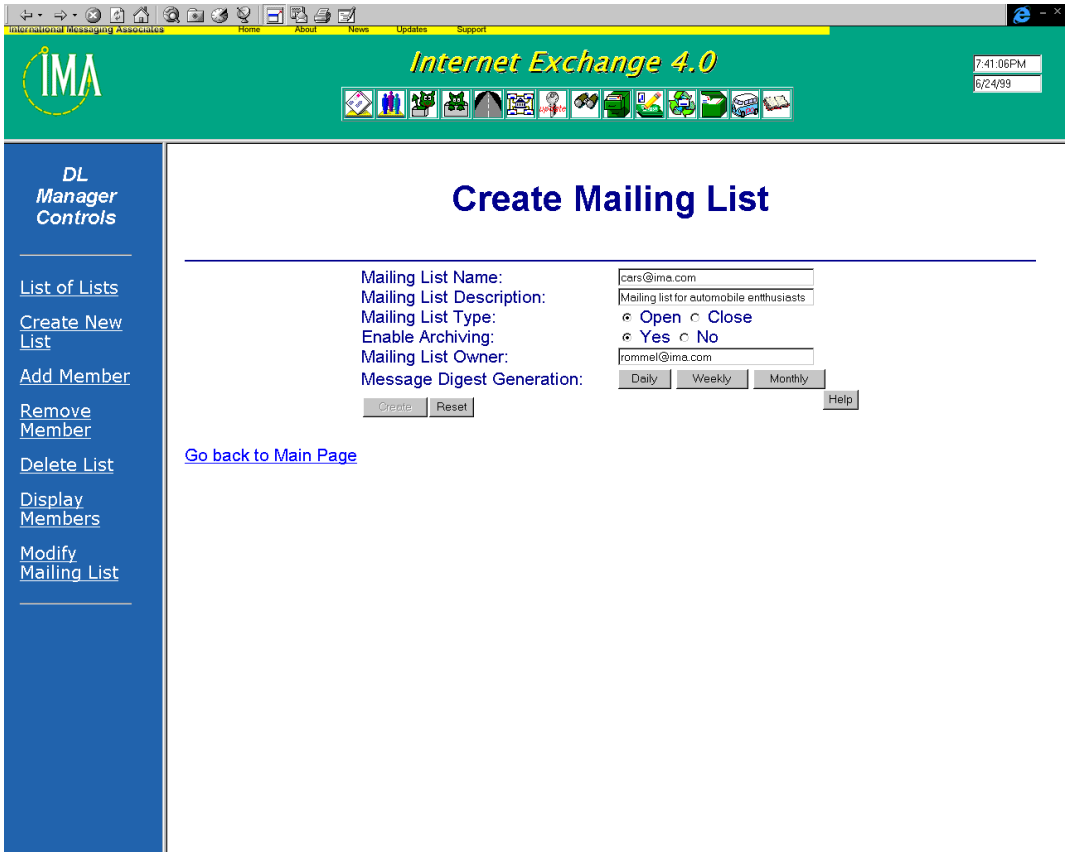

Figure 6pp - Create mailing list

Mailing List Name

The email address of the electronic mailing list to be created.

Mailing List Description

A brief description of the mailing list to be created.

Mailing List Type

Select either Open or Closed to define the type of mailing list.

Enable Archiving

Set the archiving option. Either enable or disable the archiving feature of the Distribution List Manager. When archiving is enabled, messages are saved in the archive folder under the DLMgr subdirectory.

Mailing List Owner

The email address of the person who will maintain/manage the mailing list that will be created.

Digest Generation Time

Digest Mode of Delivery works by accumulating messages in a digest. When the configured time comes, all subscribers using this mode of delivery are sent the digest message. The hour of delivery has a default value of 12 midnight. That is, every 12 midnight, a single message containing all the messages posted to the list since the previous digest will be sent. The value has a range of 00:00 to 23:59. The value 00:00 is equivalent to 12 MN and 23:59 is equivalent to 11:59 PM. There are three options for message digest generation: daily, weekly, and monthly.

There are three options for Message Digest Generation. These are: Daily, Weekly, and Monthly. Choosing the Daily option means the digest message will be generated on a daily basis at the time specified. Choosing the Weekly option means the digest message will be generated once a week at the day and time specified. Choosing the Monthly option means the digest message will be generated once a month at the day and time specified.

Choose any of the three options and click the appropriate button. The following screen will appear:

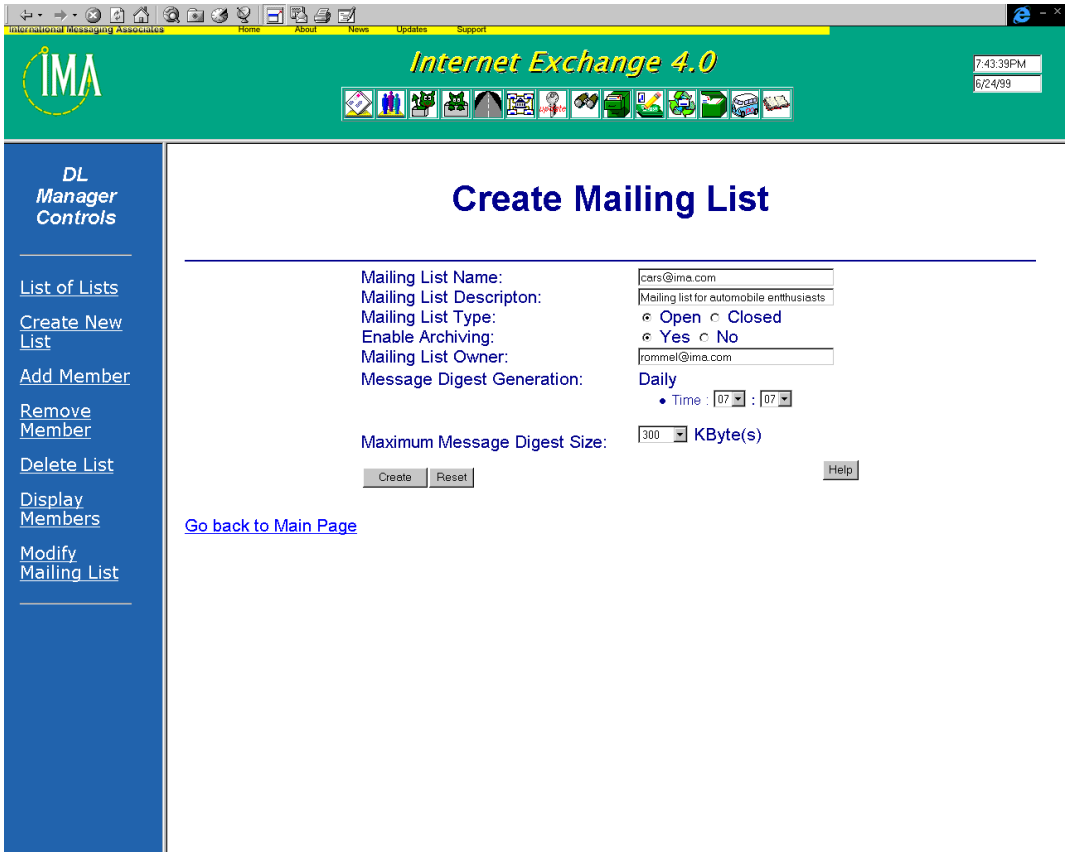

Figure 6qq - Specify digest generation time and maximum message digest size

In this screen (see Figure 6qq), the system administrator can specify the day/hour and minute when the message digest shall be generated. The maximum size of the message digest is also specified in this screen. If the message digest exceeds the limit, it will be divided into several smaller messages.

The next screen (see Figure 6rr) displays the paths of the subscription text file (for automatic subscription), unsubscription text file (for automatic unsubscription), welcome text file (for new members), disclaimer text file (for attaching disclaimer to outgoing messages).

Click on the *Create* button to add the new mailing list to the Directory server. A new screen displaying the attributes of the new mailing list will appear (see Figure 6ss).

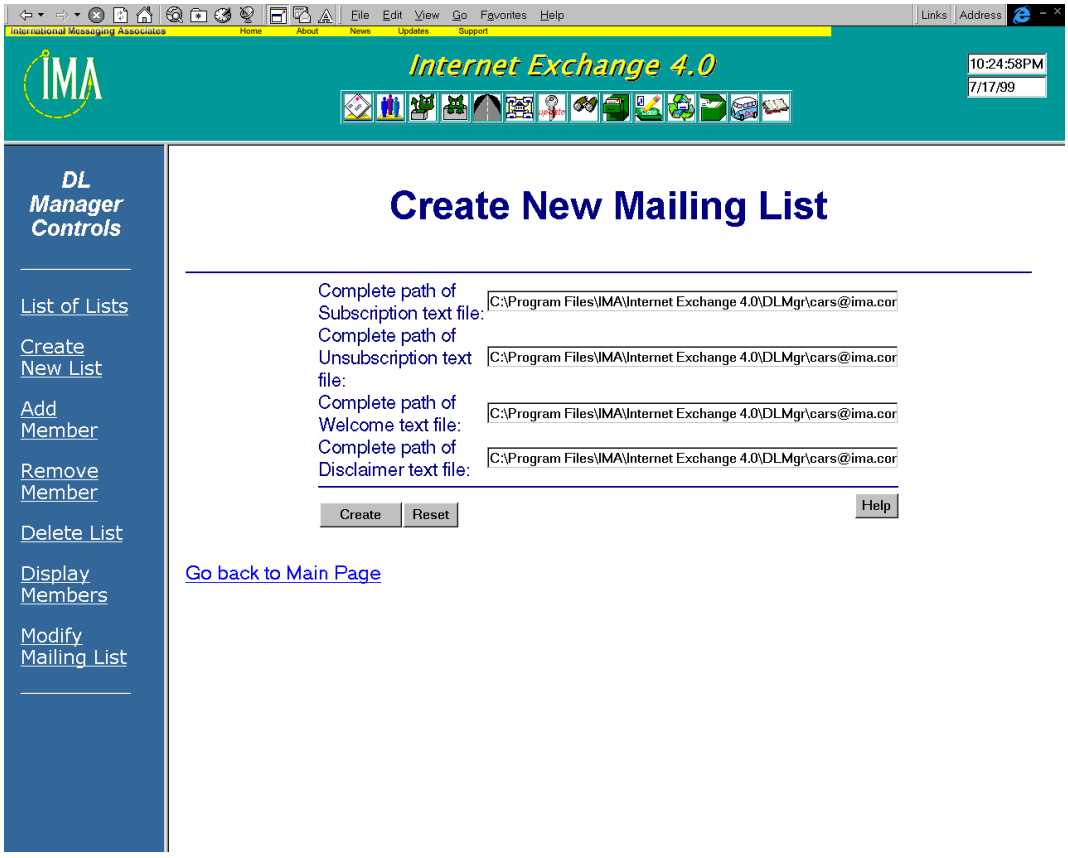

Figure 6rr - Display paths of text files used by the DL Manager

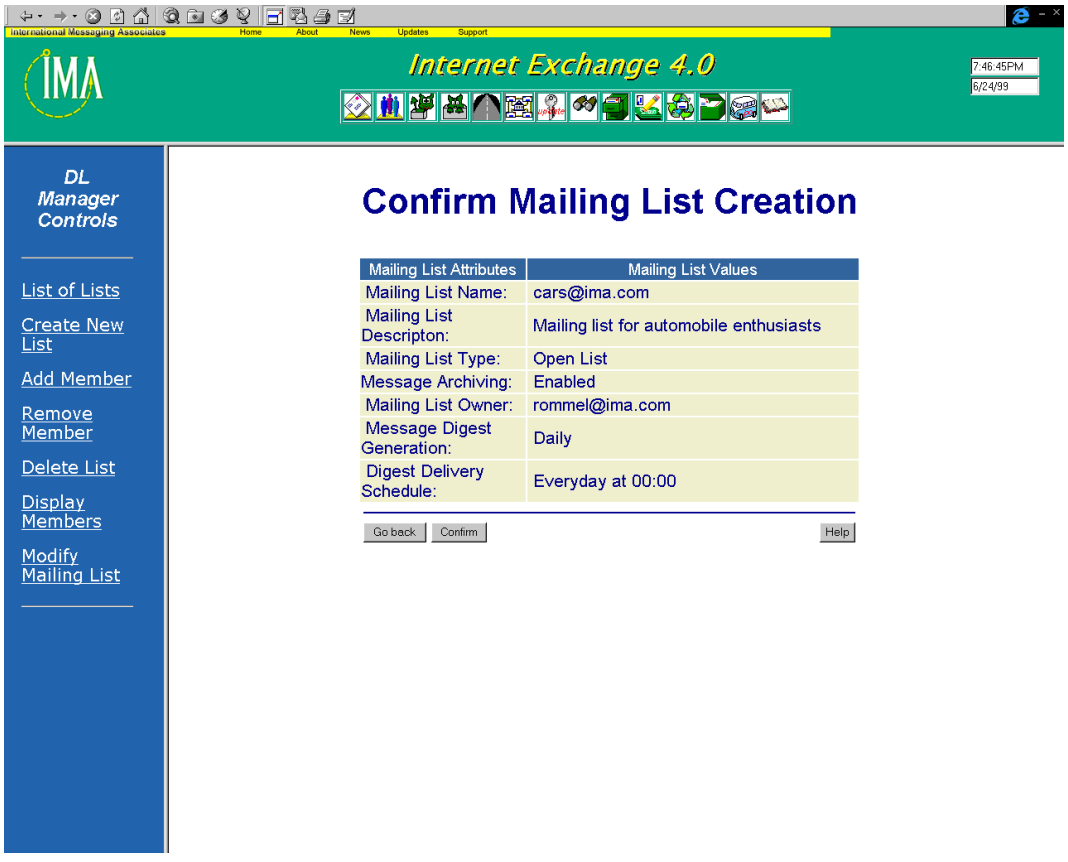

Figure 6ss - Confirm creation of mailing list

Add new mailing list member

To add a new subscriber to an existing mailing list, click on the *Add Member* link on the main DL Manager configuration screen. The following screen will appear:

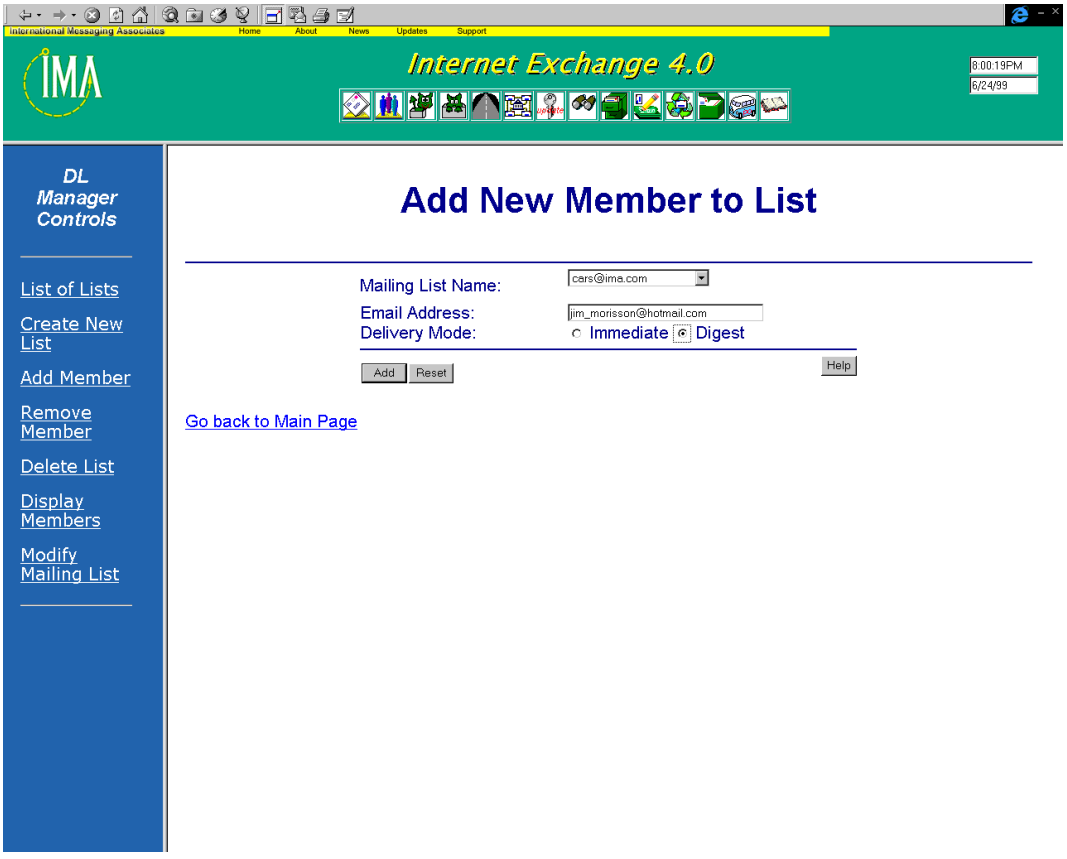

Figure 6tt - Add new list member

Mailing List Name

Select the mailing list to which the new member will be added from the pull-down menu.

Email Address

The email address of the subscriber to be added.

Delivery Mode

The mode of delivery for the new member. This can either be *Immediate* or *Digest*.

After specifying all the parameters for the new subscriber, click on the *Add* button. A new screen will appear displaying the attributes of the mailing list member (see Figure 6uu). Click on the *Confirm* button to add the new member to the list.

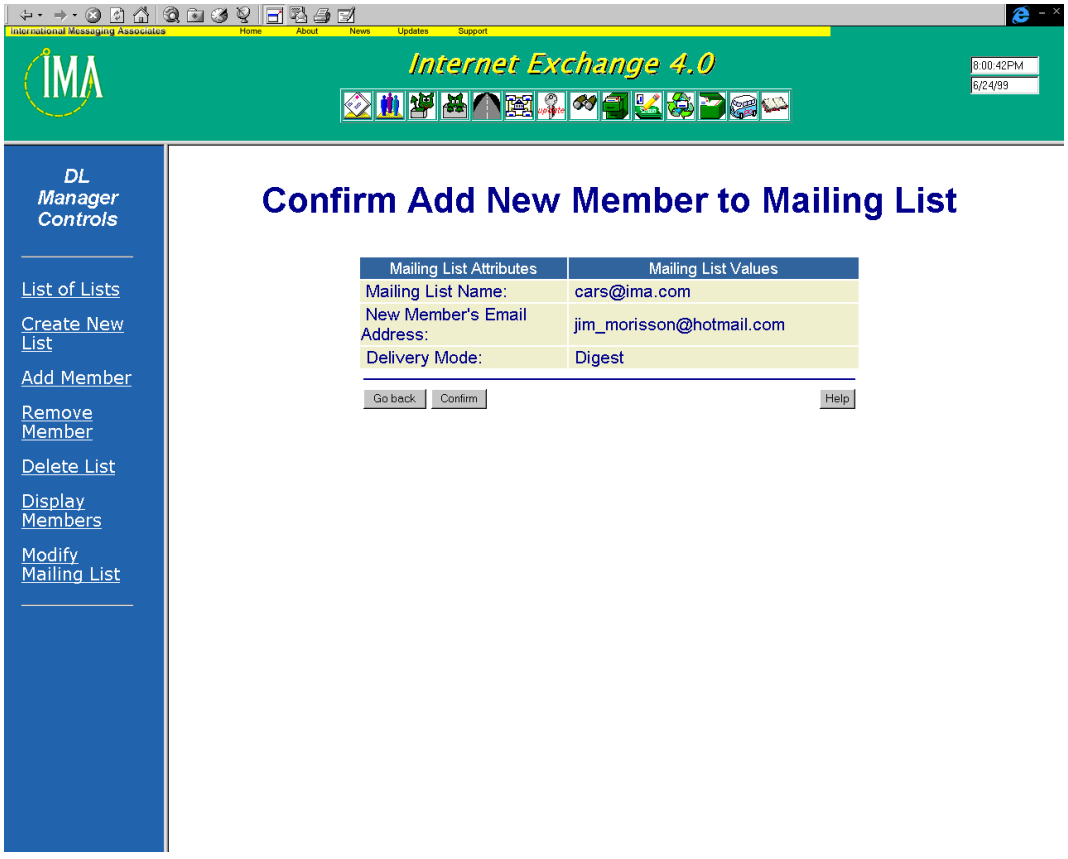

Figure 6uu - Confirm addition of new member

Remove mailing list member

To remove a subscriber from an existing mailing list, click on the *Remove Member* link on the main DL Manager configuration screen. The following screen will appear:

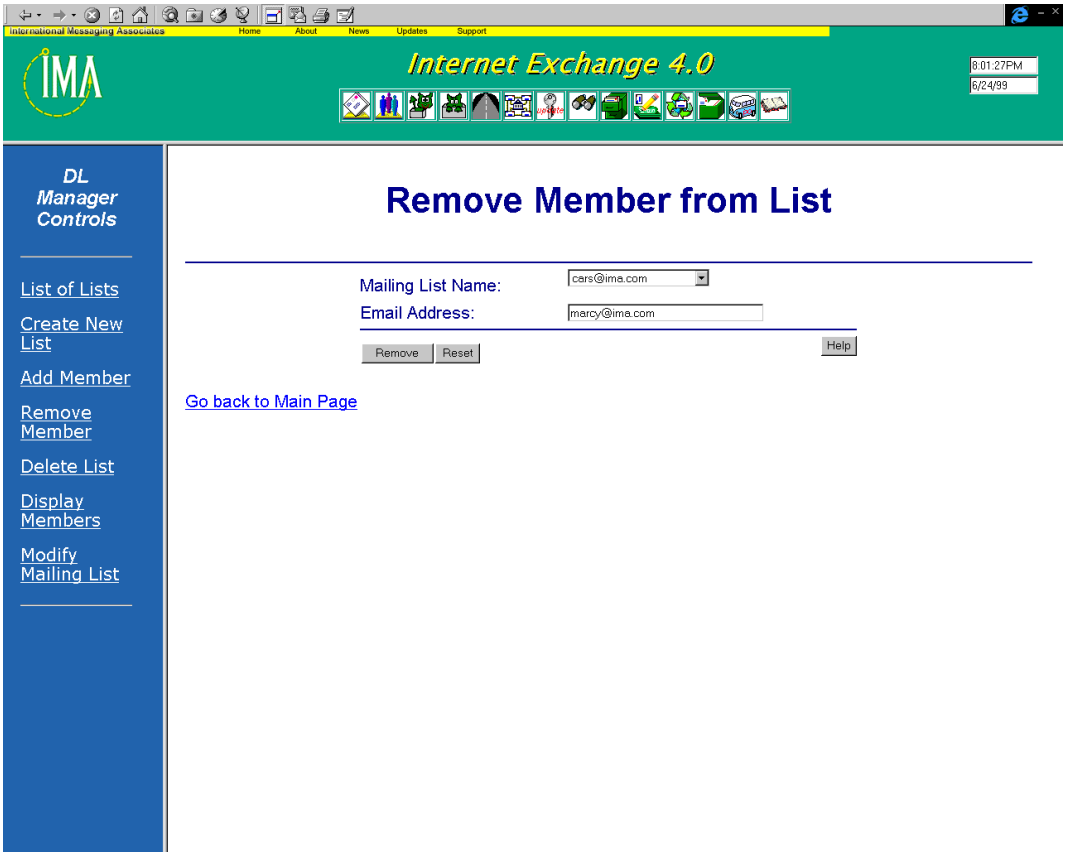

Figure 6vv - Remove mailing list member

Mailing List Name

The name of the mailing list of which the member to be removed is a current member.

Email Address

The email address of the member to be removed from the mailing list.

Click on the *Submit* button to remove the email address of that particular member from the selected mailing list. The user will be asked to confirm the removal of member to the list before the member is permanently removed from the system (see Figure 6ww).

Click on the *Reset* button clears all the text boxes.

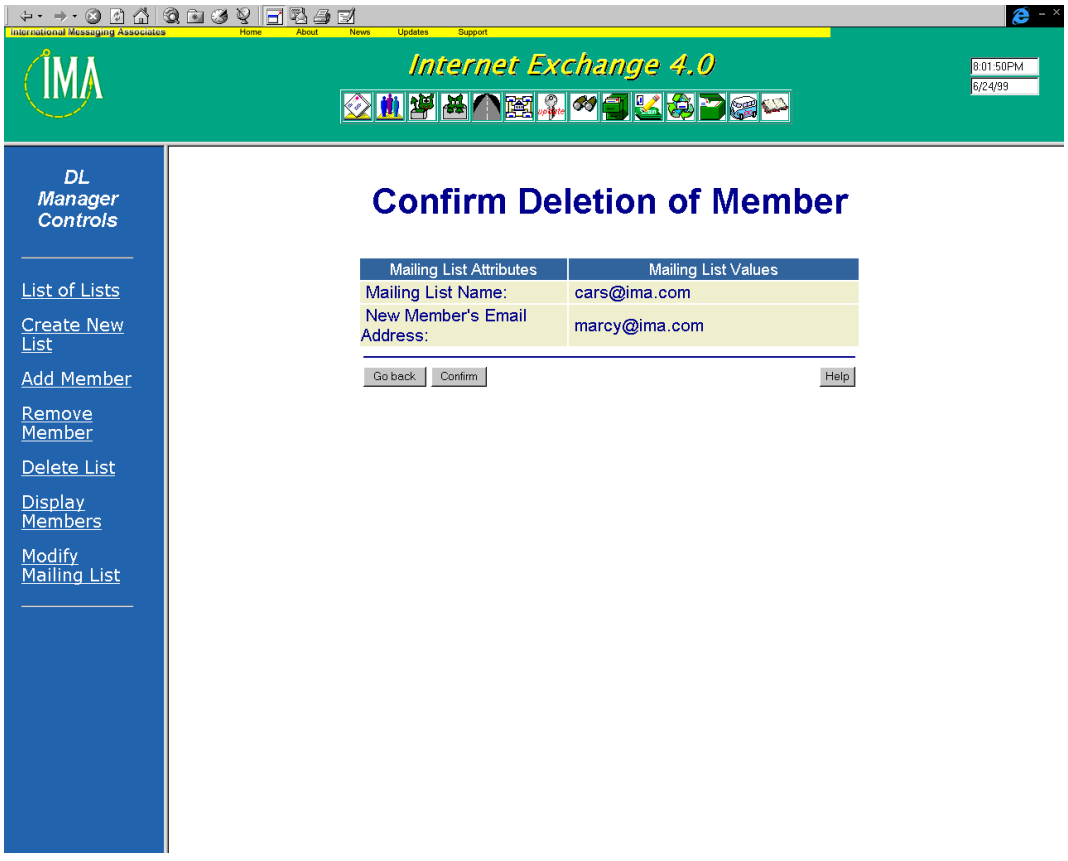

Figure 6ww - Confirm removal of list member

Delete mailing list

To remove a mailing list and its subscribers, click on the *Delete List* link on the main DL Manager configuration screen. The following screen will appear:

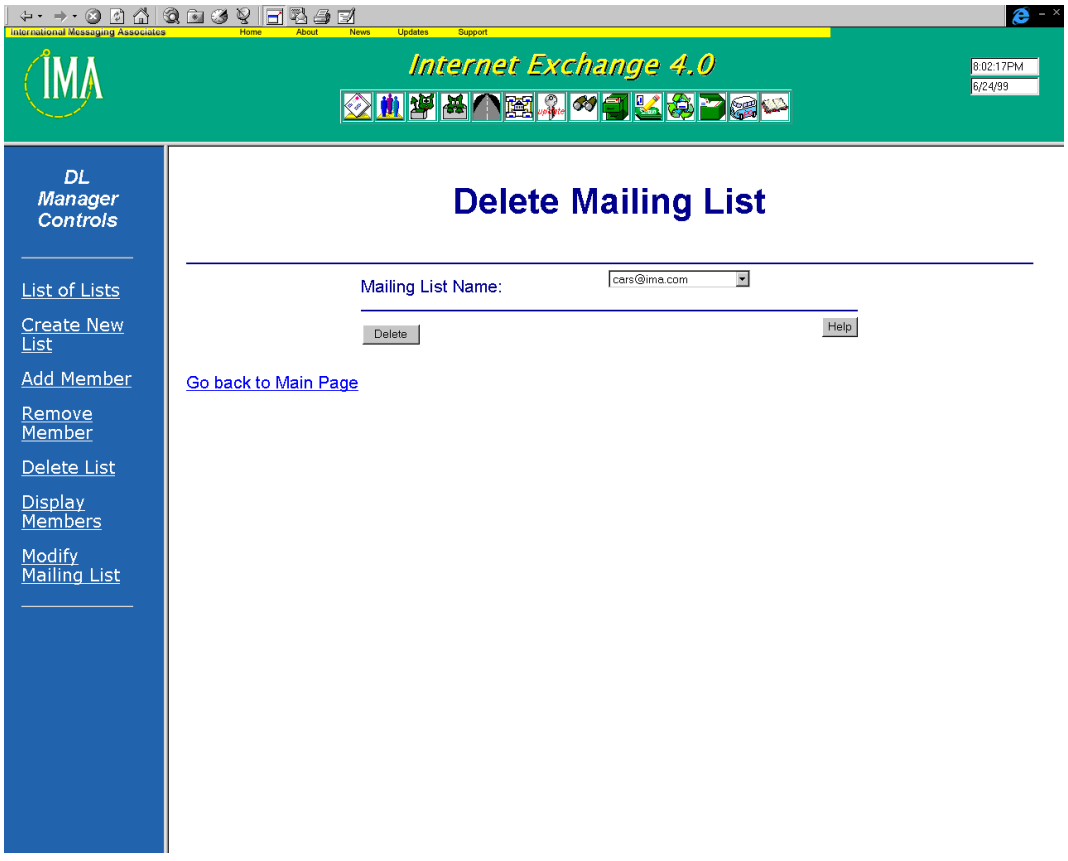

Figure 6xx - Remove mailing list

Mailing List Name

Select the name of the mailing list to be deleted from the pull-down menu.

Click on the *Submi*t button to remove the mailing list and all its members. The user will be asked to confirm the deletion of the mailing list before the list is permanently removed from the system (Figure 6yy).

The *Reset* button clears all the text boxes.

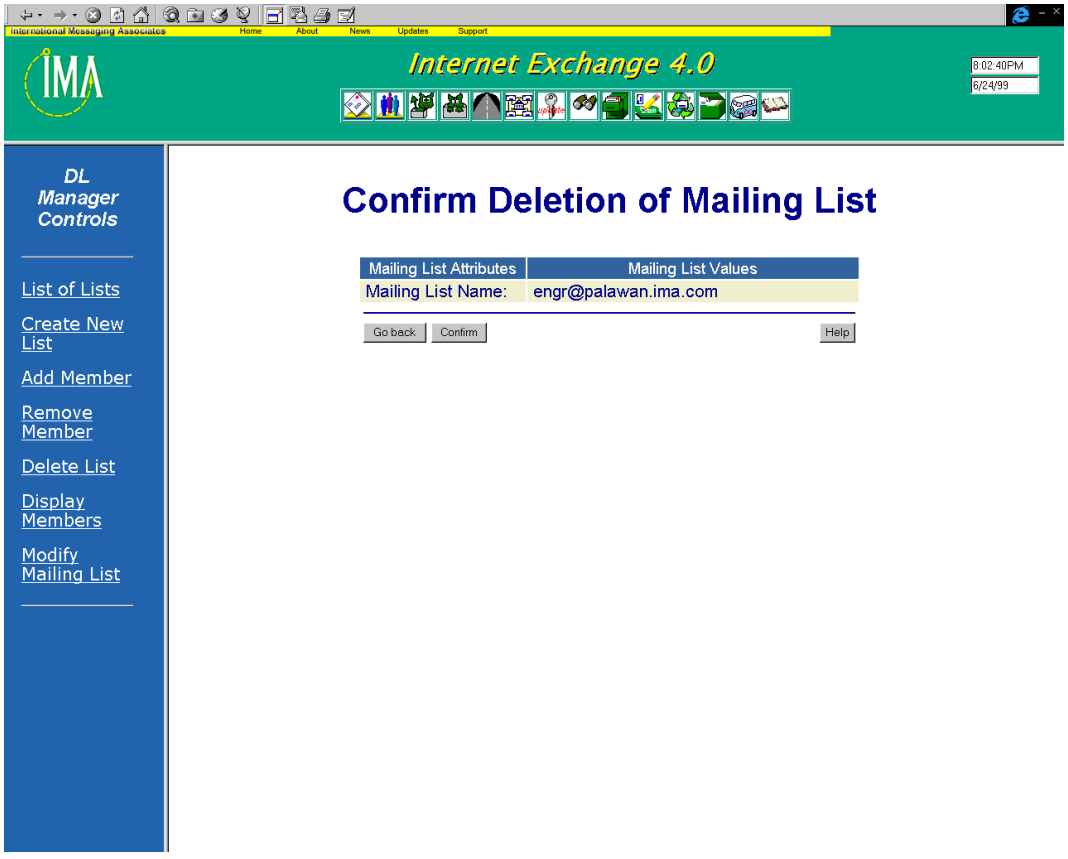

Figure 6yy - Confirm removal of mailing list

Display mailing list members

To display all registered subscribers of an existing mailing list, click on the *Display Members* link on the DL Manager configuration screen. The following screen will appear:

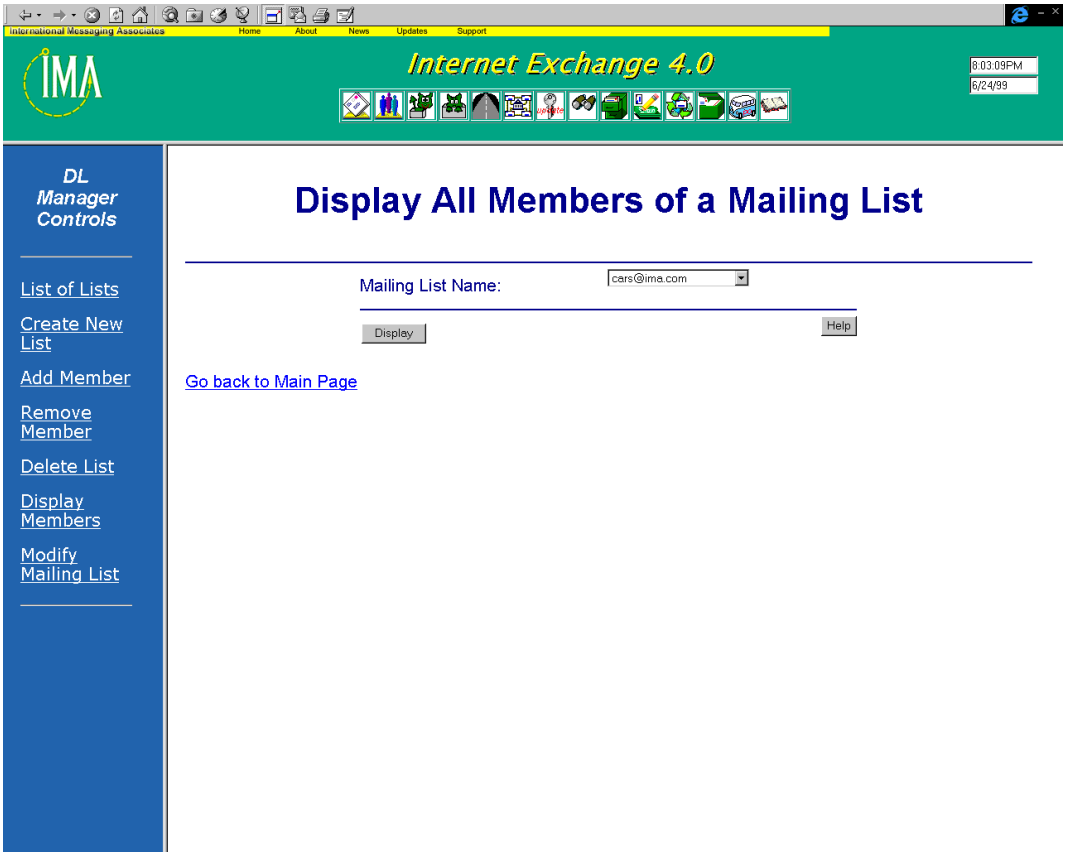

Figure 6zz - Display mailing list members

Mailing List Name

Select the mailing list name from the pull-down menu.

Click on the *Display* button to view all subscribers to that particular list (see Figure 6a-1). Click on the *Reset* button clears all the text boxes.

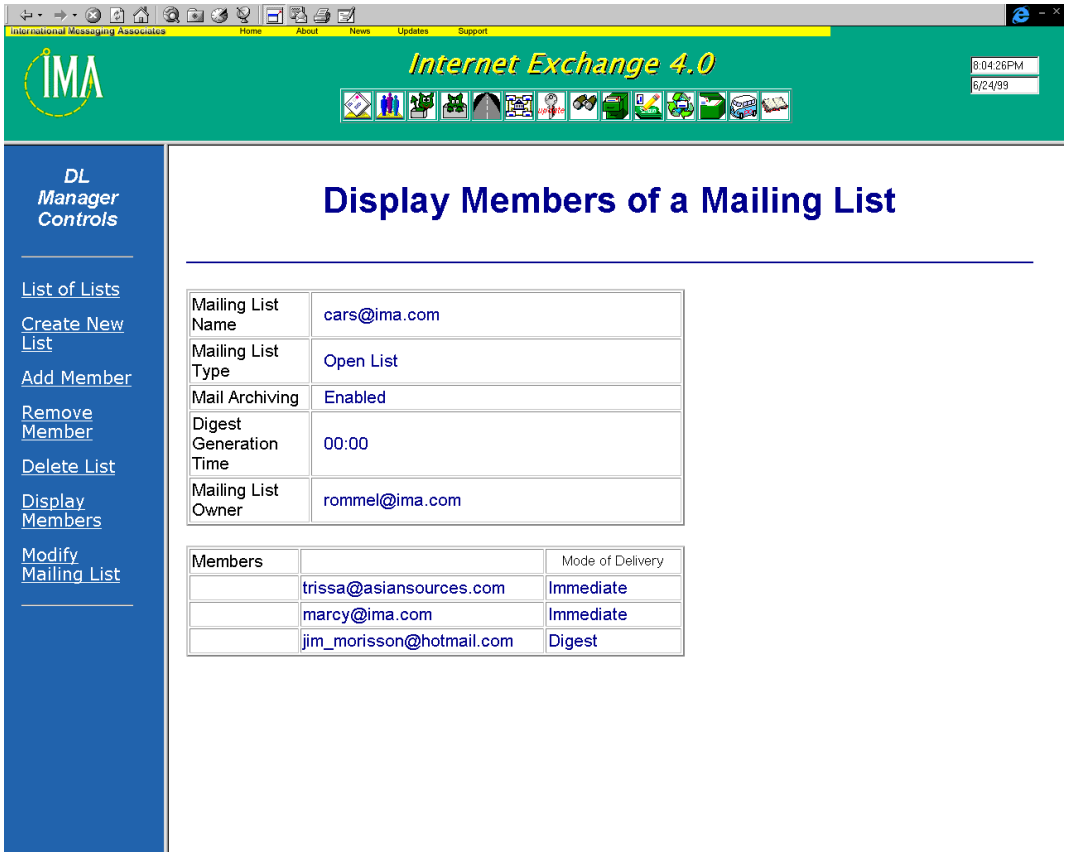

Figure 6a-1 - Display mailing list attributes and members

The screen above displays the following information in tabular form:

- Mailing List Name
- Mailing List Type
- Mail Archiving
- Digest Generation Time
- Mailing List Owner
- Mailing List Members
- Mode of delivery for each member

Modify mailing list

To modify the settings of an existing mailing list, click on the *Modify Mailing List* link on the DL Manager configuration screen. The following screen will appear:

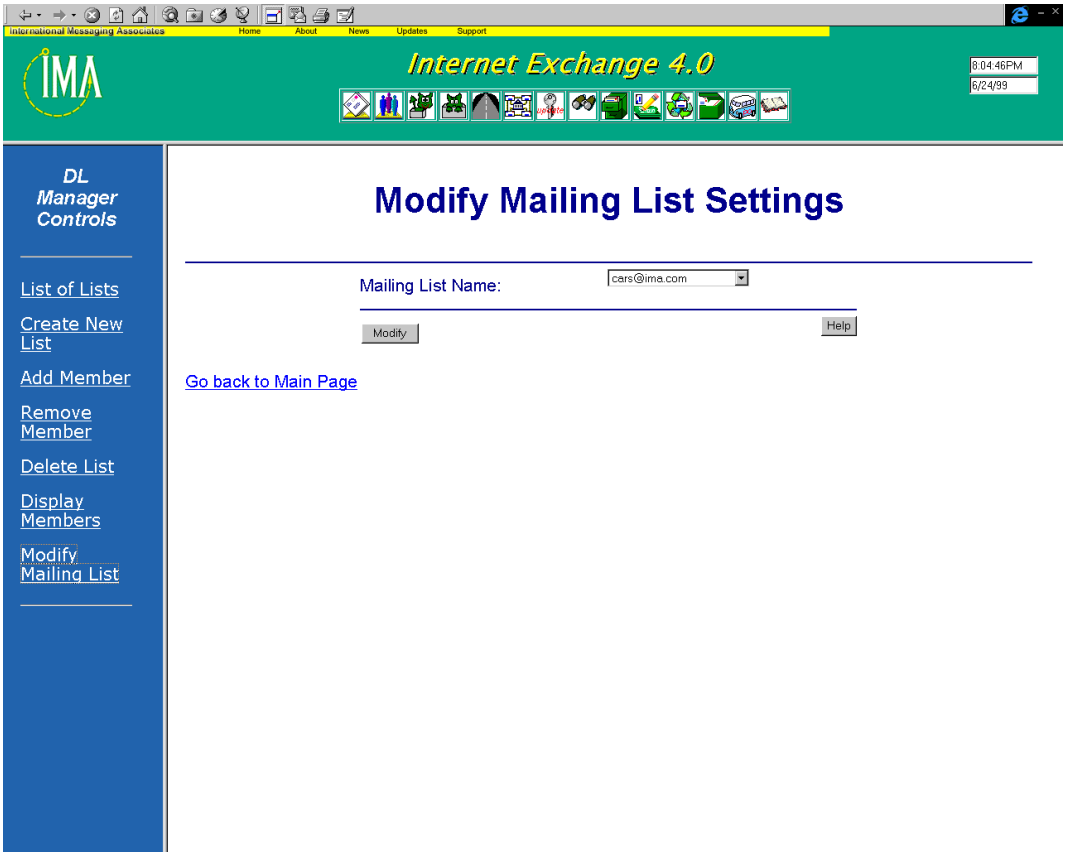

Figure 6a-2 - Modify mailing list settings

To make modifications to an existing mailing list, select the mailing list name from the pull-down menu and click on the *Modify* button. A new screen displaying the mailing list's attributes will be displayed (see Figure 6a-3)

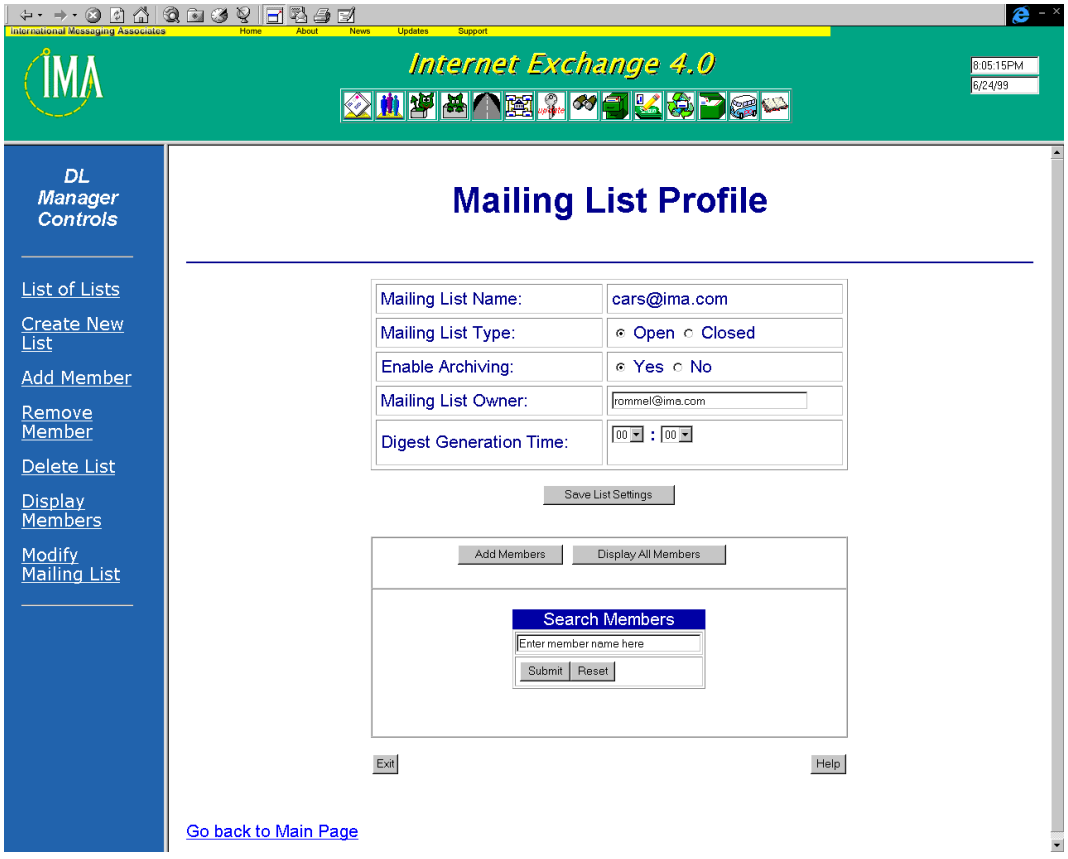

Figure 6a-3 - Display mailing list settings

After making the necessary changes, click on the *Save List Settings* button to save the new settings.

To add a new member to the list, click on the *Add Member* button. The screen for adding new members to a mailing list will be displayed (see Figure 6tt).

To view the complete list of members, click the *Display All Members* button. This will generate a list of all the members of that particular list (see Figure 6a-4).

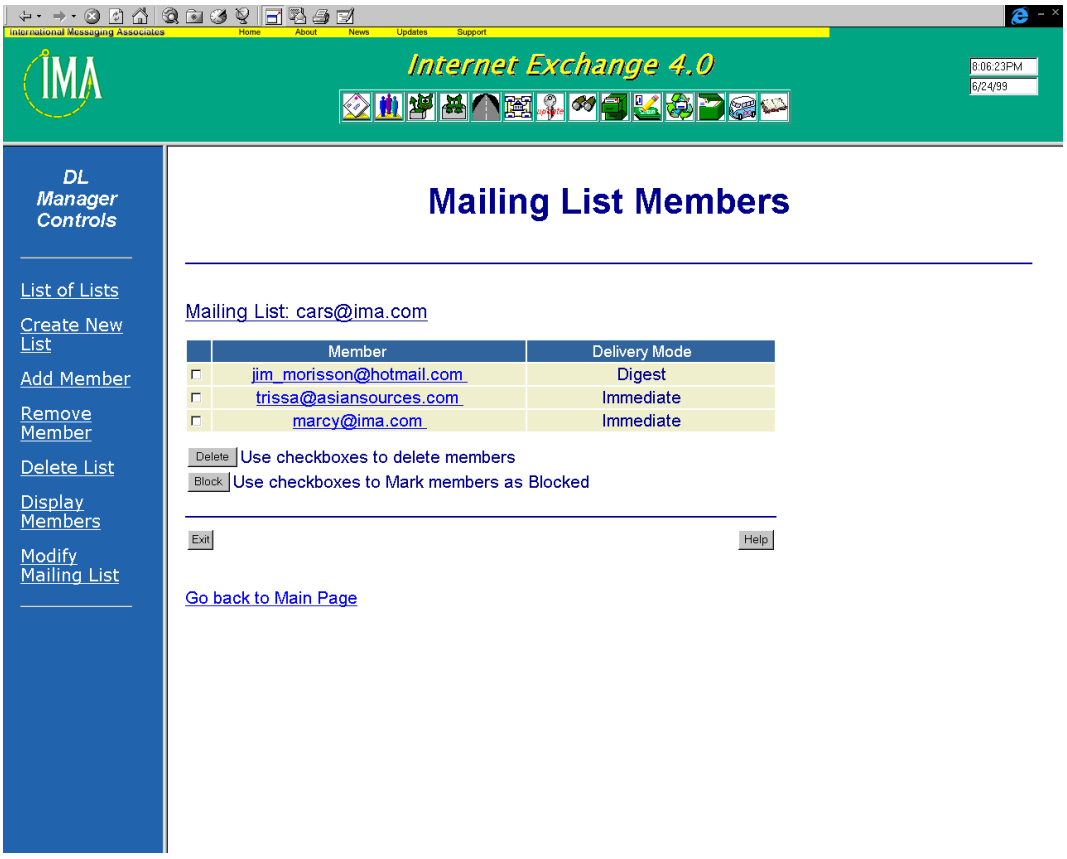

Figure 6a-4 - Display all mailing list members

Each member displayed has a hyperlink. To modify the current delivery mode of a particular member, click the member's email address on the screen. The screen for changing the delivery mode for that particular member will appear (see Figure 6a-5). After changing the mode of delivery click on the *Update* button to save the new setting for that member.

In the screen above, the system administrator has the option to delete members from the list. To do this, check the appropriate boxes and click on the *Delete* button.

To prevent a member from posting messages to the list, select the appropriate checkbox and click on the *Block* button. The said member will be able to receive posted messages but will be unable to post messages to the list.

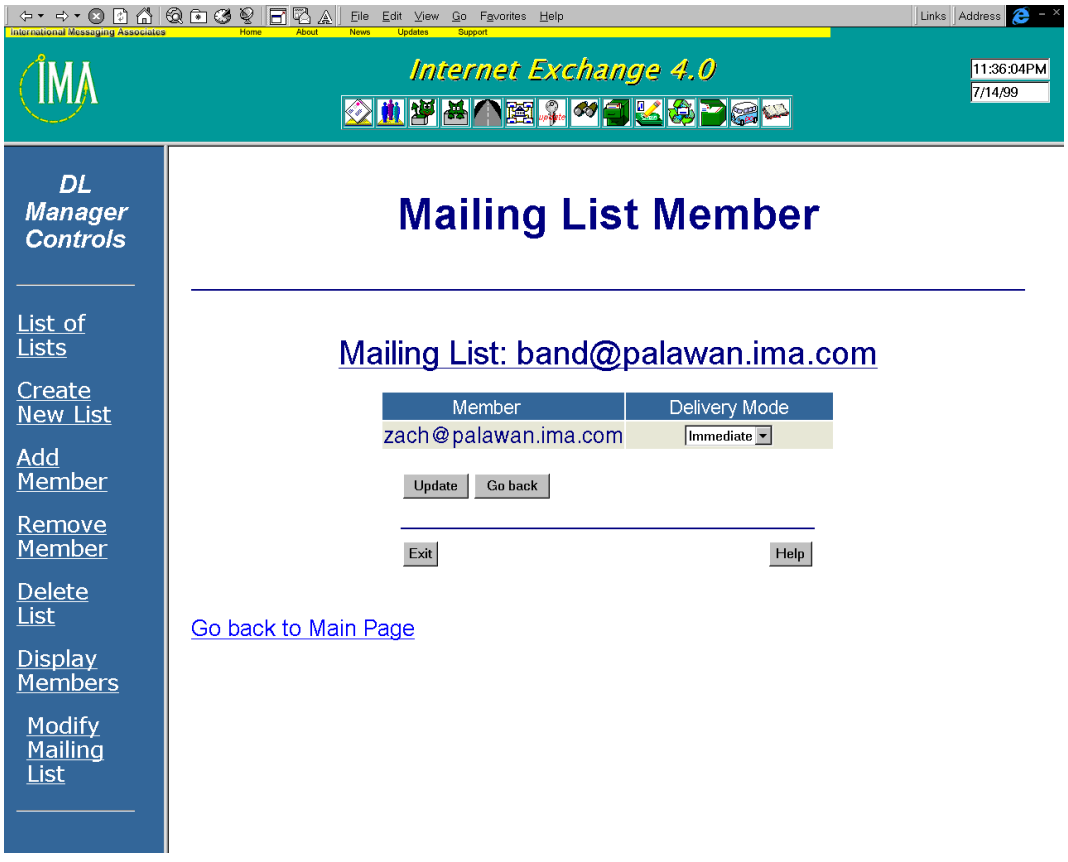

Figure 6a-5 - Change mail delivery mode

DIRECTORY SERVER

Internet Exchange 4's Directory Server is based on the open Internet directory standard, the Lightweight Directory Access Protocol (LDAP). The LDAP is a protocol designed to provide read/write access to open X.500 directory service and proprietary directories that support the X.500 standard without incurring the resource requirements of its predecessor, the DAP or Directory Access Protocol. Unlike the DAP, the LDAP does not require the upper layers of the OSI protocol stack and runs directly on TCP/IP or other reliable transport protocol.

To configure the various features of the LDAP-based Directory Server, click on the LDAP icon on the main Web Administration Interface. The following screen will appear:

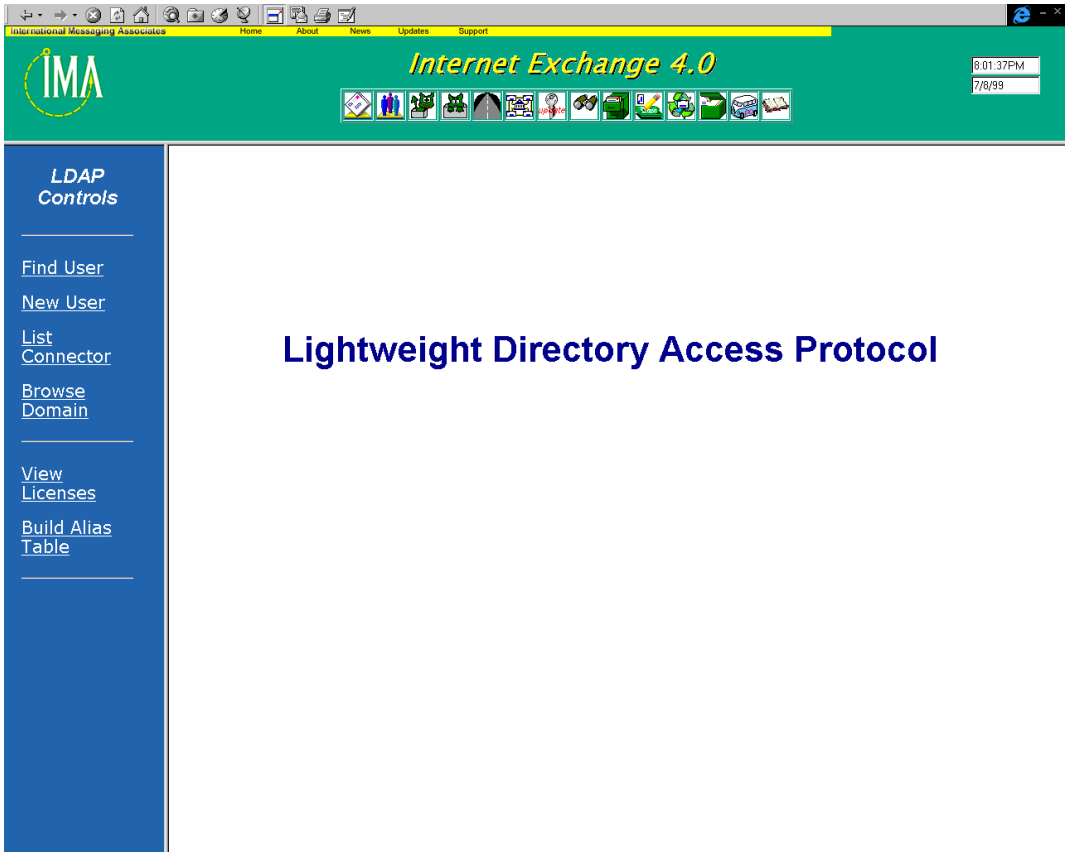

Figure 6a-6 - Main LDAP Administration Interface

The Internet Exchange Directory supports the following functions:

- Find users
- Add new users
- List connectors
- Browse domains
- View licenses
- Build alias tables

Find users

To search for users registered in the Directory Server, click on the *Find User* link on the main LDAP configuration screen. The following screen will be displayed:

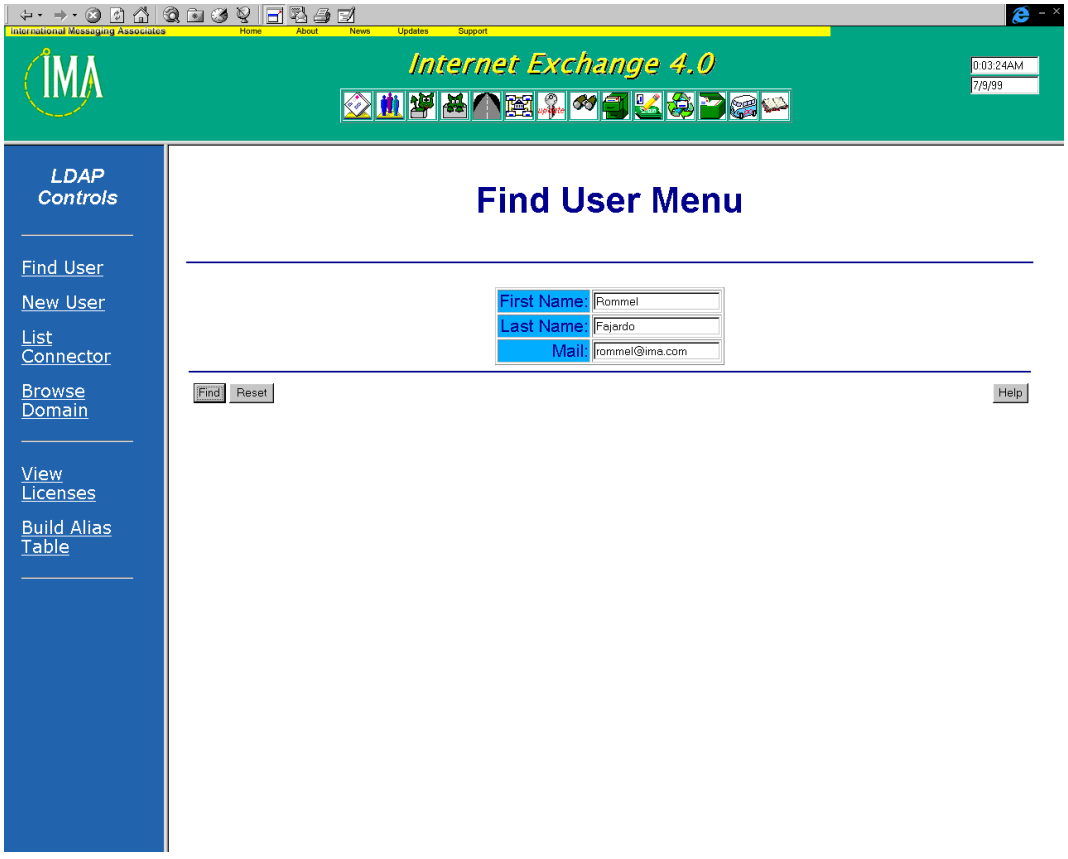

Figure 6a-7 - Enter search parameters

First Name

The first name of the user. Use of wildcards (an asterisk *) in this field is allowed.

Last Name

The last name of the user. Use of wildcards (an asterisk *) in this field is allowed.

Mail

The email address of user. Use of wildcards (an asterisk *) in this field is allowed. Make sure that the email address has a domain part. This component is used in searching for LDAP entries. The mail address can either be of the format *username@host.domain*, like *rommel@ima.com*, or any combinations of wild characters and letters. Valid entries are:

- r*@ima.com
- *.ima.com
- *ima.com

After entering all the parameters required, click on the *Find* button. If the Directory Server finds a user whose attributes match those entered by the system administrator, a new screen will appear:

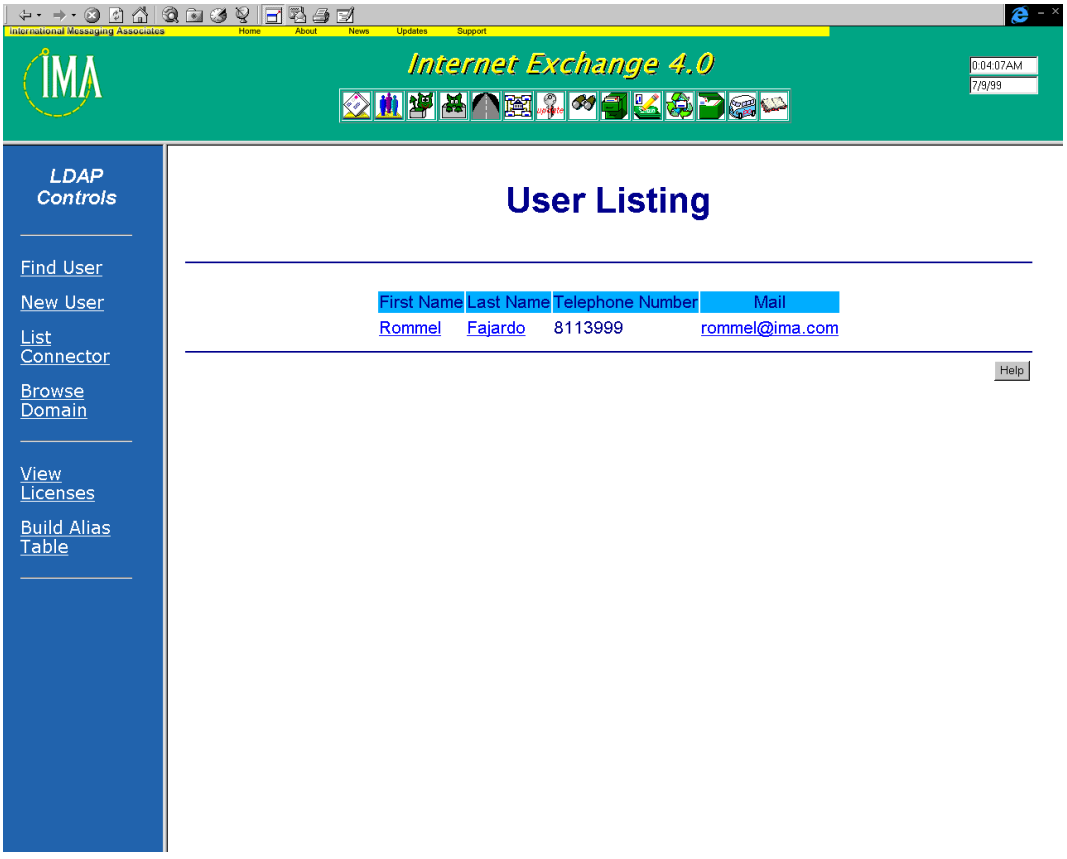

Figure 6a-8 - Search results

To view the attributes of the user, click on the user name or email address. A new screen will be displayed (see Figure 6a-9).

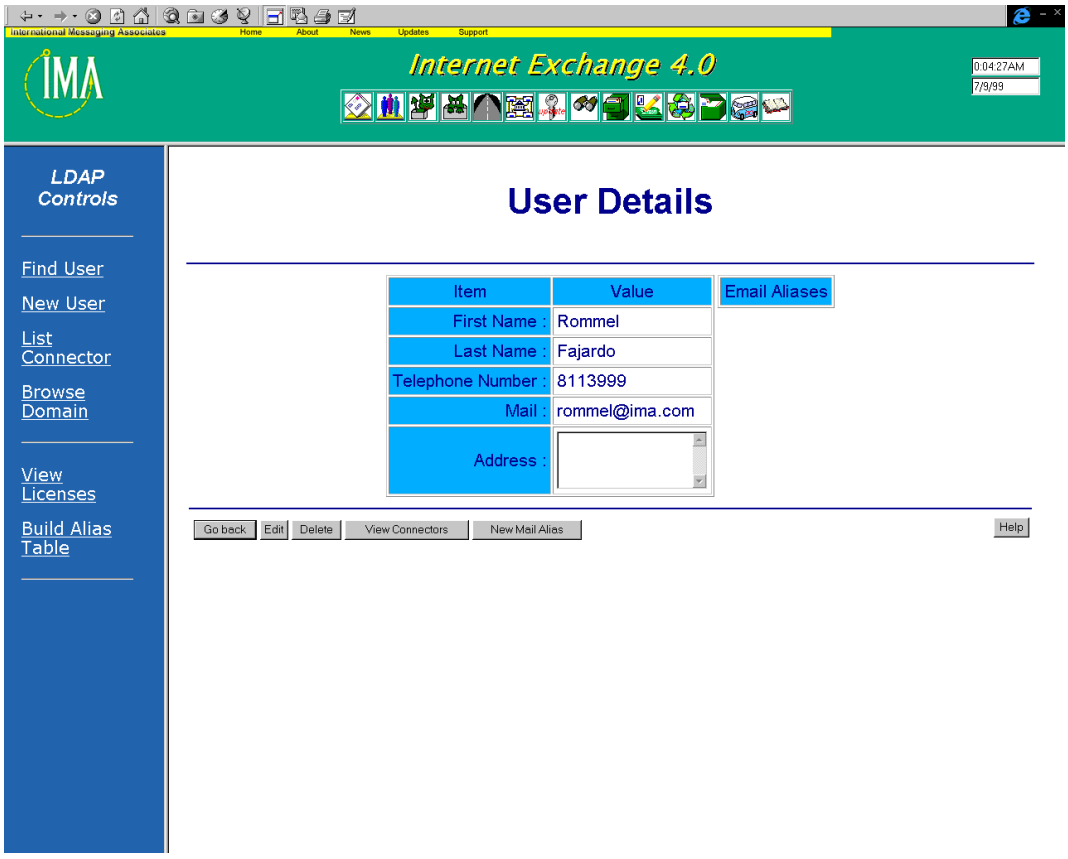

Figure 6a-9 - User attributes

This window displays the attributes of the selected user including:

- First Name
- Last Name
- Telephone Number
- Address
- Email address

Click on the *Edit* button to modify the current entry. To remove the user from the LDAP database, click on the *Delete* button. Click on the *Reset* button to ignore all modifications. The original values will then be displayed.

View connectors

Internet Exchange 4 features a number of connectors. Each of these connectors is associated with a specific module (e.g. the cc:Mail module connects to the cc:Mail Connector to deliver messages to a cc:Mail post office). The system administrator may set up any number of connectors for a specific user. An incoming message is then delivered to each connector defined for that user.

Each connector is identified by its name, the identifier for that connector, and the permission level. The permission level can be configured for each connector. However, Internet Exchange 4 only applies this option to the Notes and cc:Mail Connectors. These attributes as maintained by each connector (e.g. the Notes and cc:Mail migration tools automatically create LDAP accounts and update connector information). The Message Store also adds and delete connector information as required.

Click on the *View Connectors* button (see Figure 6a-9) to view the channels and identifier pair for this entry. The identifier enables the Directory Server to identify the recipient to which a specific connector is assigned. The following screen will appear:

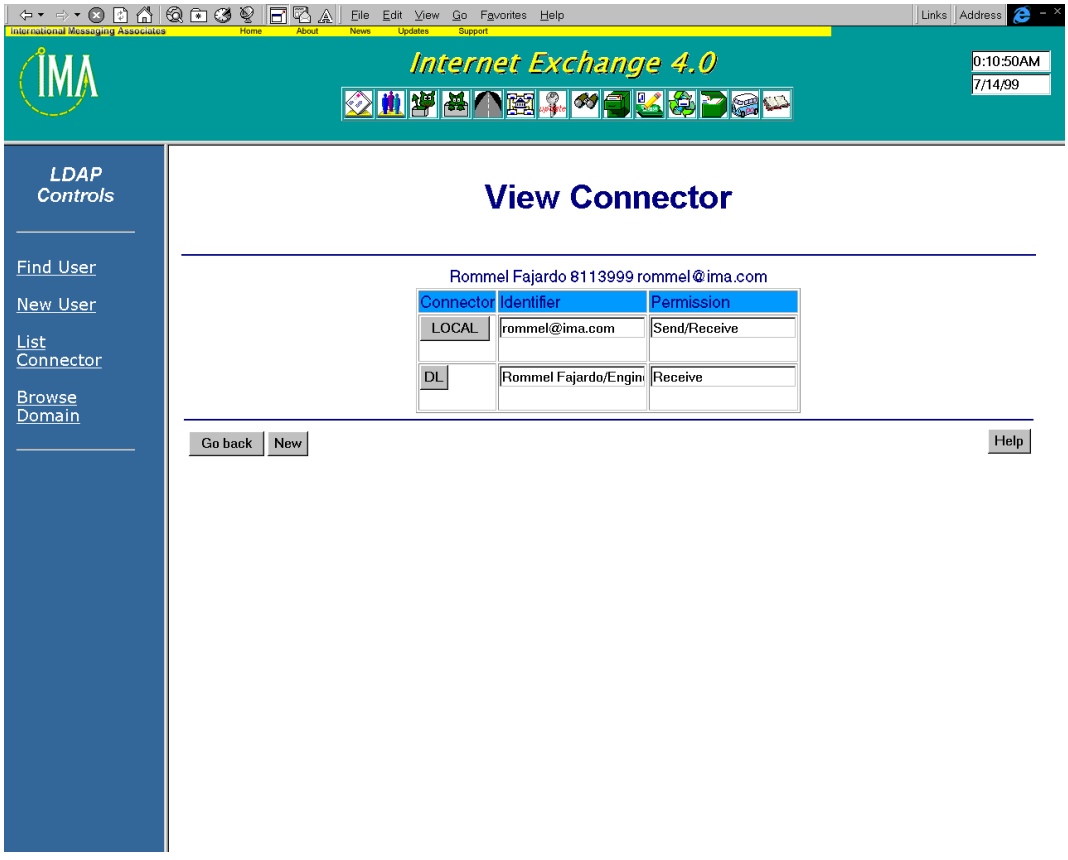

Figure 6a-10 - View Connectors

Add new connectors

To add new connectors to a particular entry, click on the New button (see Figure 6a-10). The following screen will appear:

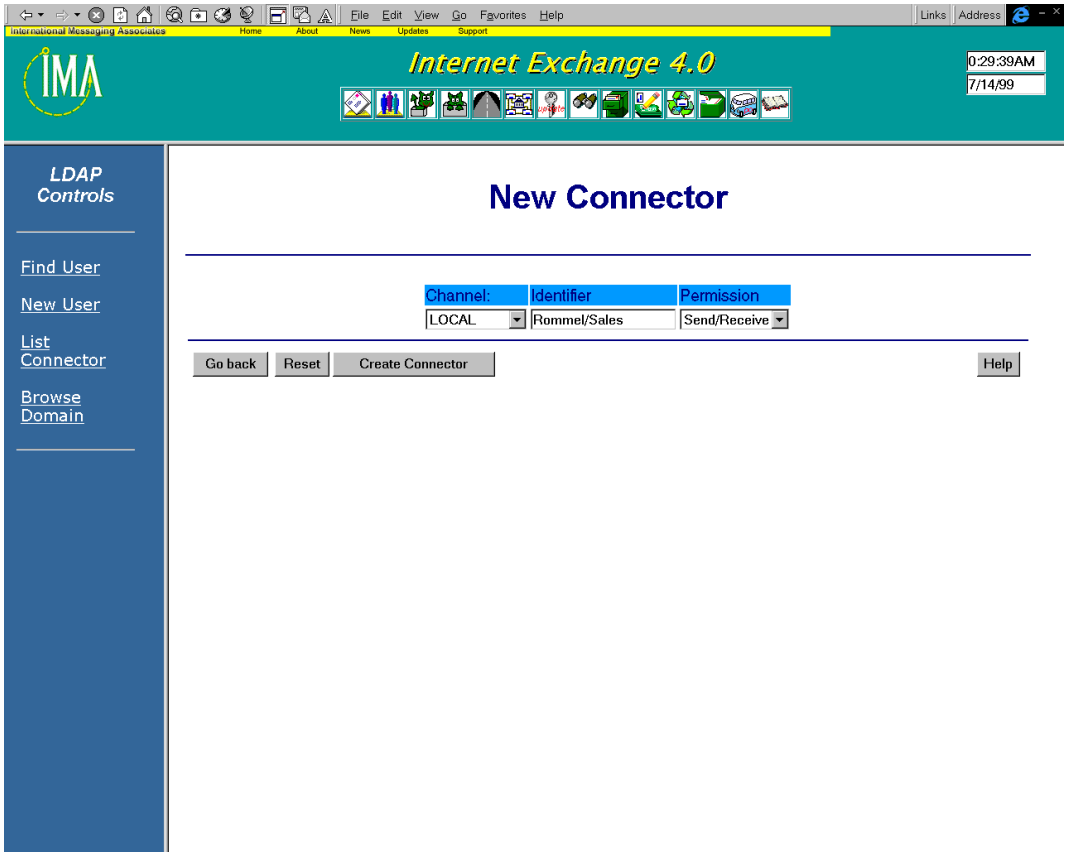

Figure 6a-11 - Add Connector

Select the new channel to be added and enter the corresponding identifier. Select the permission level from the pull-down menu. Click on the *Create Connector* button to add the new connector.

Mail Alias

To create a new mail alias for a user, click on the *New Mail Alias* button (see Figure 6a-9). the following screen will appear:

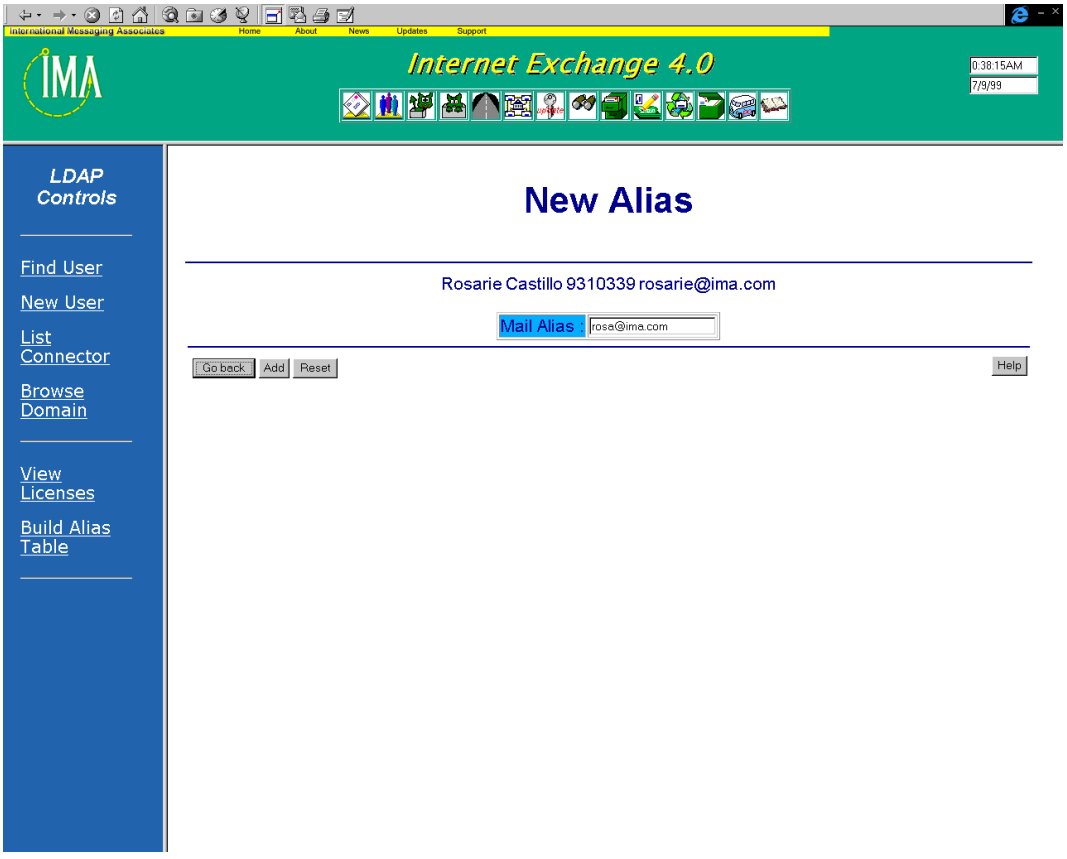

Figure 6a-12 - Create new mail alias

Enter the new mail alias for the entry selected and click on the *Add* button. This will serve as an alias email address for that particular user.

Add new user

To add a new user to the Directory Server, click on the *New User* link on the main LDAP Controls configuration screen. The following screen will be displayed:

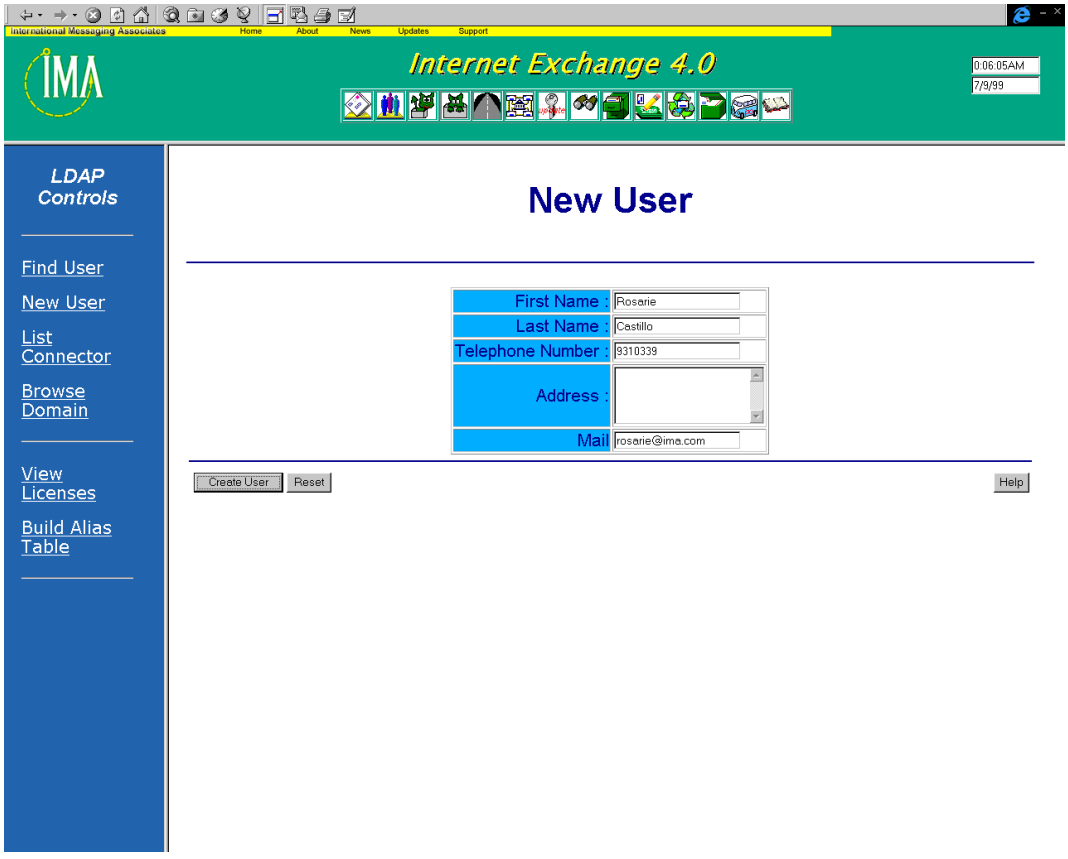

Figure 6a-13 - Add new user

First Name
The first name of the user to be added.

Last Name
The Last name of the user to be added.

Telephone Number
The contact telephone number of the user to be added

Address Name
The home/business address of the user to be added.

Mail
The valid email address of the user to be added.

After entering all the required parameters, click on the *Create User* button. The following

screen will be displayed:

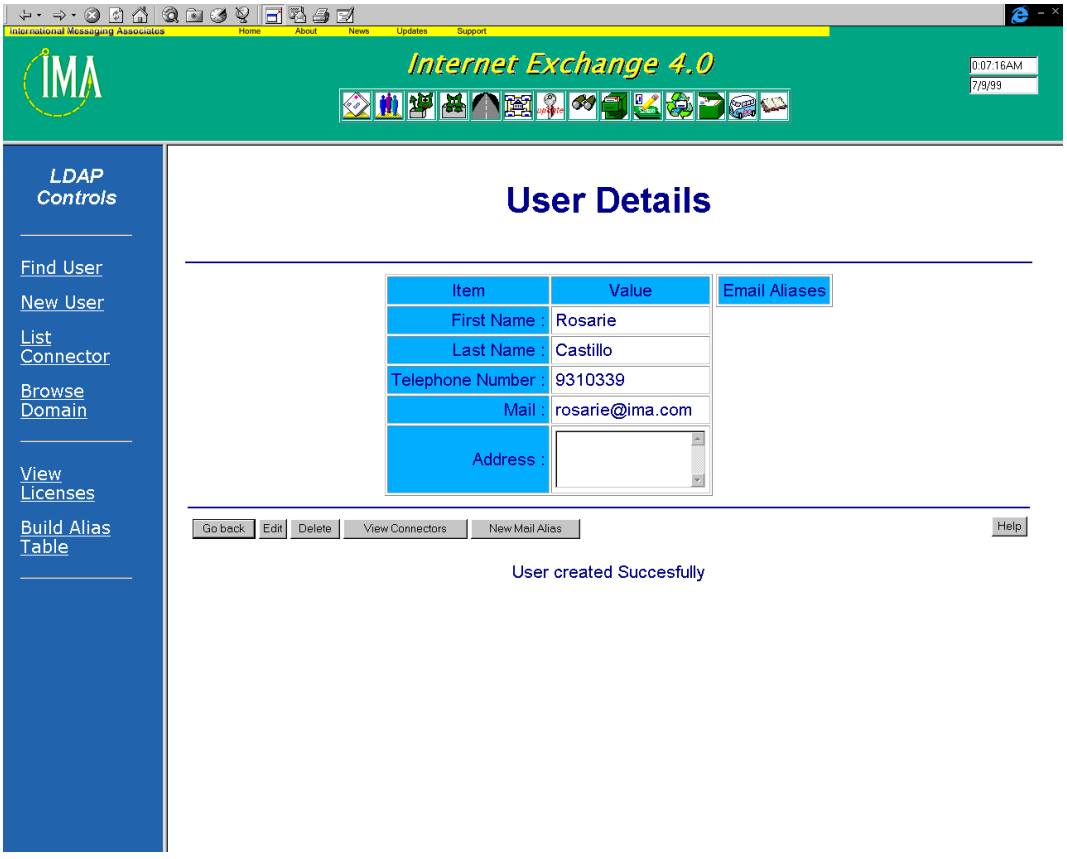

Figure 6a-14 - New user attributes

Edit user

To edit the attributes of a new user, click on the Edit button (see Figure 6a-14). A new screen for modifying user attributes will appear (see Figure 6a-15). In this screen you can change the following parameters:

- First Name
- Last Name
- Telephone number
- Address
- Mail (email)

Click on the *Update* button after making the necessary changes to save the new attributes.

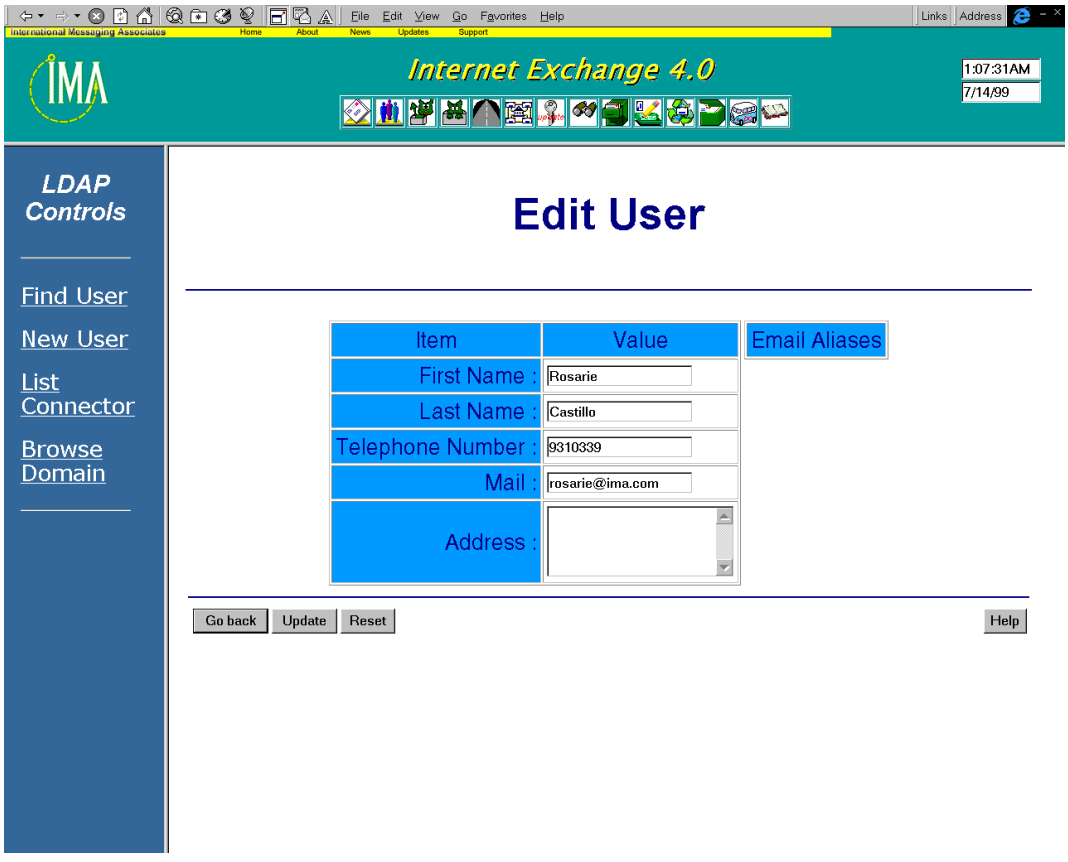

Figure 6a-15 - Edit user attributes

To create connectors for the new user, click on the *View Connectors* button (see Figure 6a-14). A new screen for viewing existing connectors will appear (see Figure 6a-10). Since the user's name is still to be added to the Directory Server, this screen will show that there are no connectors configured for the user.

To create a connector for the user, click on the *New* button. A screen for configuring **Internet Exchange 4** connectors and their attributes will appear (see Figure 6a-11). After selecting a connector for the user and specifying the connector's attributes, click on the *Create Connector* button to save the new settings.

To create a mail alias for the new user, click on the *New Mail Alias* button (see Figure 6a-9). A screen for creating a new mail alias will be displayed (see Figure 6a-12).

List connectors

To view all users for a particular connector, click the *List Connector* link on the main LDAP Controls configuration screen. The following screen will be displayed:

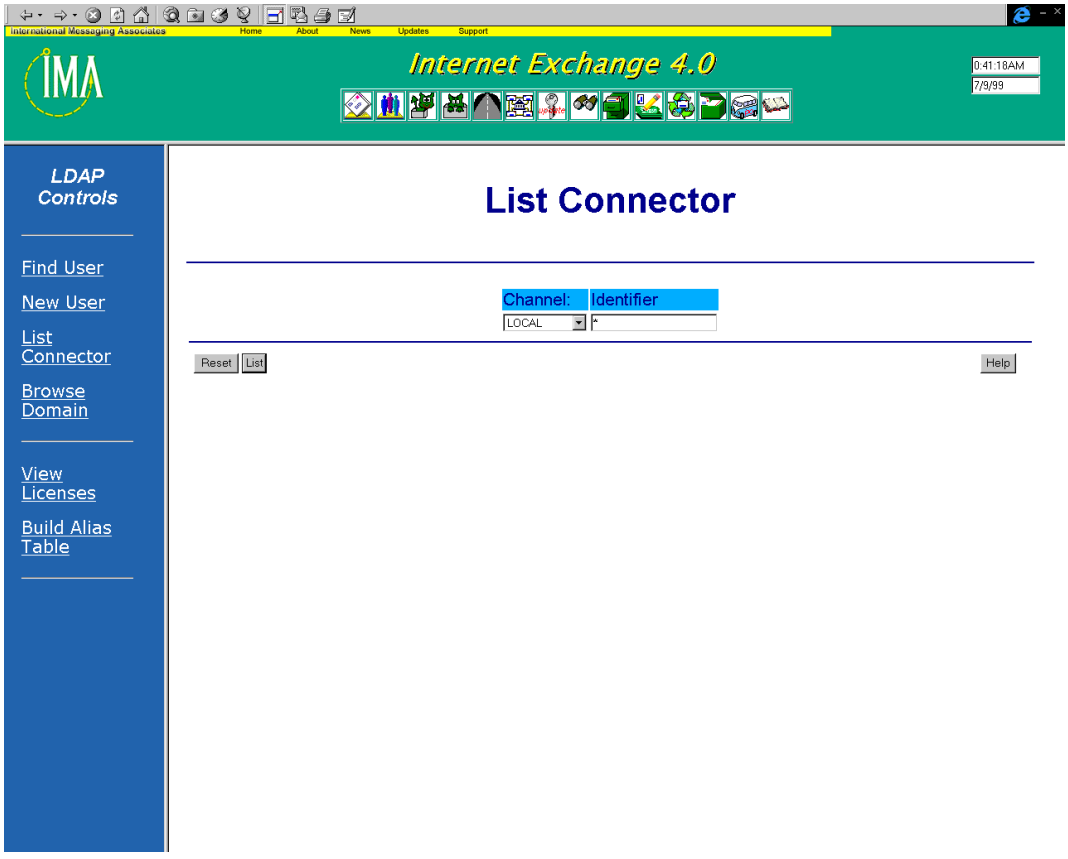

Figure 6a-16 - Select connector

Select a connector from the pull-down menu. Then click on the List button to view all users for that connector (see Figure 6a-17).

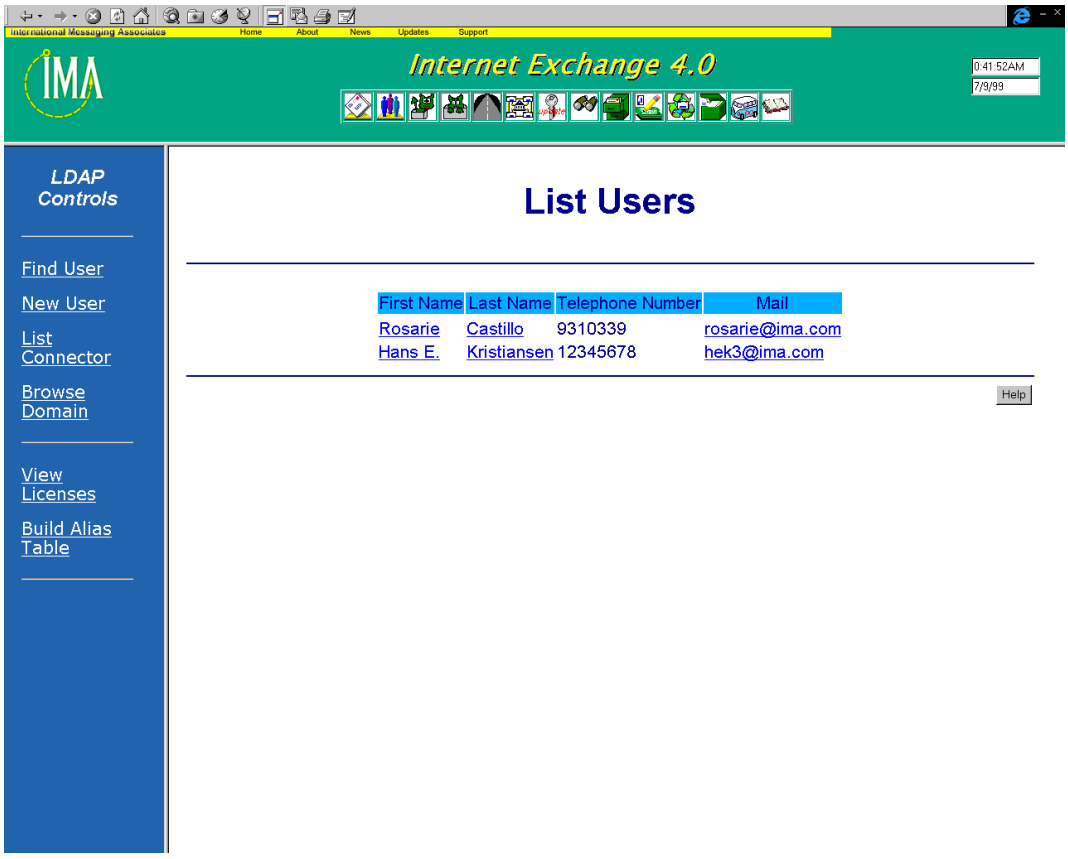

Figure 6a-17 - List connector users

Browse domain

To browse a particular domain/subdomain, click on the Browse Domain link on the main LDAP Controls configuration screen. The following screen will be displayed:

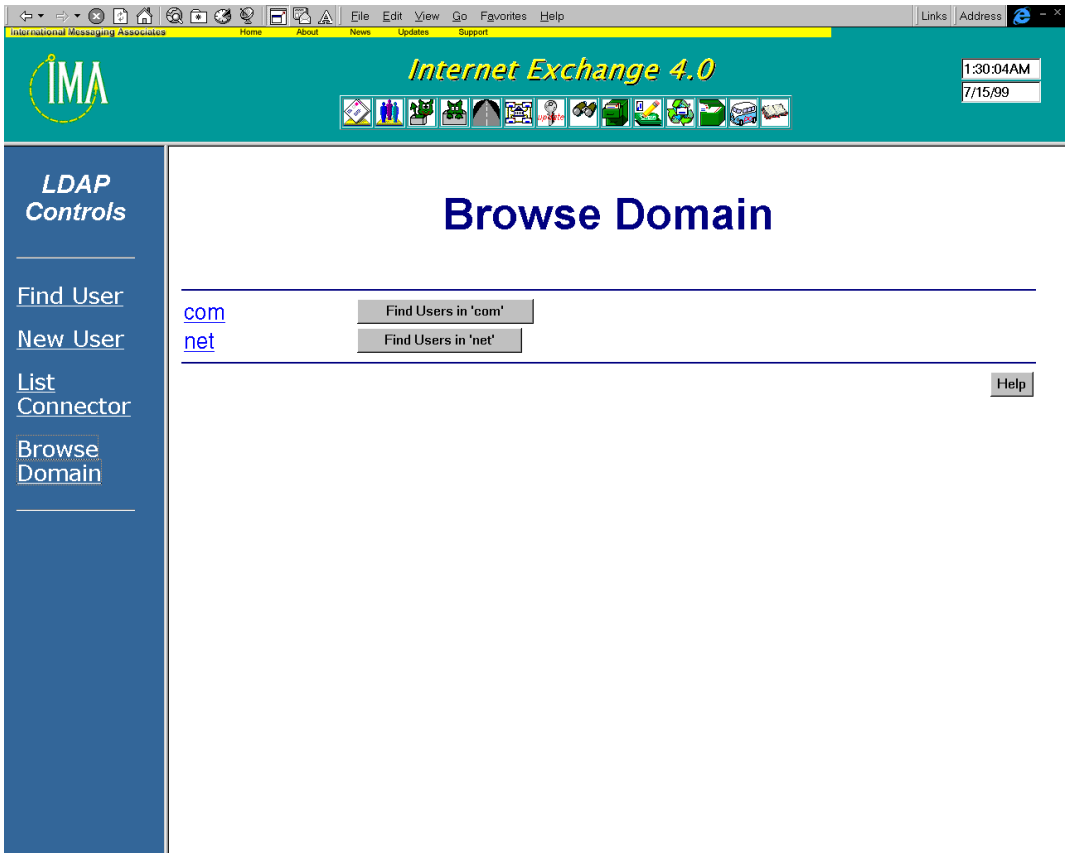

Figure 6a-18 - Select domain

This window displays different domains. To view users in the “com” domain, click on the *Find users in ‘com’* button. To view users in the “net” domain, click on the *Find users in ‘net’* button. A screen showing all the users for the domain selected will be displayed (see Figure 6a-19)

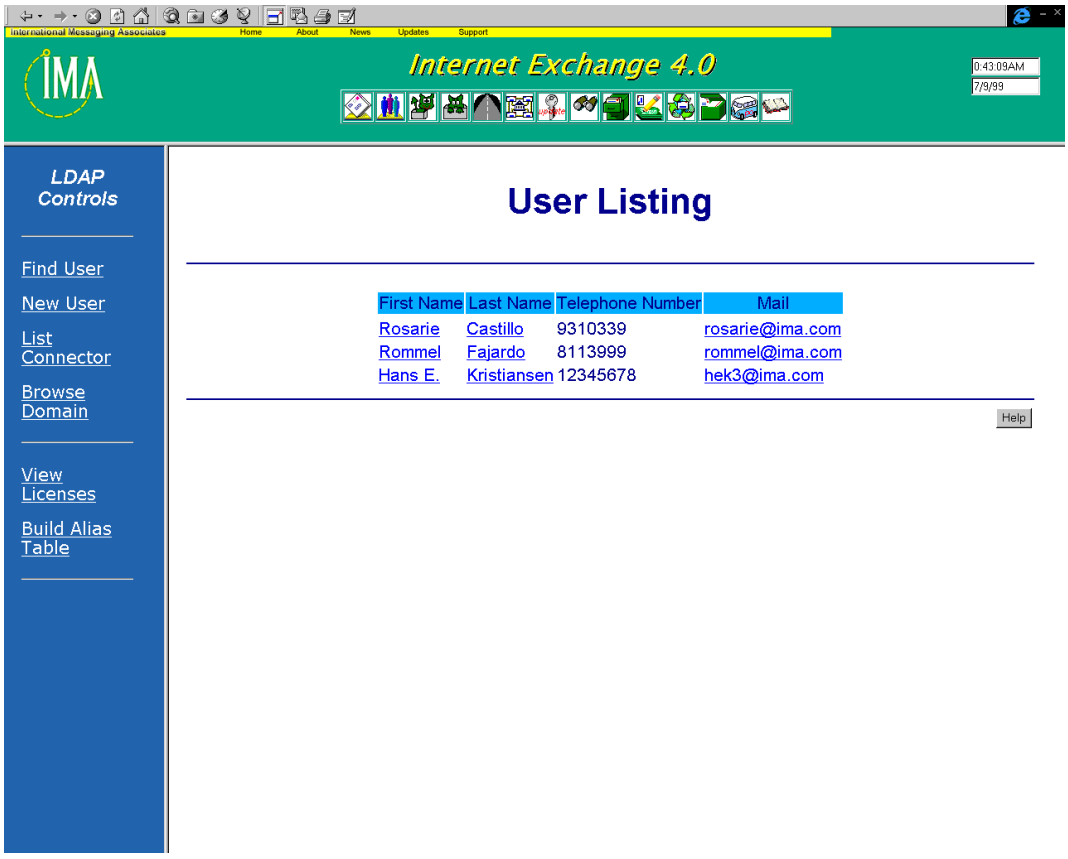

Figure 6a-19 - View domain users

CONFIGURING THE MTA

The **Internet Exchange 4** Message Transfer Agent (MTA) accepts the messages sent by the senders via their user agents (UA's). It also accepts messages transmitted to it by other MTA's for forwarding to the recipients' UA's and performs routing decisions by analyzing the recipient list in each message.

To configure the various features of the **Internet Exchange 4** MTA, click on the MTA icon on the main Web Administration Interface. The following screen will appear:

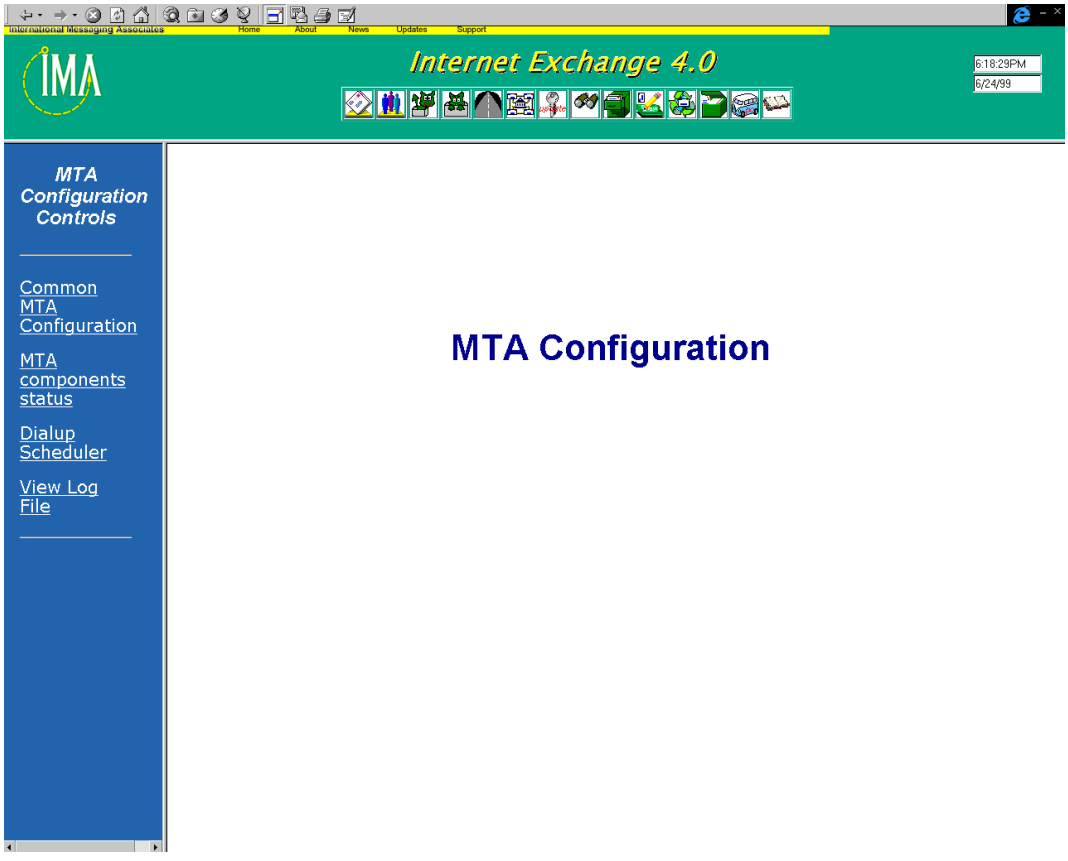

Figure 6a-20 - Main MTA Configuration Screen

Common MTA configurations

To configure common MTA options, click on the *Common MTA Configuration* link. A new screen will be displayed (see Figure 6a-21).

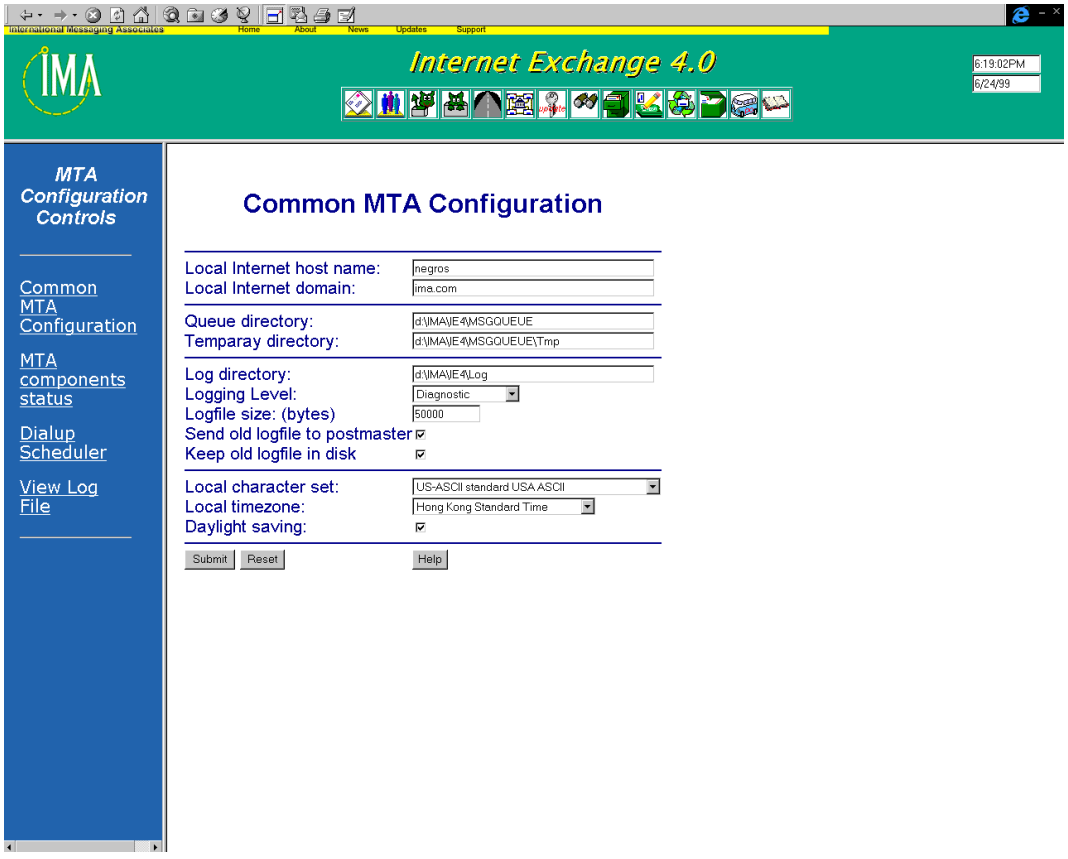

Figure 6a-21 - Common MTA parameters

Local Internet host name

The Internet host name of the machine that runs the Internet Exchange Messaging Server (IEMS).

Local Internet domain

The Internet domain name of the machine that runs the IEMS (e.g. ima.com).

Queue directory

The directory under which the Notes Connector stores the configuration databases, including the MIME mappings database (MAGIC.BTR), peers capabilities database (PEER.BTR), and the certifier-to-Internet domain mappings database (LNPOD.BTR).

Temporary directory

The location of the IEMS temporary directory. The Notes Connector needs to write temporary files during message conversion process. Such files are stored here.

Log directory

The location of the IEMS log files directory. The log file IEMTA.LOG is written to this directory. You can set this directory to shared directory in the network so that you can read the file remotely on a user station. Doing so, however, may degrade the performance of the

software as writing data via network is generally slower than writing data directly on to local hard disk.

Logging level

IEMS offers four levels of debugging, namely:

- Errors only
  Only erroneous activities are logged.

- Message logging
  Information about the delivery of all messages is logged.

- SMTP session
  All SMTP conversations are logged. This level records each incoming and outgoing SMTP command.

- Diagnostic
  Additional diagnostic data is logged including information concerning core operations. This option is for debugging purposes and is not usually needed. Due to the large amount of debugging information produced, this level of logging is recommended only for situations where very detailed logging information is required. This is because under the Diagnostic mode, extensive logging activity will slow down the operation of the gateway.

Logfile size

The largest logfile size permitted before it is saved in another name and a new log is started. The default limit is 50,000 bytes, allowing the Windows Notepad application to read the file. Acceptable values are in the range between 10,240 bytes (10Kb) and 2,000,000,000 (roughly 2Gb). The default value of zero indicates no limit.

Send old log file to postmaster

This option causes old logfiles to be automatically mailed to the postmaster.

Keep old log files in disk

Prevents deletion of old log files. Storage of such files, however, uses up disk space very rapidly and the administrator should deal with them regularly.

Local character set

Allows a character set identifier to be tagged to all outgoing mail. For recipients in most Anglo-Saxon countries, US-ASCII should be used. Those in other countries, meanwhile, will have to choose a different ISO character set. For Japanese users, ISO-2022-JP should be used.

Local time zone

Select from the list of locations offered. If the local timezone is not listed, then the desired time zone must be entered manually into the IEMTA.INI file using a editor as follows:

[Gateway]
Timezone=tzn[+|-]hh[:mm[:ss]] [dzn]

where *tzn* must be a three-letter time-zone name, such as PST, followed by an optionally signed number, hh, giving the difference in hours between UCT and standard time. To specify exact local time, the hours can be followed by minutes, :mm; and seconds, :ss; and if applicable, a three-letter daylight-saving-time zone, dzn, such as PDT.

If the timezone value is not set, the default is PST8PDT, which corresponds to the Pacific timezone of the USA. If the timezone "Use system TZ variable" is selected, the timezone information is then obtained from the user defined TZ environment variable. Under Windows 95, this can be set in the AUTOEXEC.BAT system startup file. Under Windows NT, it is usually set in the system registry. In either case, the machine must be rebooted in order to make the change effective.

Daylight saving

Indicates whether the local timezone uses daylight saving during summer.

MTA Component Status

To view the status of the different MTA status, click on the *MTA components status* link on the main MTA configuration page. The following screen will be displayed:

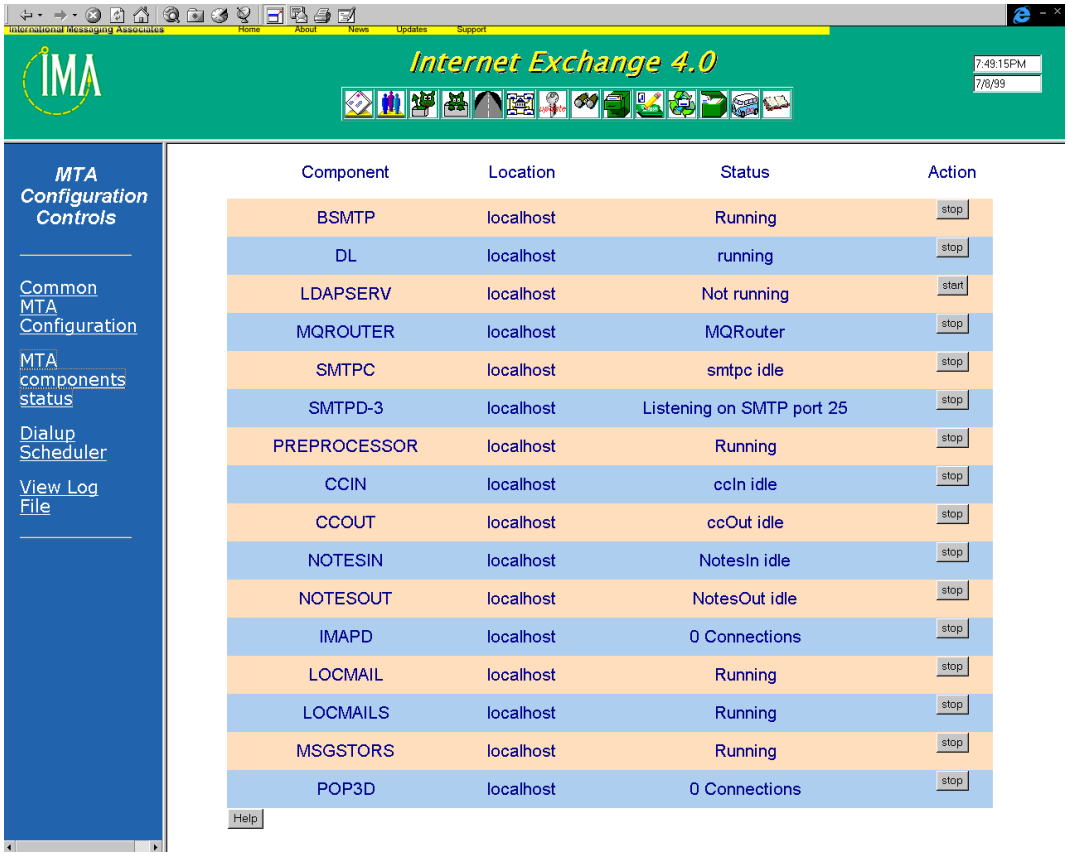

Figure 6a-22 - View MTA components status

Component
Identifies the specific MTA component/module, for example SMTPD. If there are multiple SMTPD threads running on the entire system across different machines, you will see SMTPD-1, SMTPD-2 and so on.

Location
Displays the TCP/IP host name of the machine running the component.

Status
Displays the current status returned by the component.

Action
Displays the action that can be implemented on the component.

- If the component is already running, a *Stop* button is displayed.
- If the component is not running, a *Start* button is displayed.
- If the component is not installed, a *No Action* button is displayed.

Dial-up Scheduler

The **Internet Exchange 4** Dialup Scheduler allows the system administrator to choose which days of the week to run the dialup schedule for Remote Access service (RAS). RAS is the remote access service for Windows and is actively supported on all WIN32 platforms. It is a useful feature not only for dialup issues but also for any Windows supported dialup mechanism.

The Dial-up Scheduler supports the following functions:

- Provides a user interface to enable the gateway administrator to configure dialup schedules and other RAS connection-related profiles
- Performs RAS dialup at the scheduled dialup time
- Performs RAS connection hang-up at the scheduled hang-up time

Before configuring Dial-up Scheduler, the Windows system must be configured for the appropriate dial-up mechanism. For Windows 95/98 and Windows NT 4.0, use Windows **Setup/Programs/Accessories/Dial-up Networking** to configure the appropriate dialup mechanism.

To configure the Dial-up Scheduler, click on the *Dialup Scheduler* link on the main MTA configuration page. The following screen will appear:

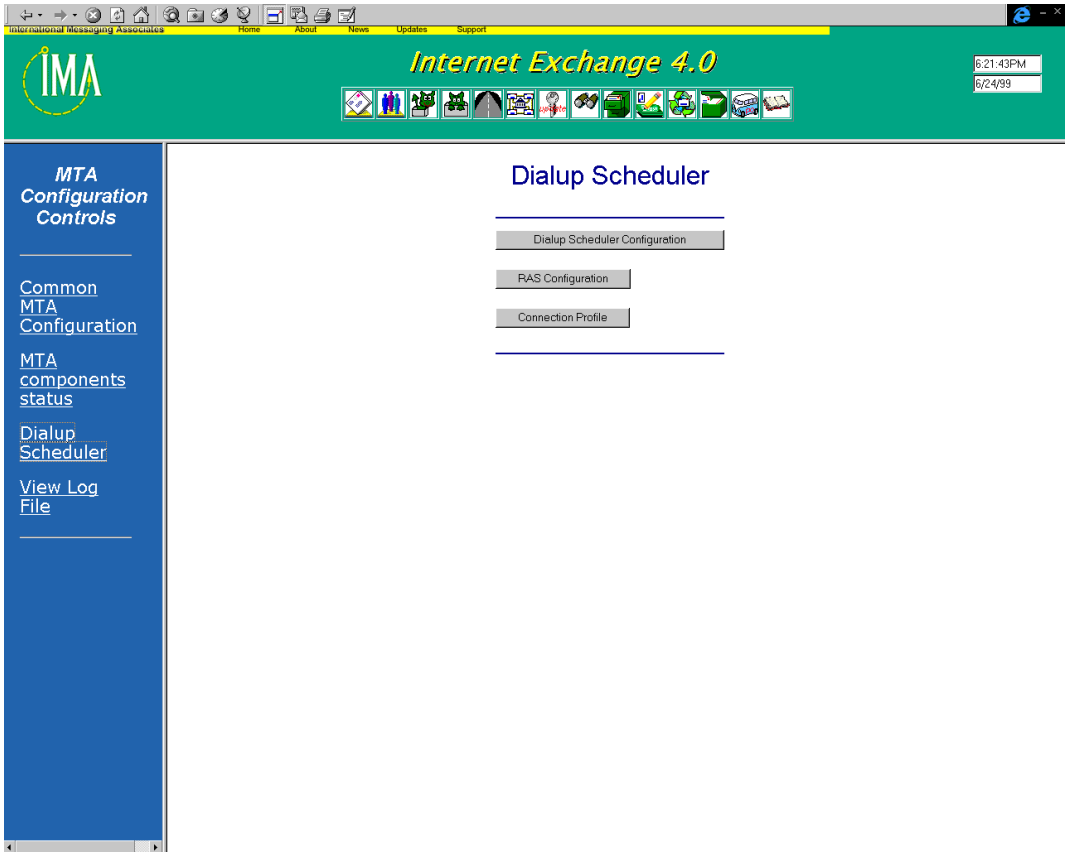

Figure 6a-23 - Main Dial-up Scheduler Configuration Page

Click the *Dial-up Scheduler* button to view the Dial-up Scheduler configuration screen (see Figure 6a-24).

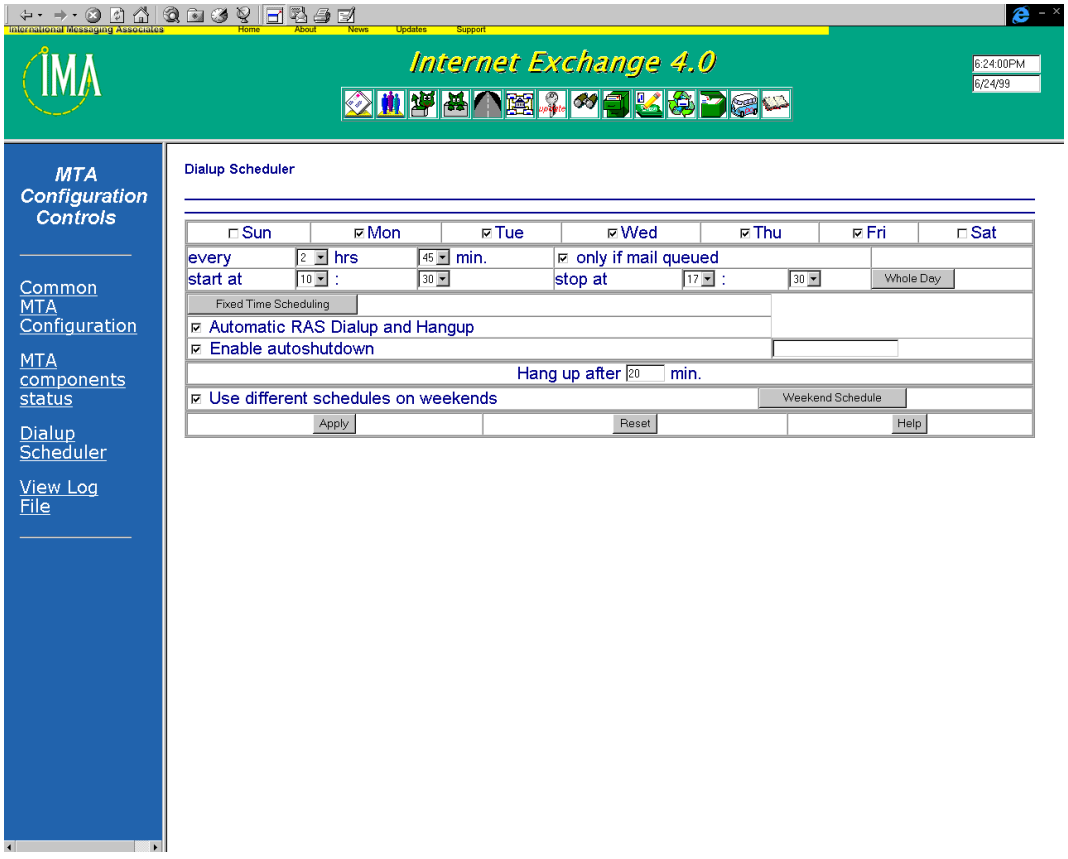

Figure 6a-24 - Configuring Dial-up Scheduler

Select the day(s) when the Dial-up Scheduler should run.

Sun
Specifies that dial-up is to be executed every Sunday.

Mon
Specifies that dial-up is to be executed every Monday.

Tue
Specifies that dial-up is to be executed every Tuesday.

Wed
Specifies that dial-up is to be executed every Wednesday.

Thu
Specifies that dial-up is to be executed every Thursday.

Fri
Specifies that dial-up is to be executed every Friday.

Sat
Specifies that dial-up is to be executed every Saturday.

Every
Selects periodic dial-up schedules, with the period specified by the hour and the minute settings.

Only if mail queued
Specifies that the Dial-up Scheduler checks if there are mail queued in the SMTPOut queue before establishing a dial-up connection. If there are no mail in the queue, the Dial-up Scheduler will not attempt the dial-up. This option is only valid for the periodic dial-up schedule.

Start at/Stop at
Specify the start time and the end time of the periodic dial-up schedule. Periodic dial-ups will be allowed within this time interval.

Whole Day
Configures the periodic dial-up schedule to remain active throughout the whole day.

Fixed Time Scheduling
Configures the Dial-up Scheduler to perform only one dial-up on every scheduled day. Click on the *Fixed Time Scheduling* button to display the fixed scheduling options screen (Figure 6a-25).

Automatic RAS Dialup and Hang-up
Activate the *Automatic RAS dialup and hang-up* option to enable RAS support. With this function enabled, Internet Exchange automatically starts a RAS connection during gateway startup. When the gateway shuts down, the RAS connection will terminate automatically.

Enable autoshutdown
Activate the *Enable autoshutdown* option to enable automatic shutdown of the RAS connections.

Hang-up Time
This parameter specifies the time (in minutes) that the RAS Dial-up Scheduler should hang-up a connection after the connection has been established.

Use different schedule on weekends
This option, when enabled, specifies that a different dial-up schedule is to be used for the weekends (i.e.. Saturdays and Sundays). The schedule for the weekends can be configured by clicking on the Weekend Schedule button to bring up a dialog box for configuring the weekend dial-up schedule (Figure 6a-26).

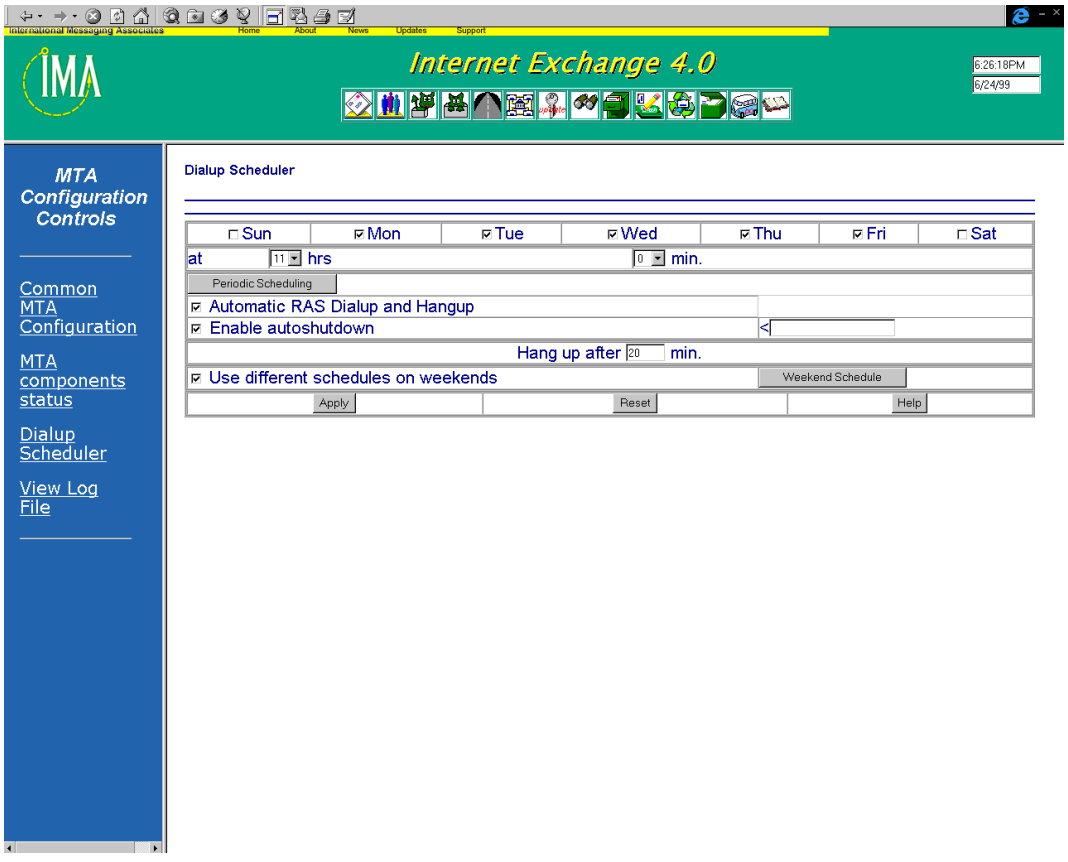

Figure 6a-25 - Fixed time scheduling window

Weekend Dialup Schedule

When activated, this allows the system administrator to configure the dialup schedules for the weekends. Click on the *Weekend Schedule* button to display the Weekend Schedules configuration screen (Figure 6a-26).

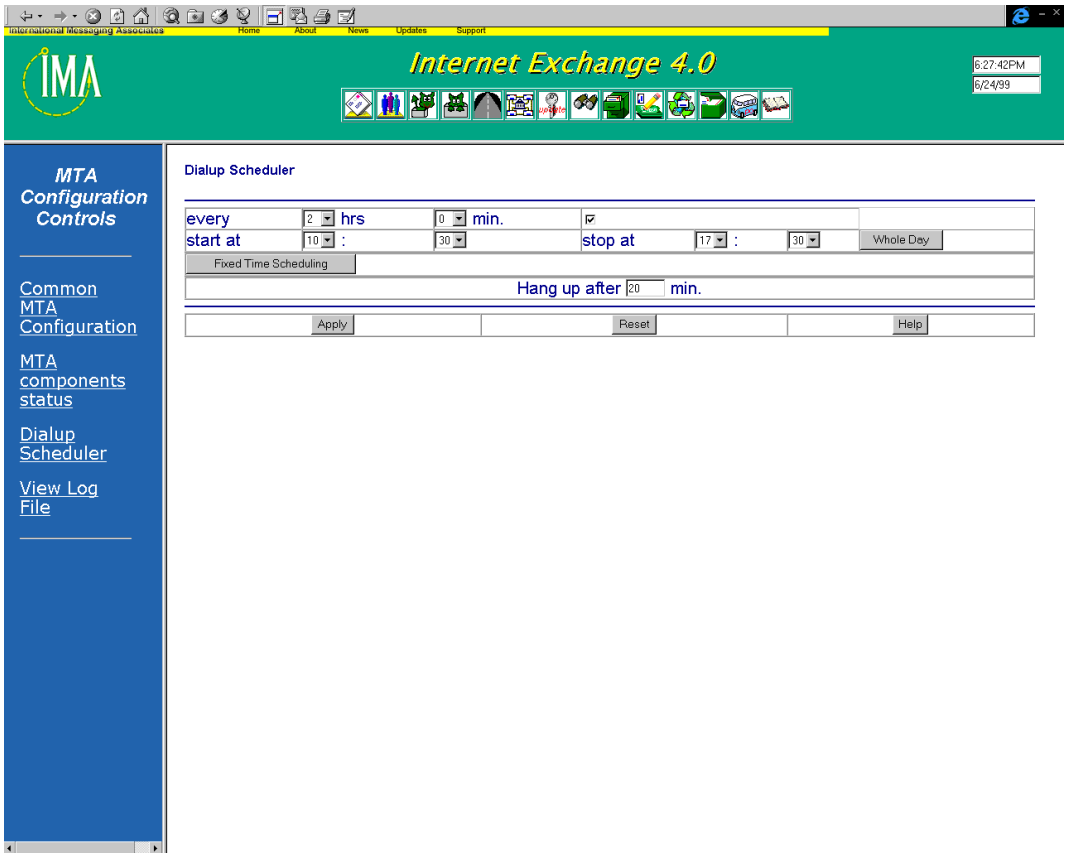

Figure 6a-26 - Configure weekend dialup schedules

Periodic Scheduling

When enabled, this configures the Dial-up Scheduler to perform periodic dial-up operations on every scheduled day. Click the *Periodic Scheduling* button to display the Periodic Scheduling configuration window (Figure 6a-27).

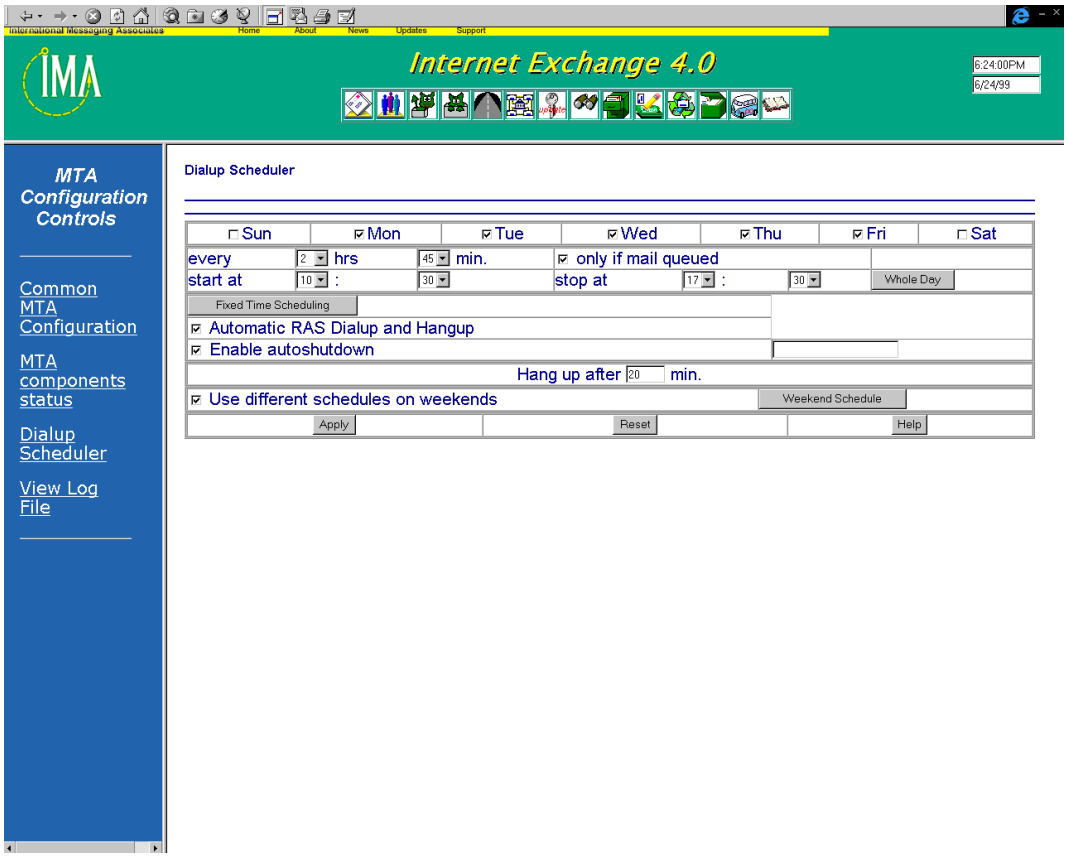

Figure 6a-27 - Configure periodic dialup schedules

Automatic RAS Dialup and Hang-up

Activates the *Automatic RAS dialup and hang-up* option to enable RAS support. With this function enabled, Internet Exchange automatically starts a RAS connection during startup. When the Internet Exchange shuts down, the RAS connection terminates automatically.

Enable autoshutdown

Allows automatic shutdown of the RAS connection.

Hang-up Time

Specifies the time (in minutes) that the RAS Dial-up Scheduler should hang up a connection after the connection has been established.

Use different schedule on weekends

Specifies that a different dial-up schedule is to be used for the weekends. The schedule for the weekends can be configured by clicking on the *Weekend Schedule* button to bring up a dialog box for configuring the weekend dial-up schedule.

RAS Configuration

To configure RAS settings, click on the *RAS Configuration* button on the main Dial-up Scheduler configuration screen. The following screen will be displayed:

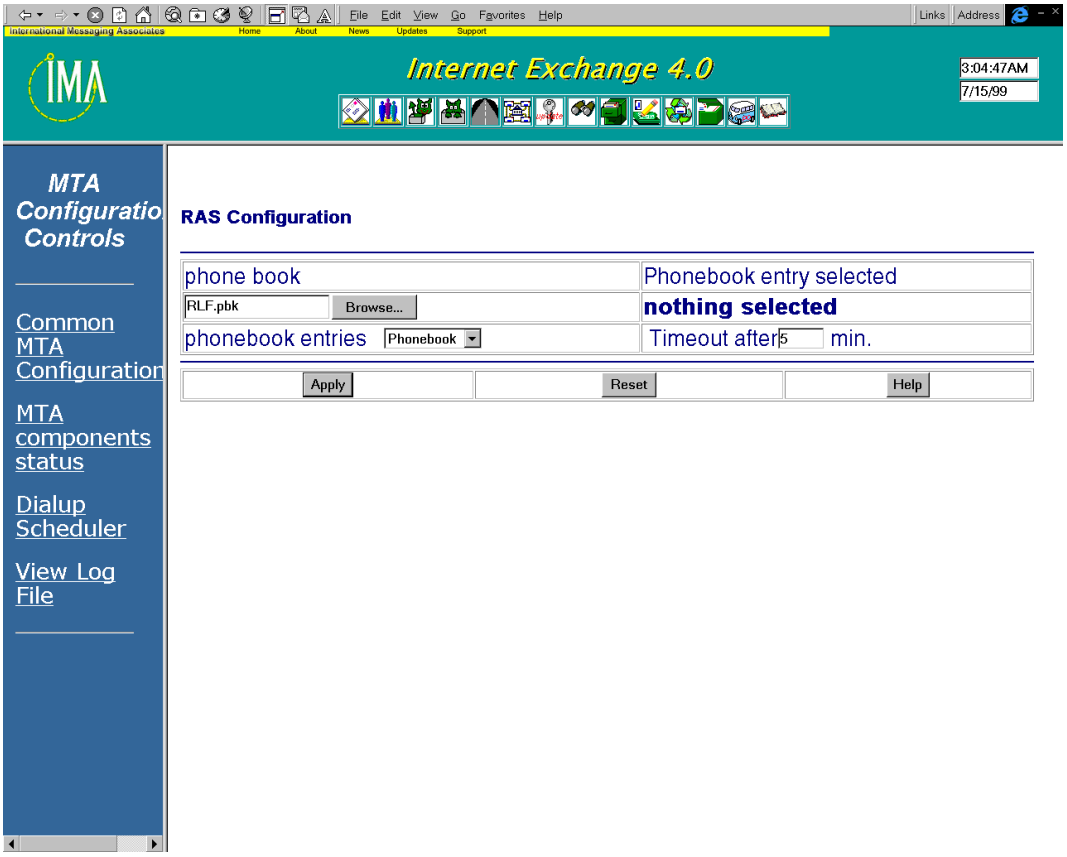

Figure 6a-28 - RAS Configuration Page

Phonebook

The first RAS Configuration entry, Phonebook, allows the gateway administrator to specify the phone book entry to be used by Internet Exchange for RAS connection. If Internet Exchange is running on Windows 95, the only possible entry is System Phone Book. More than one phonebook can be chosen under Windows NT; use the Browse button to search through the file system for other phonebooks (files with the .PBK extension).

Phonebook entry selected

Displays the RAS profile name to be used. Internet Exchange uses the RAS profile name for making RAS connection during start up.

Phonebook entries

Displays the first number to be tried during dialups. The phonebook contains several entries which are tried by the Dial-up Scheduler based on their order in the phonebook.

Timeout after

Specifies the timeout value (in minutes). The Dial-up Scheduler waits for a RAS dial-up connection to be established. If the RAS dial-up connection fails, the Dialup Scheduler

will redial automatically until the timeout value is reached.

Click the *Apply* button to save all the settings for the RAS configuration. The *Reset* button clears all the new entries.

Connection Profile

Click on the *Connection Profile* button on the main Dial-up Scheduler window to configure different aspects of the RAS connection itself. The following screen will be displayed:

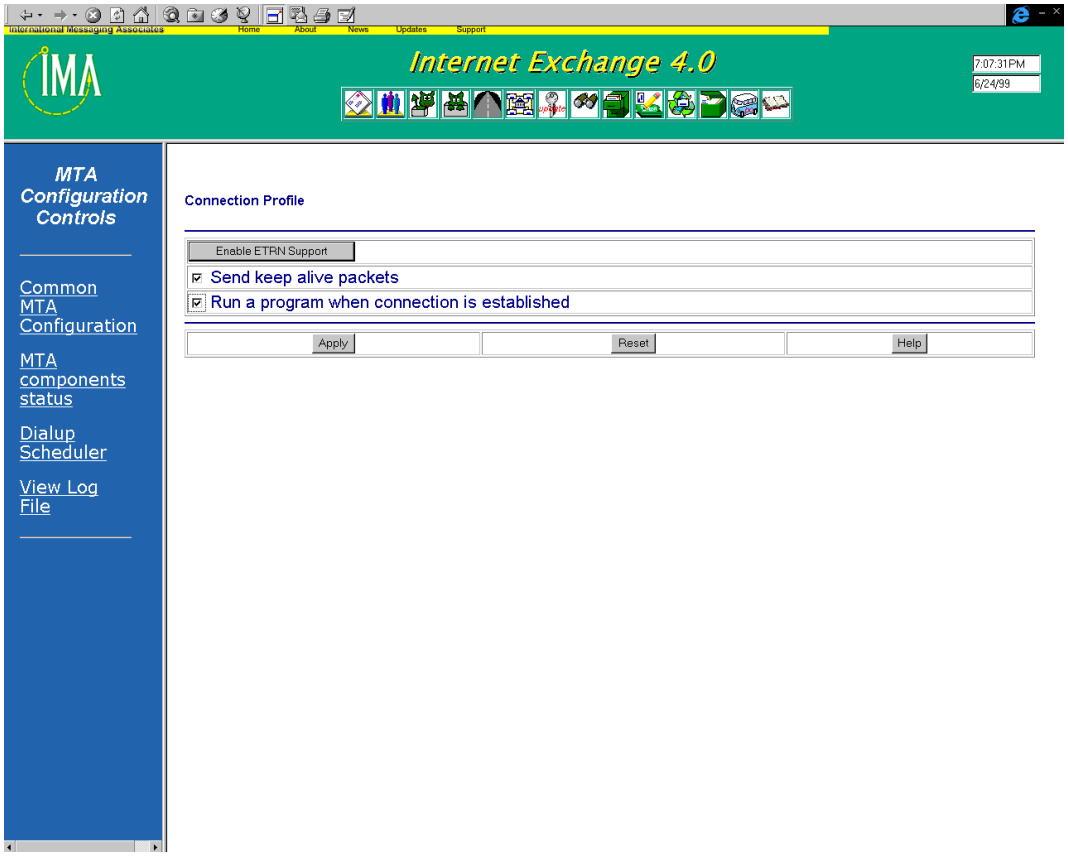

Figure 6a-29 - Connection Profile

Disable/Enable ETRN Support

Specifies the machine's FQDN to all remote SMTP hosts when the gateway is sending out mail during the dialup connection.

Alternate Name List

Enables the gateway to send ETRN requests that specify its alternate name list to all remote SMTP hosts.

Send ETRN

Sometimes it might be useful to enable sending ETRN request only to a specific host to which there might not be any outbound mail. This ensures that even though there is no outbound mail to that host when SMTPC runs, the host still receives ETRN requests. An

option to add/delete hostnames is also available. To add a hostname, enter the hostname and click the *Add* button. To remove a particular hostname from the list, select an entry from the list and click the *Delete* button.

Send keep alive packets

Click on Send keep alive packets to enable this option. For TCP connections established over a dialup connection (typically PPP or some ISDN connections), some TCP/IP stacks can be configured to time out and automatically disconnect after a predetermined period of zero network activity. Under this condition, it is necessary for the gateway to keep the stack active if SMTPD is to continue to be able to receive incoming mail. This option enables SMTPD to keep sending alive packets to maintain the dialup connection.

Run a program

Click on Run a program when connection is established to enable this option. This option also allows the administrator to define the path of the program to be run after the connection is made. Extra parameters regarding this function can be entered in the next section.

Click the *Apply* button to immediately implement the settings or click on *Reset* to discard the changes.

View Log File

Internet Exchange logs the transactions for each operation that has been carried out. An archive of the old log files can be viewed using this option. To view a log file, click on the *View Log File* link on the main Dial-up Scheduler configuration window. The following screen will appear:

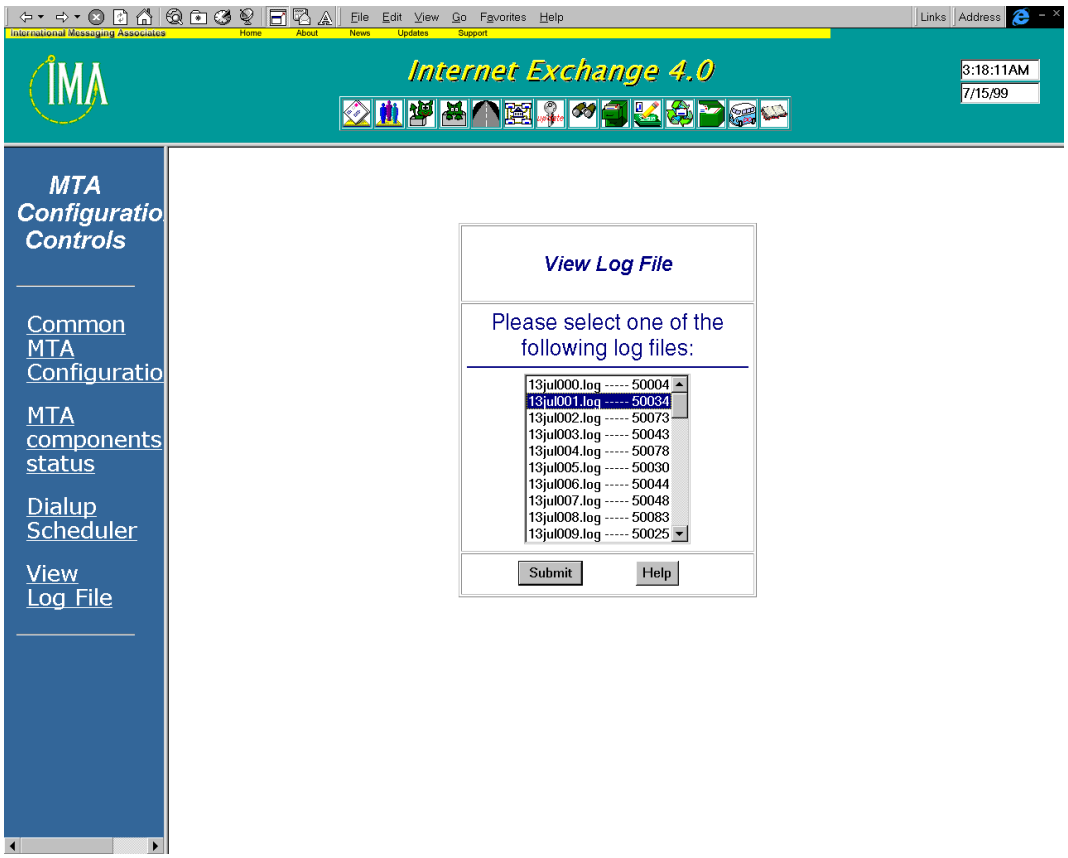

Figure 6a-30 - Select log file for viewing

Select the log file that you wish to view from the list. Then click on the *Submit* button to view the contents of the selected log file (see Figure 6a-31).

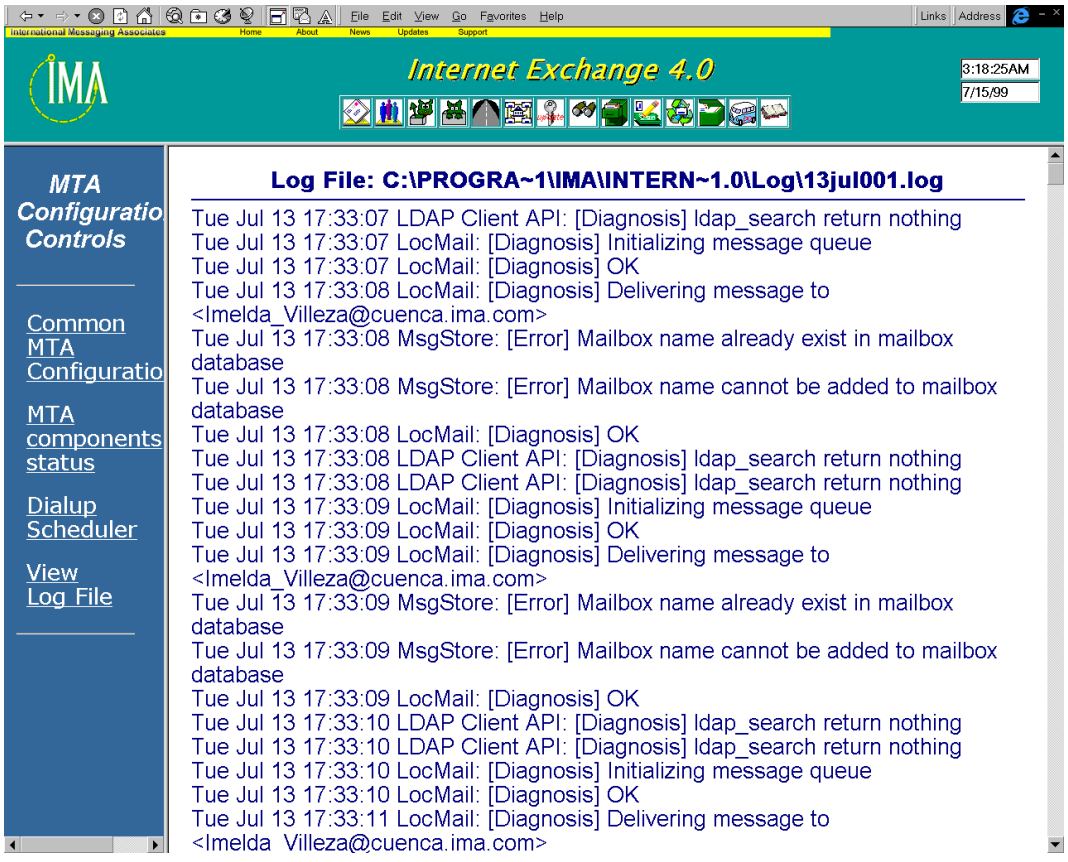

Figure 6a-31 - View log file

Chapter 7

End User Administration

INTRODUCTION

Internet Exchange 4 end users are provided with a Web-based Administration Interface for configuring the following modules:

- LDAP Directory Server
- Message Store
- Distribution List Manager

To log on to the End User Web Administration Interface, go to the **Internet Exchange 4** Authentication Page (see Figure 7a).

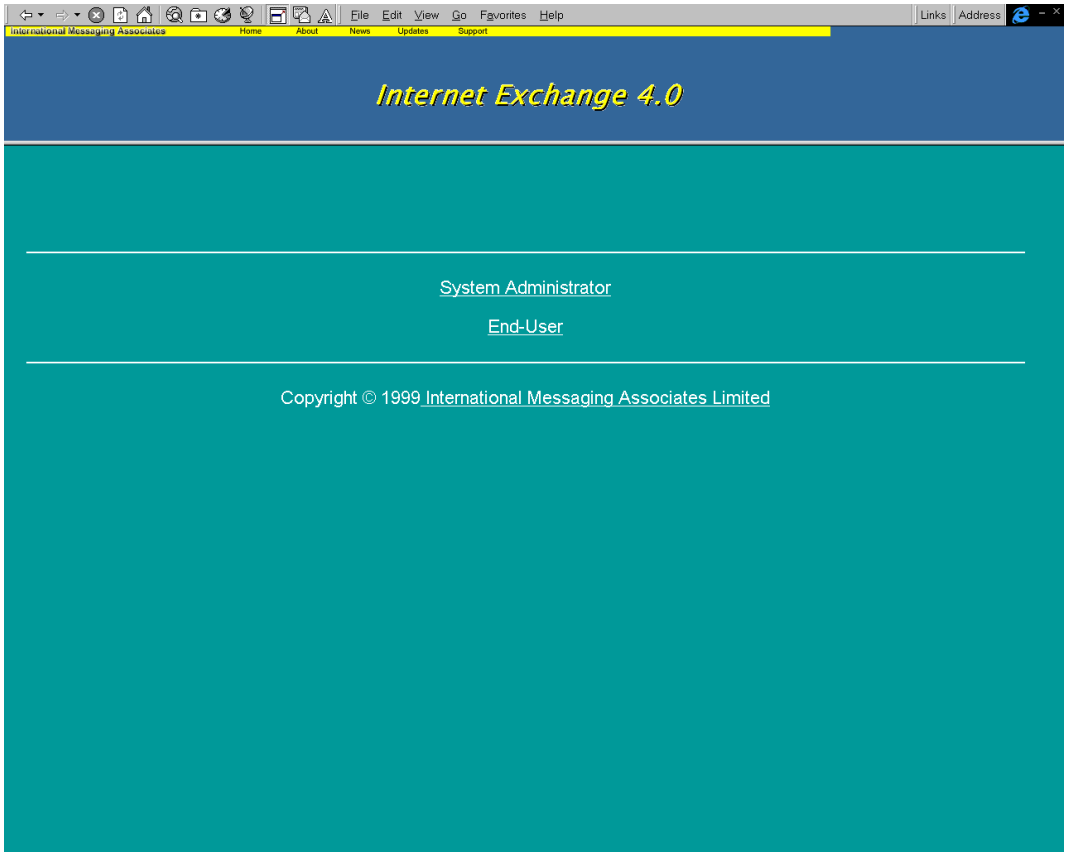

Figure 7a - End user log on screen page

Click on the *End-User* link. The authentication page for end users will appear (see Figure 7b).

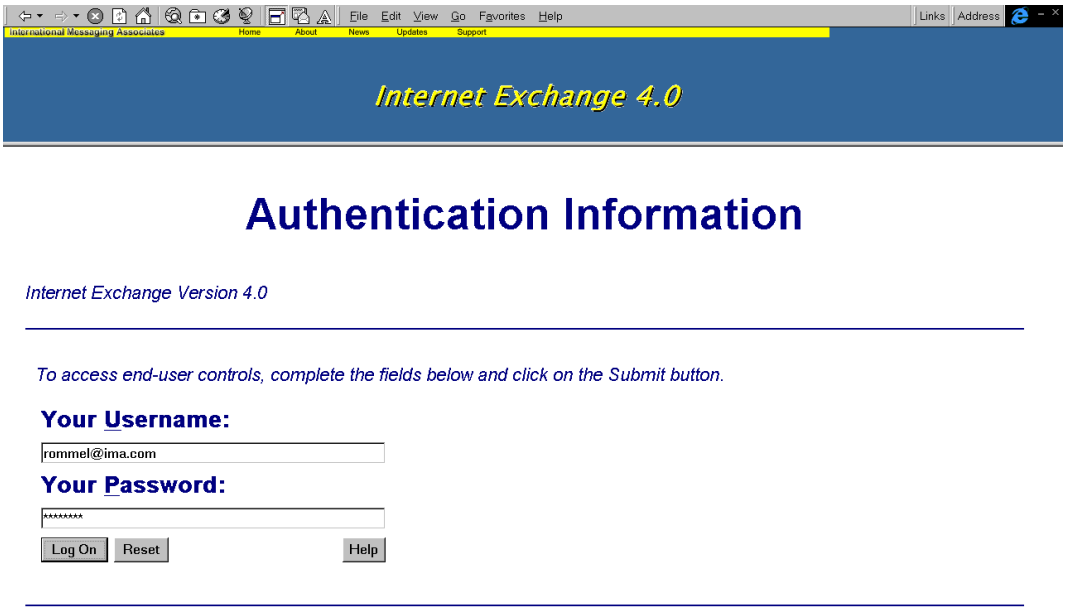

Figure 7b - Authentication Page

User Name

The email address of the user as it is entered in the Message Store.

Password

The password for the user. The password will appear as a row of asterisks for security purposes.

After entering the user name and password in the text boxes provided, click on the *Log On* button. If the user name and password are verified to be correct, the Main Web Administration Interface for end users will appear (see Figure 7c).

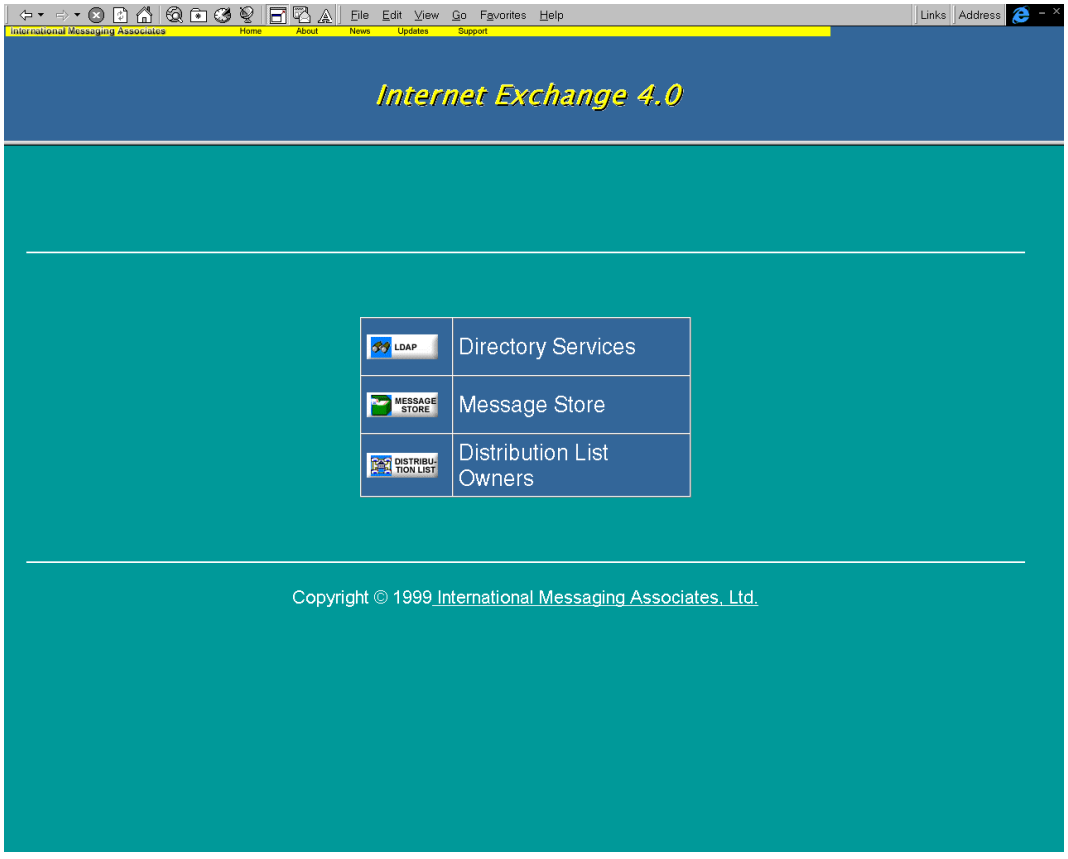

Figure 7c - Main Administration Interface for end users

DIRECTORY SERVER

To configure the LDAP Directory Server, click on the *LDAP* icon on the Main Web Administration Interface. The following screen will be displayed:

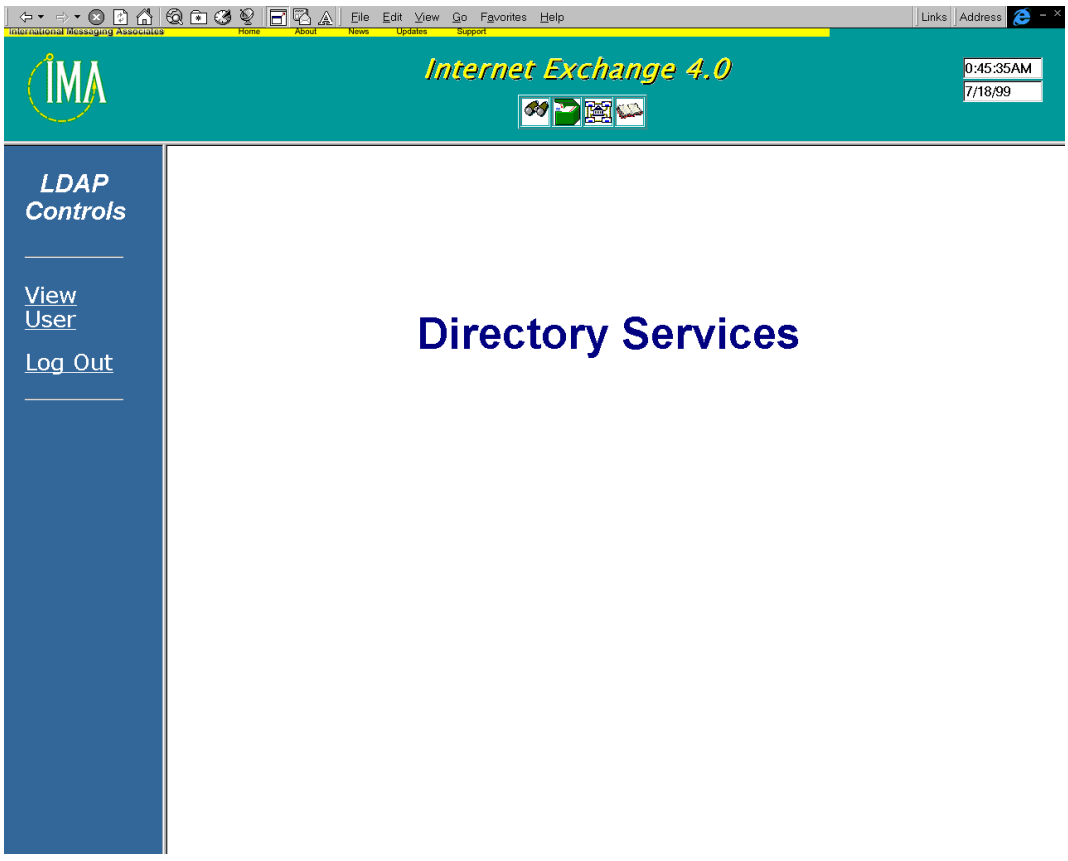

Figure 7d - LDAP Controls Configuration Page

View user

To view your profile, click on the *View User* link. A new page displaying various user attributes (i.e. first name, last name, telephone number, address, etc.) will appear (see Figure 7e).

Log Out

To log out of the LDAP Directory Server, click on the *Log Out* link at the left portion of the screen.

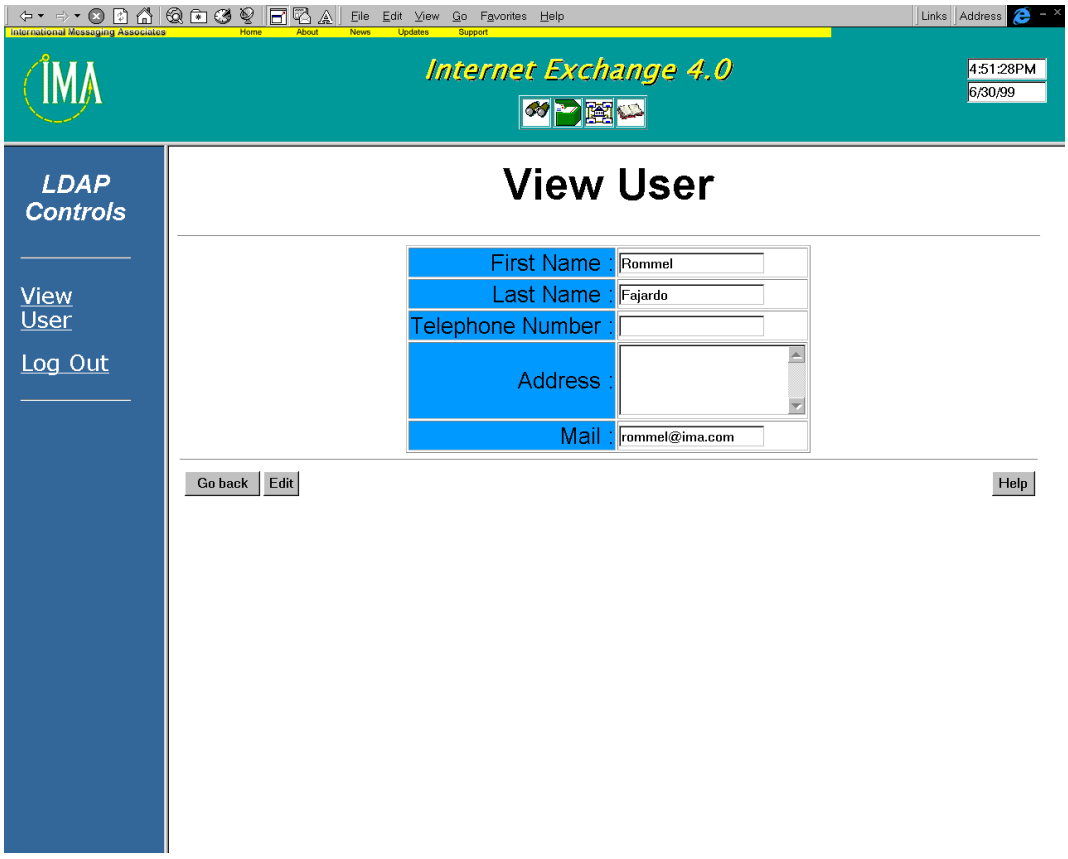

Figure 7e - View user attributes

Edit user profile

To modify the different user attributes, click on the Edit button. A new screen for editing the user profile will be displayed.

MESSAGE STORE

End users are provided with a Web-based interface for modifying their MailSort configuration and for updating passwords. To update/edit these parameters, click on the Message Store icon on the main administration interface (see Figure 7c). The following screen will be displayed:

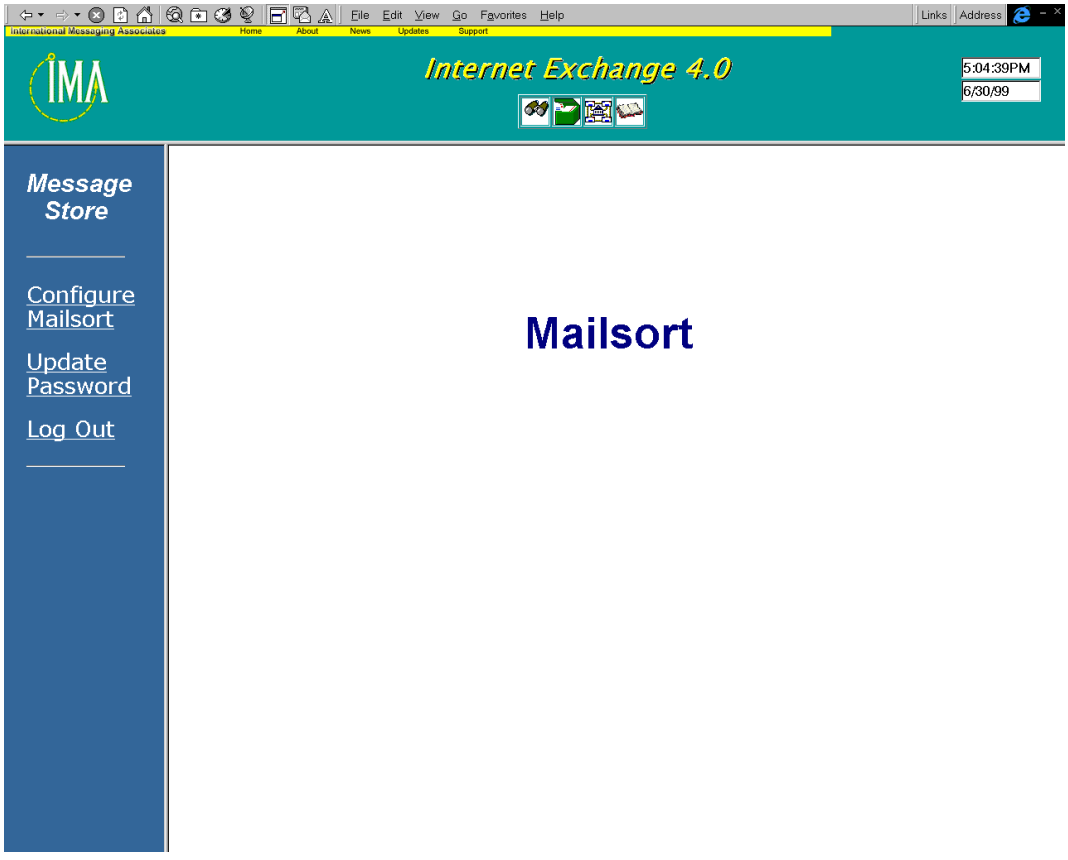

Figure 7f - Configure MailSort/Update Password

To configure MailSort, click on the *Configure MailSort* link. A new page for creating/editing message filters via the MailSort engine will appear (see Figure 7g).

NOTE: If you already have existing filters, the initial MailSort configuration page will display all the existing filters.

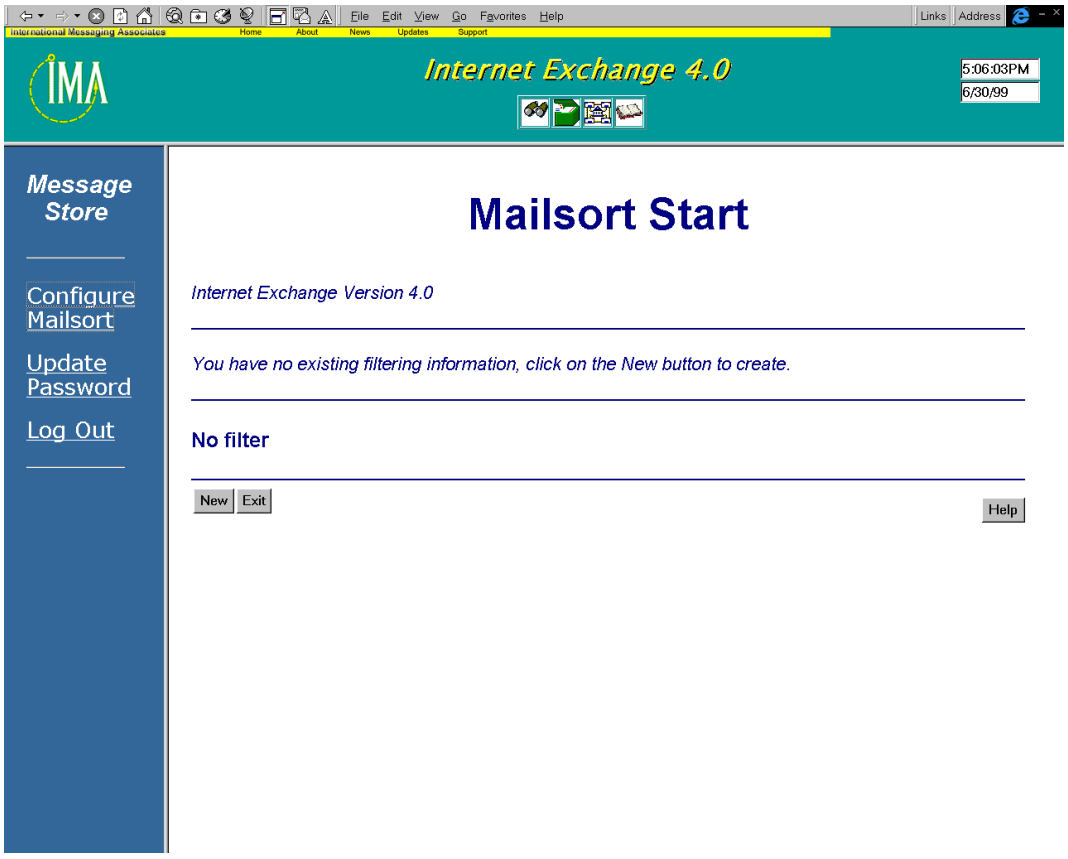

Figure 7g - Create new filter

Creating a filter file

By clicking on the *New* button, another web-based interface for entering the information needed to create a filter file is invoked (see Figure 7h), provided that there is still no filter file that exists for the user.

To create a new filter block, fill up the text boxes with information that will tell the local mail delivery agent where to send a particular message.

For example, a user may want the local mail delivery agent to deliver all messages with a *From:* field containing *John Doe* to be delivered to the *engr* mailbox (which has been created by the user in the Message Store). To do this*:*

1. Select the *From:* header and enter *John Doe* in the opposite text box.

2. Select the option *move to* and choose the *engr* mailbox from the list provided. The user also has options to copy the message to another mailbox or forward it to another email address. An option to send an automatic reply is available (you must have an existing filter file to activate this option). Activating the *reject* option will tell MailSort to reject the message.

3. Select *Yes/No* to configure filtering action.

4. Click the *OK* button to create a new filter block.

5. To create another filter block, repeat the procedure.

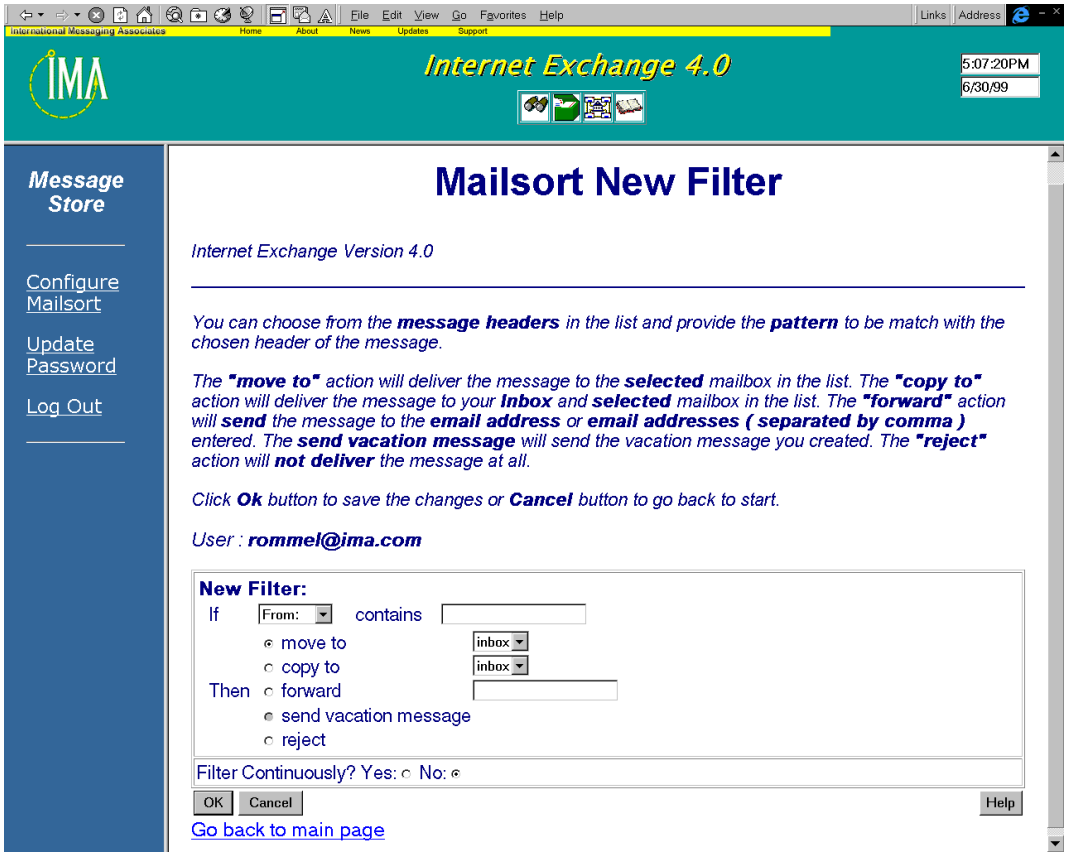

Figure 7h - Enter filter information

Editing an existing filter file

After clicking on the OK button, a new page displaying filter block information will be displayed. Information contained in existing filter blocks can be changed or updated using the *MailSort Filter Information* window. Users with existing filter files are automatically brought to this window upon logging on to MailSort. To display and edit a filter block, click on the *Edit* button for that filter block (see Figure 7i).

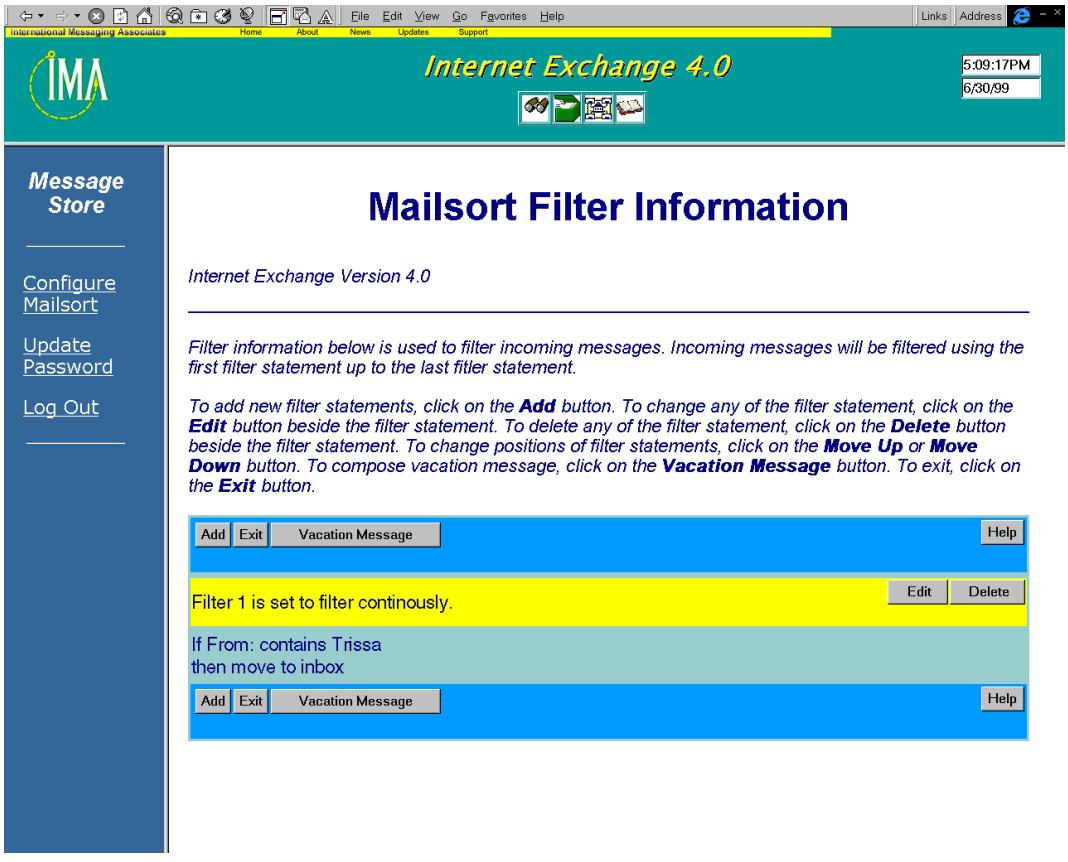

Figure 7i - Display filter blocks

In the Edit page (see Figure 7j), the user can update the one filter data at a time.

1. Select the header field which the Mailsort engine must scan (i.e. *From:, To:, Cc:, Bcc:, Subject:*) to compare the pattern.

2. In the opposite text box, enter the word or phrase that the Mailsort engine must search for in the selected header.

3. Check the action that you want to be taken by the Mailsort engine for messages that meet the defined criteria (i.e. *move to, copy to, forward to, send vacation message, reject*).

4. Select *Yes/No* to configure filtering action.

5. Click on the *OK* button to save the new filter information for that particular filter block.

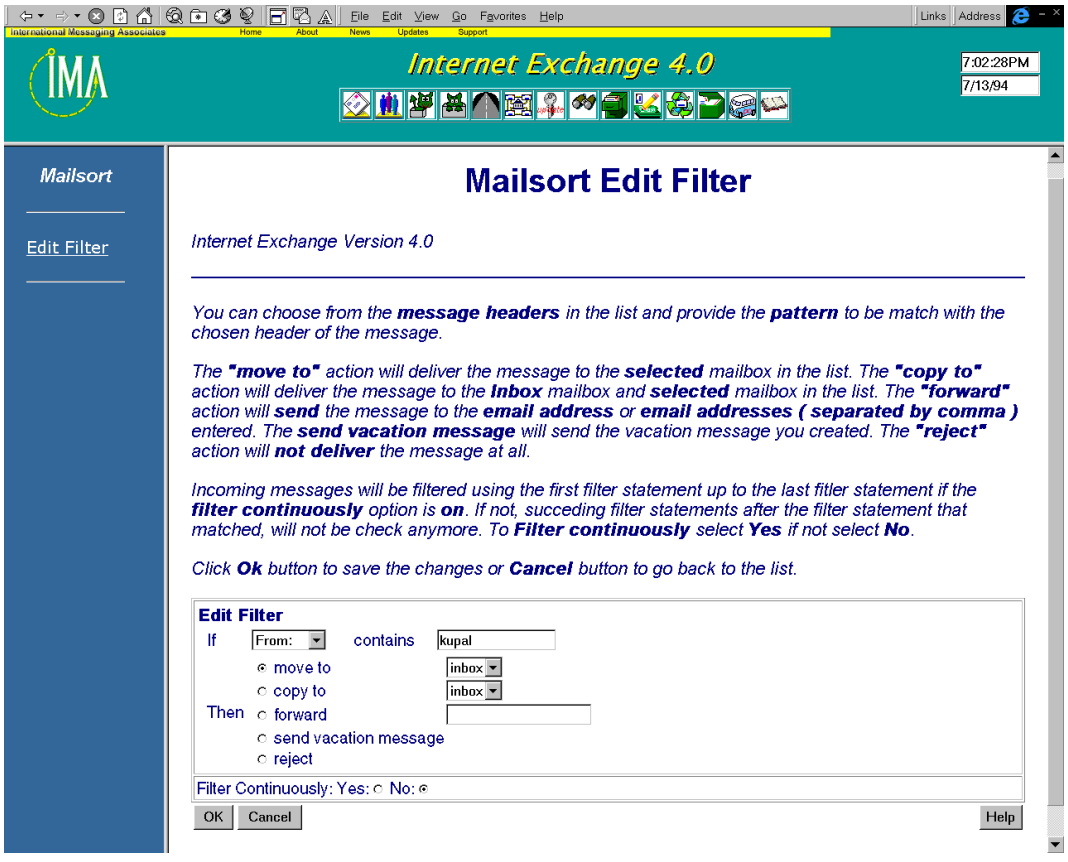

Figure 7j - Edit filter block

Vacation Utility

In the screens for creating/editing filter blocks (Figures 7h and 7j), an option to specify a vacation message is available. This feature is useful when an automatic reply needs to be sent to incoming messages when the *Send Vacation Message* is set in the filter block. Click on the *Vacation Message* button, and the screen shown on (Figure 7k) will be displayed.

The following information needs to be specified for this feature.

Message Subject

Use this field to specify the message subject/header.

Message Body

Use this field to compose the message that needs to be sent out.

NOTE: Vacation messages will not be generated for standard formatted distribution lists. Also, the MailSort Vacation Utility only sends replies to a specific sender every seven days. Thus, if the Vacation Utility has already replied to a sender, it will not send any more messages to that sender until after seven days.

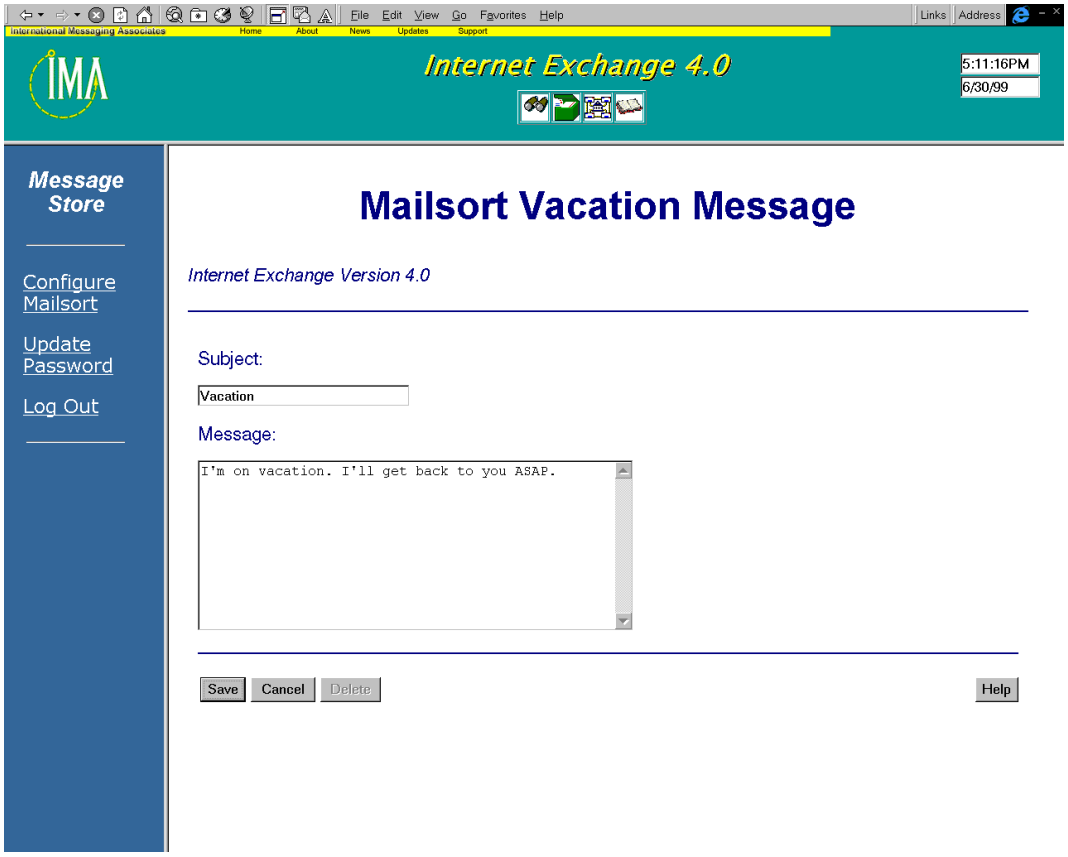

Figure 7k - Create vacation message

Click the *Save* button to save the message. This message will be used when replying to incoming messages.

Click the *Cancel* button to cancel the current message being composed.

Click the *Delete* button to delete the saved message.

Change password

To modify your password, click on the *Update Password* link. A new page will be displayed (see Figure 7l).

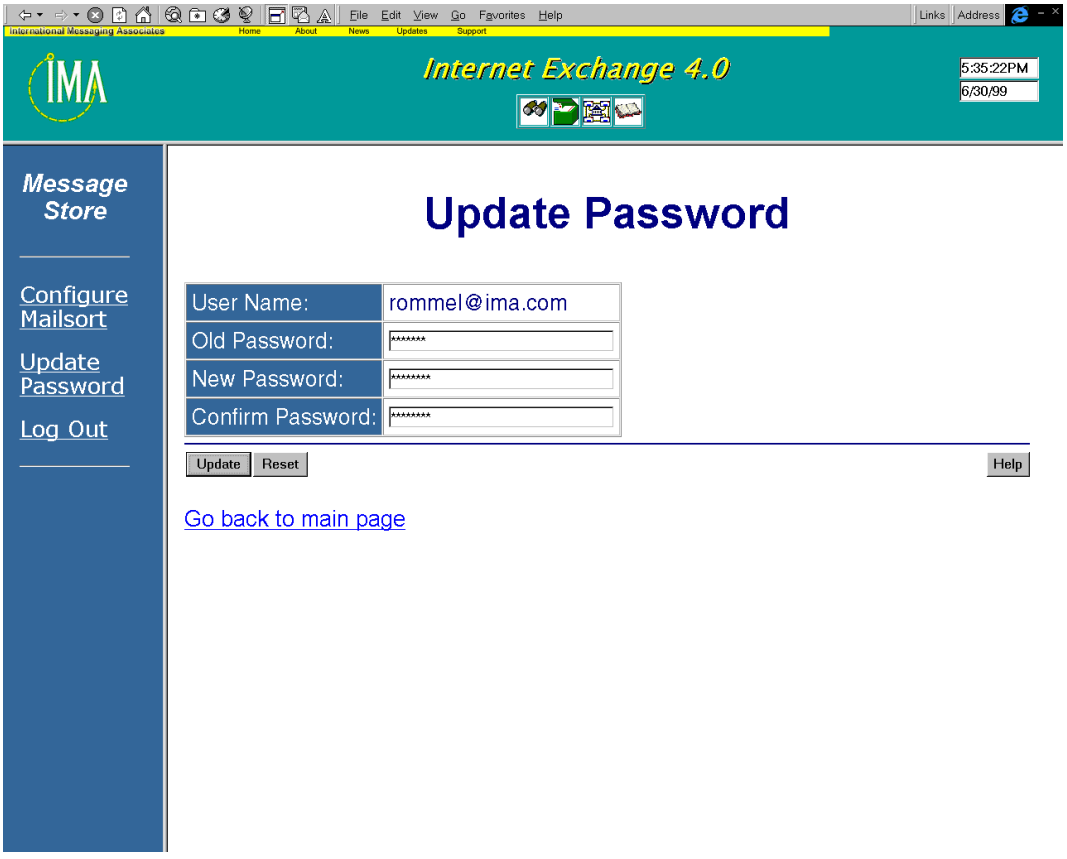

Figure 7l - Change user password

Enter your old password in the textbox provided. Then type your new password and press *Enter*. You will need to re-type your new password in another textbox. Click on the *Update* button to save the new password. For security purposes, the passwords will appear as rows of asterisks.

To log out of the Message Store administration interface, simply click on the *Log Out* link.

DISTRIBUTION LIST MANAGER

Mailing list owners are provided with a Web-based administration interface for configuring the **Internet Exchange 4** Distribution List Manager. To configure the properties of the Distribution List Manager, go back to the authentication page (see Figure 7b). In the *user* and *password* fields, enter the mailing list name and the list owner's password, respectively. For example:

User name: *music@ima.com*
Password: *******

Then click on the *Log On* button. If the user name and password are verified to be correct, the main Web administration interface for end users will be displayed (see Figure 7c). Click on the *Distribution List Owners* link. The following page will be displayed:

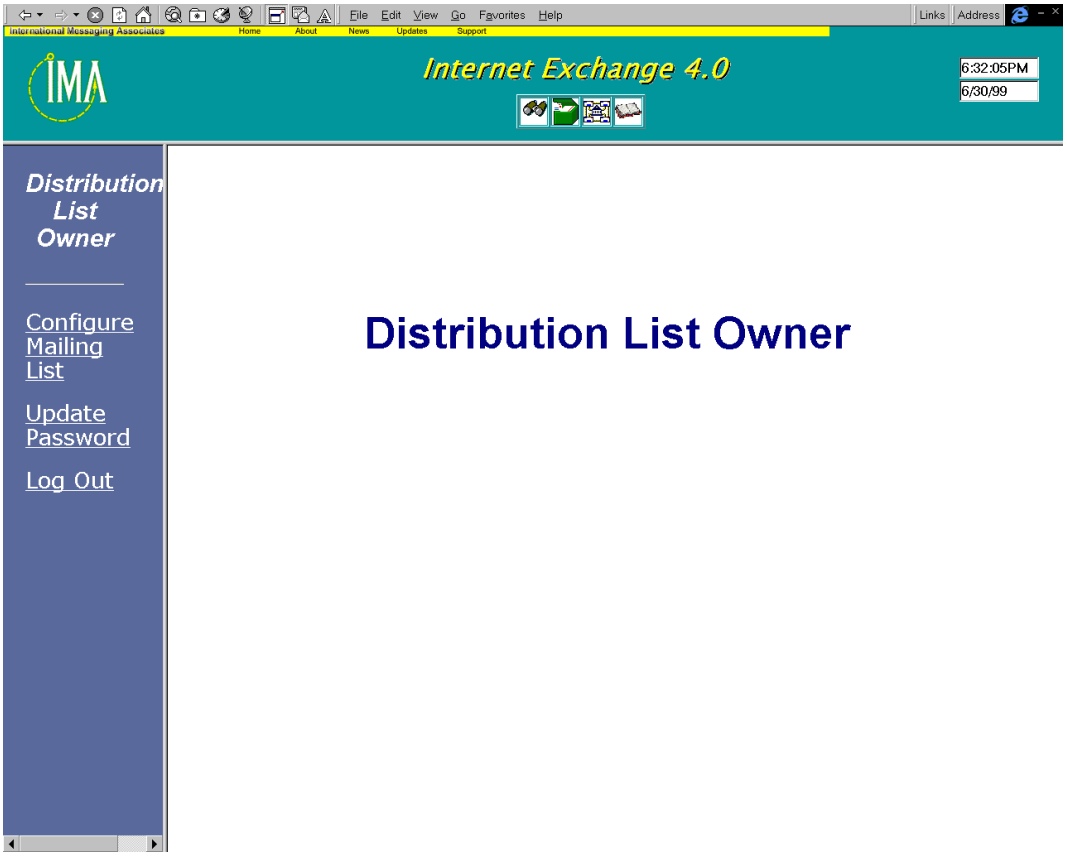

Figure 7m - Main Distribution List Owner Configuration Page

Configure mailing list

To modify an existing mailing list, click on the *Configure Mailing List* link. A new screen for updating/editing the mailing list's attributes will appear (see Figure 7n).

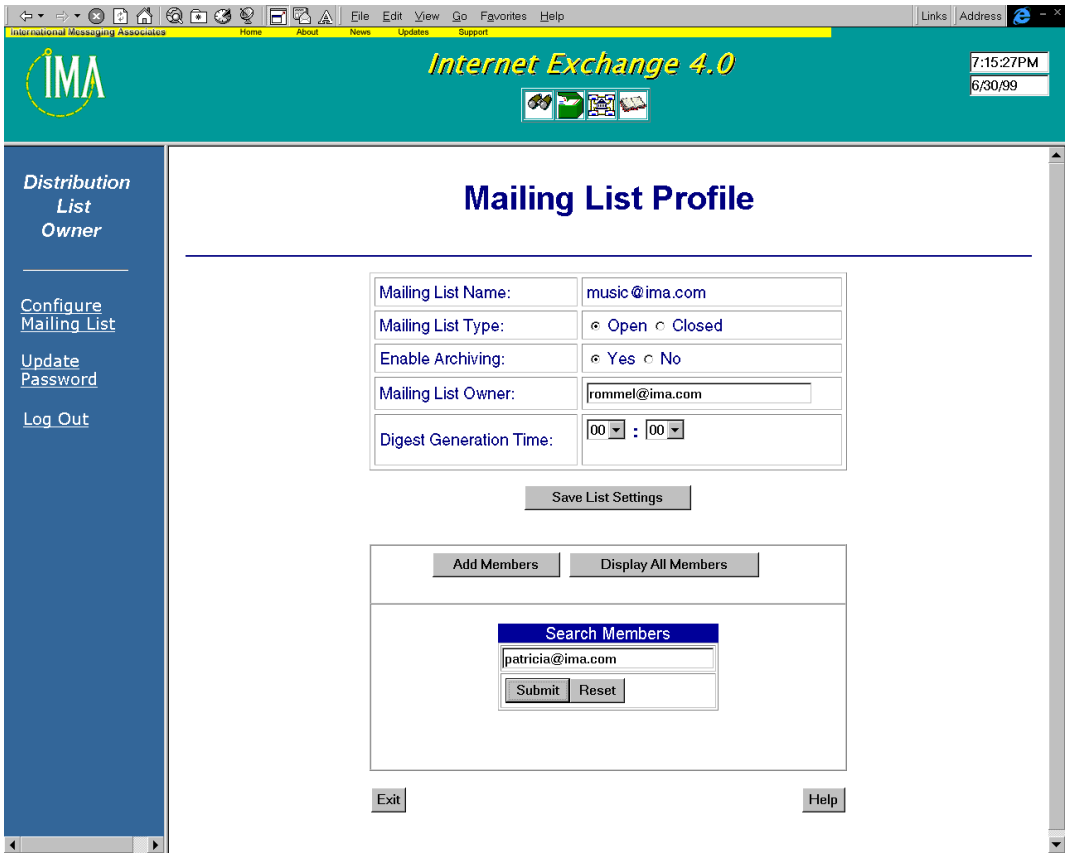

Figure 7n - Mailing list profile

The Mailing List Profile page displays information on the selected distribution list, such as the mailing list name and type, archiving option, list owner, and digest generation time. These attributes (except the mailing list name) can be modified by the distribution list owner.

Add new mailing list member

To add a new member to the mailing list, simply click on the *Add Members* button. The following screen will be displayed:

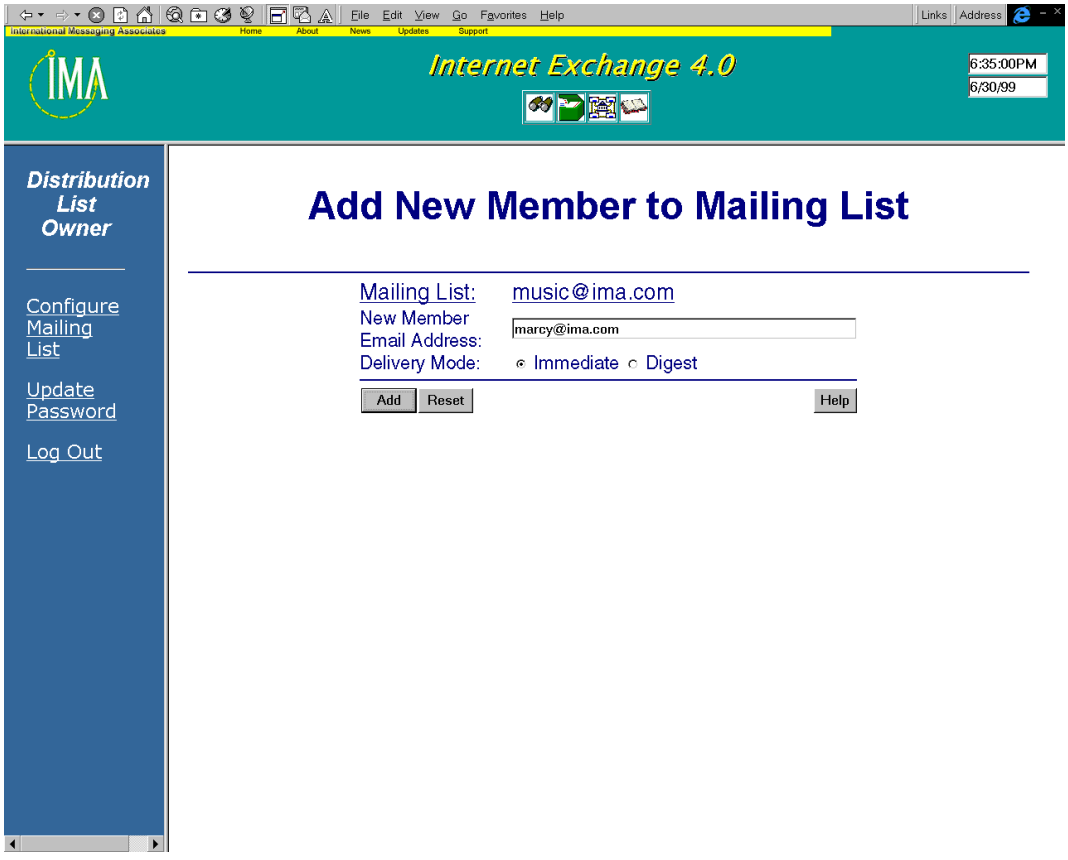

Figure 7o - Add new member

Enter the valid email address of the user to be added to the mailing list and select the preferred delivery mode. Then click on the *Add* button to update the mailing list.

Display mailing list members

To display all the members of the mailing list, click on the *Display All Members* button on the Mailing List Profile page. A new page showing all current list members and the mode of delivery for each member will be displayed (see Figure 7p).

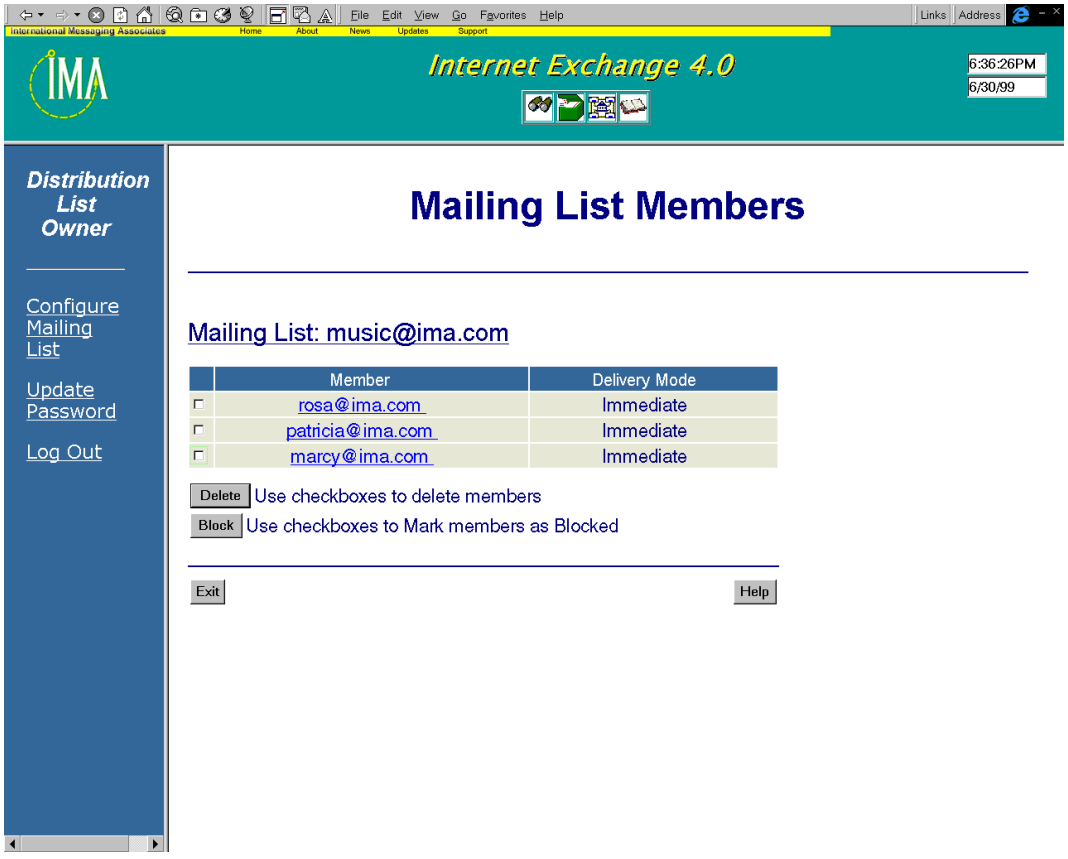

Figure 7p - View mailing list members

Remove mailing list member

To remove a member from the mailing list, check the appropriate checkbox and click on the *Delete* button.

Block mailing list member

To prevent a member from posting messages to the list, check the appropriate checkbox and click on the *Block* button. The member will then be unable to post messages to the list. However, he/she will still be able to receive messages posted by the other members.

Search for a mailing list member

To search for a specific mailing list member, go to the Mailing List Profile page and enter the email address of the member in the *Search Members* textbox and click on the submit button. If the owner of the email address provided is a valid mailing list member, a new screen will be displayed showing the different attributes for that particular member (see Figure 7q).

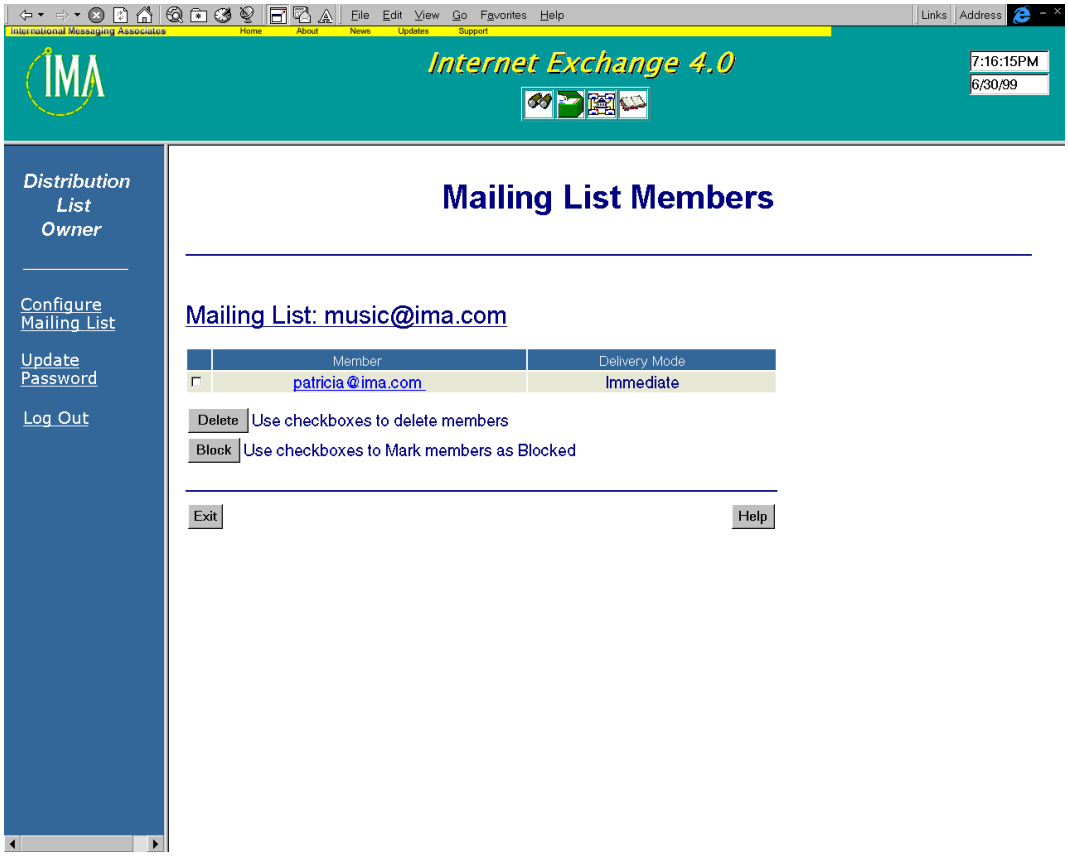

Figure 7q - Search result

Update list owner password

To update the password for the mailing list owner, click on the *Update Password* link. A new screen will be displayed (see Figure 7r).

Enter your old password in the textbox provided. Then type your new password and press *Enter*. You will need to re-type your new password in another textbox. Click on the *Update* button to save the new password. For security purposes, the passwords will appear as rows of asterisks.

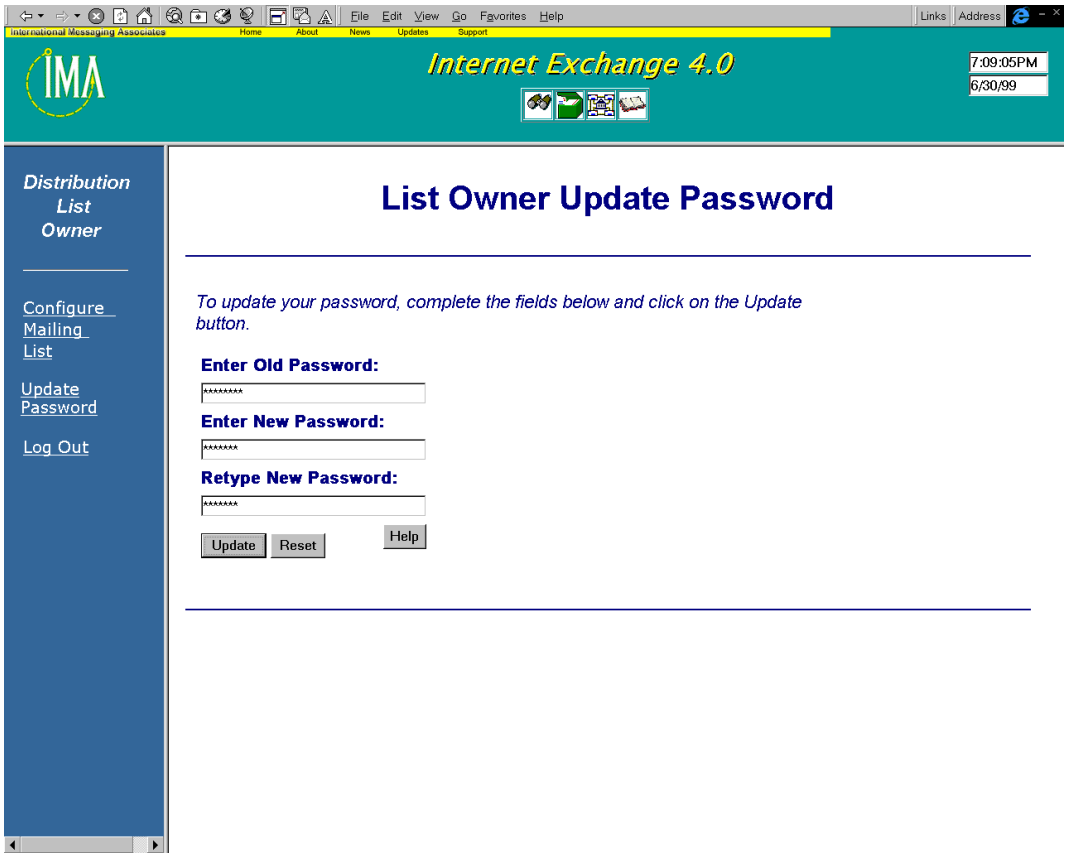

Figure 7r - Update password

Error Handling

ERROR HANDLING FOR THE SMTP DAEMON

The following is a list of errors that may be encountered by the SMTP Daemon and the possible reasons why they occur:

Out of memory
– System has run out of memory.

Network error
– Peer MTA is down.
– Problem in the network path between the Internet Exchange MTA and the peer MTA.
– Misconfiguration of the local TCP stack.

File I/O problem
– Message file is inaccessible.
– File is missing or corrupted.
– File system or hard disk failure.

ERROR HANDLING FOR THE SMTP CLIENT

Out of memory
– System has run out of memory.

Network error
– Peer MTA is down.
– Problem in the network path between the Internet Exchange MTA and the peer MTA.
*– Misconfiguration of the local TCP stac*k.

File I/O problem
– Message file is inaccessible.
– File is missing or corrupted.
– File system or hard disk failure.

Mail Routing Problem
– Misconfiguration of the DNS settings in the name server or host table (if host table is used), the name server is down.

ERROR HANDLING FOR THE POP3/BATCH SMTP MODULE

The following is a list of errors that may be encountered by the POP3/BSMTP Module and the possible reasons why they occur:

Memory allocation failure
– There is not enough memory to allocate.

MQ Initialization Error
– Message Queue was not properly set up.

POP3C Initialization Error
– An error was encountered in attempting to initialize the POP3C thread.

BSMTP Generator Initialization Error
– An error was encountered while attempting to initialize the BSMTP Generator/Encoder thread.

BSMTP Processor Initialization Error
– An error was encountered in attempting to initialize the BSMTP Processor/Decoder thread.

Unable to Open file
– An error was encountered in attempting to open a file.

Unable to Read file
– An error was encountered in attempting to read data from file.

Unable to Write file
– An error was encountered in attempting to write data to a file.

Unable to Create file
– An error was encountered in attempting to create a file.

Unable to Open file
– An error was encountered in attempting to open a file.

Unable to open socket
– Unable to create/initialize a socket.

Unable to connect to POP3 server
– Cannot connect to a specified POP3 server.

Connection Failed
– Socket connection to POP3 server was successful but returned an ERR reply.

Socket error while receiving from server
– Socket error was encountered while receiving data from POP3 server.

Remote side closed connection
– Remote server suddenly closed connection with the POP3 client.

Message Parse Error
– An error was detected while parsing the BSMTP message.

Invalid BSMTP message
– The message passed to the BSMTP processor/decoder is not of application/Batch-SMTP MIME content type.

Unable to create entry in MQ
– An error has occurred while trying to create an entry into the Message Queue.

Unable to insert message to MQ
– An error has occurred while trying to post a message to Message Queue.

Unable to open MQ channel
– An error has occurred while trying to Open an MQ channel.

Unable to fetch message from MQ
– An error has occurred while trying to fetch message from Message Queue.

Error Closing MQ Channel
– An error has occurred while trying to close an MQ channel after inserting a message.

Error Closing MQ Entry
– An error has occurred while trying to Close an MQ entry.

ERROR HANDLING FOR THE ANTI-VIRUS MODULE (PHASE 1)

The following is a list of errors that may be encountered by the Anti-virus Module at Phase 1 and the possible reasons why they occur:

Parameters not valid
– The preprocessor has passed invalid parameters to the Anti-Virus DLL.

Cannot find message file
– The specific message ID file is missing or cannot be access by Anti-Virus DLL.

Cannot find the specific virus scanner
– The virus scanner specified in [AntiV]ProgramPath cannot be found.

Cannot find Sophos SAVI.DLL
– SAVI.DLL cannot be found in the directory specified in [AntiV]ProgramPath.

Unable to initialize Sophos SAVI.DLL
– Anti-Virus plug-in cannot initialize SAVI.DLL.

Unrecognized virus scanner type
– Value of [AntiV]ProgramType does not equal to EXE or DLL.

Message (id) cannot be parsed
– The message file is mal-formatted and the message parser fails to parse the message.

Unable to decode attachment at X
– The X-th attachment of the e-mail message cannot be decoded.

Internal parsing error at X
– The X-th attachment of the e-mail message cannot be decoded

ERROR HANDLING FOR THE ANTI-VIRUS MODULE (PHASE 2)

The following is a list of errors that may be encountered by the Anti-virus Module at Phase 2 and the possible reasons why they occur:

FileOpen failed (proc file)
– Anti-virus plug-in is unable to open the status file.

FileOpen failed (msg file)
– Anti-virus plug-in is unable to open the message file.

FileCreate failed (vir file)
– Anti-virus plug-in is unable to create the .VIR file in the Virus archive.

ERROR HANDLING FOR THE ANTI-SPAM MODULE

The following is a list of errors that may be encountered by the Anti-spam Module and the possible reasons why they occur:

Out of memory
– The system has run out of memory.

Invalid message
– The message is not a valid RFC822 message.
– The header recipients are not valid RFC822 addresses.

File I/O problem
– The message file is inaccessible.
– The file is missing or corrupted.
– File system or hard disk failure.

ERROR HANDLING FOR THE DISTRIBUTION LIST MANAGER

The Distribution List Manager handles the errors returned by the engine and is responsible for notifying the system administrator of such errors. The following is a list of errors that may be encountered by the Distribution List manager engine and the possible reason(s)

why they occur:

Cannot Create an entry in Message Queue
– The engine cannot create an entry in Message Queue.

Recipient's Email Address Unknown format
– The recipient address is invalid.

Sender's Email Address Unknown format
– The sender's email address is invalid.

Unable to Open Backup Database
– The engine is unable to open the backup database. Make sure that the backup database files are stored in the path provided in the RestoreDb section of IEMTA.INI.

Unable to Restore Database
– The engine is unable to find the database files. Make sure that the database files are stored in the path provided in the DBPath section of IEMTA.INI.

Message file is missing
- The engine is unable to find the message file of the fetched message.

Unable to Create a new message. Disk may be full
- The engine is unable to create a new message from the Message Queue. The Disk may be full. When encountered, the Distribution List Manager engine will automatically halt the operation.

The Web-based interface for the Distribution List Manager handles errors by displaying such errors in the browser window, thus notifying the user to perform necessary actions to solve the problem. The following is a list of errors that may be encountered by the Distribution List Manager's Web-based interface and the possible reason(s) why they occur:

List Address not Entered
– The user did not enter the email address of the list to be created.

Entered Email Address not Valid
– The user entered an invalid email address. The address should follow the format name@host.domain.

List Name already Exist
– The user tried to create a mailing list that is exists.

Already a Member of Mailing List
– The user tried to add a new member who is already a member of that particular list.

Could not Connect to Directory Server
– Cannot connect to LDAP Server.

Unable to Delete Mailing List
– The web interface is unable to delete the mailing list.

ERROR HANDLING FOR THE DIRECTORY SERVER

The following is a list of errors that may be encountered by the Directory Server and the possible reasons why they occur:

LDAP_SUCCESS
– Successful LDAP operation.

LDAP_OPERATIONS_ERROR
– Failure on generic operations.

LDAP_PROTOCOL_ERROR
– Failure on protocol-specific operations.

LDAP_TIMELIMIT_EXCEEDED
– Time limit for LDAP operation exceeded.

LDAP_SIZELIMIT_EXCEEDED
– Size limit for LDAP operation exceeded.

LDAP_COMPARE_FALSE
– A comparison operation return false meaning that they are different.

LDAP_COMPARE_TRUE
– A comparison operation return true meaning that they are the same.

LDAP_STRONG_AUTH_NOT_SUPPORTED
– Wrong authentication method value was given.

LDAP_PARTIAL_RESULTS
– Partial results have been retrieved and referral has been imposed.

LDAP_NO_SUCH_ATTRIBUTE
– The attribute given for a LDAP operation was not allowed found on the LDAP directory database.

LDAP_CONSTRAINT_VIOLATION
– Constraint violation. This should not happen.

LDAP_TYPE_OR_VALUE_EXISTS
– If a type or value is already existing.

LDAP_INVALID_SYNTAX
– If the first element has no value.

LDAP_NO_SUCH_OBJECT
– If an object to be searched was not found.

LDAP_ALIAS_DEREF_PROBLEM
– Alias dereferencing problem .

LDAP_INAPPROPRIATE_AUTH
– For simple authentication, If an attribute value for user password is not the same as the root password.

LDAP_INVALID_CREDENTIALS
– For kerberos type of binding, credentials given are not valid.

LDAP_INSUFFICIENT_ACCESS
– Access to certain operations are not allowed.

LDAP_UNWILLING_TO_PERFORM
– Function pointing to the operation is not performing.

LDAP_OBJECT_CLASS_VIOLATION
– Object class violation with regards to the schema used.

LDAP_NOT_ALLOWED_ON_NONLEAF
– LDAP operation is not allowed for non-leaf directory information.

LDAP_ALREADY_EXISTS
– The entry is already existing on the LDAP directory.

LDAP_SERVER_DOWN
– The LDAP server is not running on the specified host.

LDAP_LOCAL_ERROR
– Internal LDAP application error.

LDAP_ENCODING_ERROR
– Encoding of message produces an error. Message could be corrupted or misaligned.

LDAP_DECODING_ERROR
– Decoding of message produces an error. Message could be corrupted or misaligned.

LDAP_TIMEOUT
– The operation timed out.

LDAP_AUTH_UNKNOWN
– Unknown authentication method.

LDAP_FILTER_ERROR

– Bad or wrong search filter.

LDAP_USER_CANCELLED
– User cancelled the operation.

LDAP_PARAM_ERROR
– Bad parameter to an LDAP routine.

LDAP_NO_MEMORY
– Out of memory.

LDAP_URL_ERR_NOTLDAP
– URL doesn't begin with "ldap://".

LDAP_URL_ERR_NODN
– URL has no DN (required).

LDAP_URL_ERR_BADSCOPE
– URL scope string is invalid.

LDAP_URL_ERR_MEM
– can't allocate memory space for LDAP URL.

PART 4

Troubleshooting

Troubleshooting Tools

TROUBLESHOOTING THE SMTP DAEMON

Network problem

Network problems could be due to errors encountered in the routing path between Internet Exchange MTA and the peer MTA, or other network devices (i.e. routers, bridges, etc.) in the routing path. The network diagnostic tool "*ping*" is helpful in investigating various network problems. It can manually perform "*telneting to the port 25*" to the machine running Internet Exchange MTA to help diagnose the problem.

TROUBLESHOOTING THE SMTP CLIENT

Mail routing problem

It is possible that message delivery to Internet failed due to mail routing problems. Such problems can be caused by misconfiguration of DNS MX settings in the name server or host table (if host table is used), or by the failure of the name server. In this case, the DNS diagnostic tool *"nslookup"* can be used to investigate the problem.

Network problem

Network problems could be due errors in the routing path between Internet Exchange MTA and the peer MTA, the unavailability of the peer MTA, or other network devices (i.e. routers, bridges, etc.) in the routing path. The network diagnostic tool *"ping"* can help in investigating various network problems. It can manually perform "*telneting to the port 25*" to the machine running Internet Exchange MTA to help diagnose the problem.

TROUBLESHOOTING THE POP3/BSMTP

The BSMTP module may produce five kinds of errors: POP3C error, Encoder error, Decoder Error, Initialization error and MTA error. These errors, together with detailed explanations, will be written to the log file.

POP3C errors

– Errors which are encountered while connecting and/or fetching messages from a POP3 server.

Encoder errors

– These errors that may occur if the encoder found an error while creating a batch-SMTP type of message (for example, it may encounter *out of disk space* error while creating a message file).

Decoder errors

- These may occur if the decoder encountered an error while parsing a batch-SMTP message.

Initialization Errors

– These error may occur if any of the BSMTP's components failed during start-up

MTA Errors

– These error may occur if BSMTP failed to call any of the MTA API's (for example, while connecting to the LDAP server or fetching a message from the Message Queue).

TROUBLESHOOTING THE ANTI-VIRUS MODULE

The Anti-Virus plug-in logs error messages in the IEMS log file when necessary. To view the complete list of error messages and their meanings, see Chapter 7 – Error Handling.

TROUBLESHOOTING THE AUTO TEXT INSERTION ENGINE

If the logfile indicates that there is error while writing to file:

- Check if there is enough diskspace in the Internet Exchange temporary directory.
- Use disk diagnostic tool like scan disk to check if there are any problems in the file system or the hard drive is physically damaged.
- If the logfile indicates that the insertion engine is not able to read the simple plain text/or html text file.
- Check if the said files are readable/exists in the system.
- If the files are stored in a network share, make sure that the system has the proper read permission to that network share.

TROUBLESHOOTING THE ANTI-SPAM MODULE

The system administrator can check the log files whenever errors occur. The log files contain the last actions performed by the system before a particular error occurred. Through these files, the system administrator can trace which module or process encountered an error.

Log files are in text mode. Ordinary text editors such as Notepad or Wordpad can open these log files.

TROUBLESHOOTING THE DISTRIBUTION LIST MANAGER

The system administrator can check the log files whenever errors occur. The log files contain the last actions performed by the system before a particular error occurred. Through these files, the system administrator can trace which module or process encountered an error.

Log files are in text mode. Ordinary text editors such as Notepad or Wordpad can open these log files.

TROUBLESHOOTING THE DIRECTORY SERVER

The administrator can troubleshoot this application by studying the sequence of messages or errors written by the LDAP server using the LOGAPI library. LOGAPI is a library that displays different kinds of messages for different Internet Exchange version 4.0 applications. The LDAP server is just one of the application that uses this API.

PART 5

Appendices

Key Technologies

Lightweight Directory Access Protocol (LDAP)

As computer networking increasingly became popular in the 1980s, it became apparent that a global electronic directory service (EDS) was needed to optimize the potential of distributed computer systems. To achieve this goal, the International Standards Organization (ISO) and the International Telegraph collaborated with the Telephone Consultative Committee (CCITT) to develop the X.500 directory standard within the Open Systems Interconnection (OSI) model in 1988. From this standard evolved the Lightweight Directory Access Protocol (LDAP), which was developed at the University of Michigan.

The X.500 directory standard was developed to provide the networking community with an online version of the "white pages" or "yellow pages." Such online directories are very useful, particularly to Internet and intranet users. They can be used to search for a person or an organization's e-mail address or other important information (i.e. organization names, department names, telephone numbers, etc.). On the Internet, the Domain Name System serves as the directory for relating a domain name to a particular network (IP) address. For a user to send email via the Internet, he must first know the email address of the recipient. But what if he does not know the email address of the intended recipient? This is just one of the issues that the X.500 directory standard seeks to address by providing a directory that contains a wide range of information, such as peoples' names, company names, telephone numbers, and postal codes.

The X.500 EDS is based on a client-server architecture (see Figure 1). It consists of three major components: the Directory System Agent (DSA), the Directory User Agent (DUA), and the Directory Information Base (DIB). The DSA resides on the server computer and manages information in the directory, while the DUA is an application software component that allows the client computer to access the directory. The DIB serves as the repository for information.

Aside from these three key components, the X.500 EDS also uses several protocols for managing information. The Directory Access Protocol (DAP) facilitates communication between DUA's and DSA's, while the Directory System Protocol (DSP) supports communication among DSA's for distributed directory operations. The Directory Information Shadowing Protocol (DISP), on the other hand, allows two or more DSA's to share copies of their DIBs for replication purposes. Another protocol, the Directory Operations Protocol (DOP), is also used for replication purposes. However, the DOP is seldom used in X.500 EDS's. The directory is distributed among several DSA's, and if one of the DSA's fails to answer a client's request, the client is referred to another DSA, a process called chaining.

In an X.500 EDS, information is stored using an organizational tree-like structure. This is called the Directory Information Tree (DIT).

The DAP allows communication between the DUA and the DSA. It provides a wide range of directory services via such commands as *Read, Compare, Abandon, Add Entry, Remove Entry, Modify Entry, Modify RDN, Search,* and *List*.

However, DAP is a complete OSI application, meaning it requires all layers of the 7-layer OSI protocol stack. Plus it has too much code and requires excessive computing horsepower to run. These factors place an unreasonable burden on client machines that are usually not designed to support OSI protocol stack, thereby, posing a limitation to intranet and Internet users.

To provide such machines with directory-access capability, the University of Michigan developed the Lightweight Directory Access Protocol (LDAP), a simplified version of the DAP. Since then, the protocol has gone through several revisions. RFC 1777 defines LDAP version 2 as a protocol for providing read/write access to X.500 directories using less processing requirements than the DAP. LDAP2 supports simple authentication using a cleartext password as well as the Kerberos version of authentication. It can also run over a Secure Socket Layer/Transport Layer Security (SSL/TLS) transport layer.

The latest of LDAP is version 3 (LDAP3). As defined in RFC 2251, LDAP3 is a protocol designed to provide access to open X.500 directory service and proprietary directories that support the X.500 standard without incurring the resource requirements of the DAP. It is specifically targeted at management and browser applications that provide read/write interactive access to directories. Like LDAP2, LDAP3 is designed to complement the DAP when used with a directory that supports the X.500 protocols. LDAP3 supports all the security features found in LDAP2. In addition, it supports the SASL, which allows an extensible authentication and security framework. LDAP3 is also designed to return referrals to other servers to clients.

LDAP does not require the upper layers of the OSI protocol stack and runs directly on TCP/IP or other reliable transport protocols. Moreover, it uses a smaller amount of code than the DAP and requires minimal processing overhead. LDAP uses simple character strings to encode DN's and data elements (see RFC 2252, RFC 2253, and RFC 2254), as compared to the X.500, which uses highly-structured encoding even for simple data elements. This makes it easier to decode large DN's. The protocol also eliminates the need for the read and list operations, emulating them by means of the search operation.

To gain access to X.500 directories, an LDAP server must be able to support both TCP/IP and OSI protocols. This type of server answers the needs of the client by becoming a client to the X.500 server (see Figure 3). LDAP servers designed to access proprietary directories are not required to support the OSI protocols. Such servers are known as stand-alone LDAP servers since they do not rely on X.500 servers.

Simple Mail Transfer Protocol Client (SMTPC)

SMTPC is responsible for the delivery of messages on the Internet. This is carried out by SMTPC by regularly checking for messages queued in the SMTP OUT queue. When messages are found, SMTPC establishes the required number of connections with external SMTP servers and transfers the messages to the appropriate Internet mail hosts.

Simple Mail Transfer Protocol Daemon (SMTPD)

SMTPD is the module responsible for receiving messages from the Internet. It is a server process that continuously runs on the machine. This is necessary since it is impossible to predict the timing or frequency of inbound messages. When SMTPD receives a message, the message is placed in the

SMTP IN queue. Unlike *SMTPC*, *SMTPD* does not perform any message translation. It simply creates the queue entry and goes back to wait for additional connection requests.

Request for Comments (RFC's)

Request for Comments: 1487
X.500 Lightweight Directory Access Protocol

The protocol described in this document is designed to provide access to the Directory while not incurring the resource requirements of the Directory Access Protocol (DAP). This protocol is specifically targeted at simple management applications and browser applications that provide simple read/write interactive access to the Directory, and is intended to be a complement to the DAP itself. Key aspects of LDAP are:

- Protocol elements are carried directly over TCP or other transport, bypassing much of the session/presentation overhead.
- Many protocol data elements are encoding as ordinary strings (e.g., Distinguished Names).
- A lightweight BER encoding is used to encode all protocol elements.

The general model adopted by this protocol is one of clients performing protocol operations against servers. In this model, this is accomplished by a client transmitting a protocol request describing the operation to be performed to a server, which is then responsible for performing the necessary operations on the Directory.

Upon completion of the necessary operations, the server returns a response containing any results or errors to the requesting client. In keeping with the goal of easing the costs associated with use of the Directory, it is an objective of this protocol to minimize the complexity of clients so as to facilitate widespread deployment ofapplications capable of utilizing the Directory. Note that, although servers are required to return responses whenever such responses are defined in the protocol, there is no requirement for synchronous behavior on the part of either client or server implementations: requests and responses for multiple operations may be exchanged by client and servers in any order, as long as clients eventually receive a response for every request that requires one.

Consistent with the model of servers performing protocol operations on behalf of clients, it is also to be noted that protocol servers are expected to handle referrals without resorting to the return of such referrals to the client. This protocol makes no provisions for the return of referrals to clients, as the model is one of servers ensuring the performance of all necessary operations in the Directory, with only final results or errors being returned by servers to clients. Note that this protocol can be mapped to a strict subset of the directory abstract service, so it can be cleanly provided by the DAP.

Mapping Onto Transport Services

This protocol is designed to run over connection-oriented, reliable transports, with all 8 bits in an octet being significant in the data stream. Specifications for two underlying services are defined here, though others are also possible.

- Transmission Control Protocol (TCP)

The LDAPMessage PDUs are mapped directly onto the TCP bytestream. Server

implementations running over the TCP should provide a protocol listener on port 389.

• Connection Oriented Transport Service (COTS)

The connection is established. No special use of T-Connect is made. Each LDAPMessage PDU is mapped directly onto T-Data.

Elements of Protocol

For the purposes of protocol exchanges, all protocol operations are encapsulated in a common envelope, the LDAPMessage, which is defined as follows:

```
LDAPMessage ::=
 SEQUENCE {
     messageID      MessageID,
      protocolOp    CHOICE {
                bindRequest         BindRequest,
                bindResponse        BindResponse,
                unbindRequest       UnbindRequest
                searchRequest       SearchRequest,
                searchResponse      SearchResponse,
                modifyRequest       ModifyRequest,
                modifyResponse      ModifyResponse,
                addRequest          AddRequest,
                addResponse         AddResponse,
                delRequest          DelRequest,
                delResponse         DelResponse,
                modifyRDNRequestModifyRDNRequest,
                modifyRDNResponse ModifyRDNResponse,
                compareDNRequest CompareRequest,
                compareDNResponseCompareResponse,
                abandonRequest      AbandonRequest
     }
 }
 MessageID ::= INTEGER (0 .. MaxInt)
```

The function of the LDAPMessage is to provide an envelope containing common fields required in all protocol exchanges. At this time the only common field is a message ID, which is required to have a value different from the values of any other requests outstanding in the LDAP session of which this message is a part. The message ID value must be echoed in all LDAPMessage envelopes encapsulating responses corresponding to the request contained in the LDAPMessage in which the message ID value was originally used.

Request for Comments: 1521
MIME (Multipurpose Internet Mail Extensions)
Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies

STD 11, RFC 822 defines a message representation protocol which specifies considerable detail about message headers, but which leaves the message content, or message body, as flat ASCII text. This document redefines the format of message bodies to allow multi-part textual and non-textual message bodies to be represented and exchanged without loss of information. This is based on earlier work documented in RFC 934 and STD 11, RFC 1049, but extends and revises that work. Because RFC 822 said so little about message bodies, this document is largely orthogonal to (rather than a revision of) RFC 822.

In particular, this document is designed to provide facilities to include multiple objects in a single message, to represent body text in character sets other than US-ASCII, to represent formatted multi-font text messages, to represent non-textual material such as images and audio fragments, and generally to facilitate later extensions defining new types of Internet mail for use by cooperating mail agents.

This document does NOT extend Internet mail header fields to permit anything other than US-ASCII text data. Such extensions are the subject of a companion document RFC-1522.

Since its publication in 1982, STD 11, RFC 822 has defined the standard format of textual mail messages on the Internet. Its success has been such that the RFC 822 format has been adopted, wholly or partially, well beyond the confines of the Internet and the Internet SMTP transport defined by STD 10, RFC 821. As the format has seen wider use, a number of limitations have proven increasingly restrictive for the user community.

RFC 822 was intended to specify a format for text messages. As such, non-text messages, such as multimedia messages that might include audio or images, are simply not mentioned. Even in the case of text, however, RFC 822 is inadequate for the needs of mail users whose languages require the use of character sets richer than US ASCII. Since RFC 822 does not specify mechanisms for mail containing audio, video, Asian language text, or even text in most European languages, additional specifications are needed.

One of the notable limitations of RFC 821/822 based mail systems is the fact that they limit the contents of electronic mail messages to relatively short lines of seven-bit ASCII. This forces users to convert any non-textual data that they may wish to send into seven- bit bytes representable as printable ASCII characters before invoking a local mail UA (User Agent, a program with which human users send and receive mail). Examples of such encodings currently used in the Internet include pure hexadecimal, uuencode, the 3-in-4 base 64 scheme specified in RFC 1421, the Andrew Toolkit Representation, and many others.

The limitations of RFC 822 mail become even more apparent as gateways are designed to allow for the exchange of mail messages between RFC 822 hosts and X.400 hosts. X.400 specifies mechanisms for the inclusion of non-textual body parts within electronic mail messages. The current standards for the mapping of X.400 messages to RFC 822 messages specify either that X.400 non-textual body parts must be converted to (not encoded in) an ASCII format, or that they must

be discarded, notifying the RFC 822 user that discarding has occurred. This is clearly undesirable, as information that a user may wish to receive is lost. Even though a user's UA may not have the capability of dealing with the non-textual body part, the user might have some mechanism external to the UA that can extract useful information from the body part. Moreover, it does not allow for the fact that the message may eventually be gatewayed back into an X.400 message handling system (i.e., the X.400 message is "tunneled" through Internet mail), where the non-textual information would definitely become useful again.

This document describes several mechanisms that combine to solve most of these problems without introducing any serious incompatibilities with the existing world of RFC 822 mail. In particular, it describes:

- A MIME-Version header field,

  which uses a version number to declare a message to be conformant with this specification and allows mail processing agents to distinguish between such messages and those generated by older or non-conformant software, which is presumed to lack such a field.

- A Content-Type header field,

  generalized from RFC 1049, which can be used to specify the type and subtype of data in the body of a message and to fully specify the native representation (encoding) of such data.

  - A *text* Content-Type value, which can be used to represent textual information in a number of character sets and formatted text description languages in a standardized manner.
  - A *multipart* Content-Type value, which can be used to combine several body parts, possibly of differing types of data, into a single message.
  - An *application* Content-Type value, which can be used to transmit application data or binary data, and hence, among other uses, to implement an electronic mail file transfer service.
  - A *message* Content-Type value, for encapsulating another mail message.
  - An *image* Content-Type value, for transmitting still image (picture) data.
  - An *audio* Content-Type value, for transmitting audio or voice data.
  - A *video* Content-Type value, for transmitting video or moving image data, possibly with audio as part of the composite video data format.

- A Content-Transfer-Encoding header field,

  which can be used to specify an auxiliary encoding that was applied to the data in order to allow it to pass through mail transport mechanisms which may have data or character set limitations.

- The Content-ID and Content-Description header fields

  Two additional header fields that can be used to further describe the data in a message body.

MIME has been carefully designed as an extensible mechanism, and it is expected that the set of content-type/subtype pairs and their associated parameters will grow significantly with time. Several other MIME fields, notably including character set names, are likely to have new values defined over time. In order to ensure that the set of such values is developed in an orderly, well-

specified, and public manner, MIME defines a registration process which uses the Internet Assigned Numbers Authority (IANA) as a central registry for such values.

Request for Comments: 1522
MIME (Multipurpose Internet Mail Extensions)
Part Two: Message Header Extensions for
Non-ASCII Text

This memo describes an extension to the message format defined in RFC 1521, to allow the representation of character sets other than ASCII in RFC 822 message headers. The extensions described were designed to be highly compatible with existing Internet mail handling software, and to be easily implemented in mail readers that support RFC 1521.

RFC 1521 describes a mechanism for denoting textual body parts which are coded in various character sets, as well as methods for encoding such body parts as sequences of printable ASCII characters. This memo describes similar techniques to allow the encoding of non-ASCII text in various portions of a RFC 822 message header, in a manner which is unlikely to confuse existing message handling software.

Like the encoding techniques described in RFC 1521, the techniques outlined here were designed to allow the use of non-ASCII characters in message headers in a way which is unlikely to be disturbed by the quirks of existing Internet mail handling programs. In particular, some mail relaying programs are known to (a) delete some message header fields while retaining others, (b) rearrange the order of addresses in To or Cc fields, (c) rearrange the (vertical) order of header fields, and/or (d) "wrap" message headers at different places than those in the original message. In addition, some mail reading programs are known to have difficulty correctly parsing message headers which, while legal according to RFC 822, make use of backslash-quoting to "hide" special characters such as "<", ",", or ":", or which exploit other infrequently-used features of that specification.

Request for Comments: 1558
A String Representation of LDAP Search Filters

The Lightweight Directory Access Protocol (LDAP) defines a network representation of a search filter transmitted to an LDAP server. Some applications may find it useful to have a common way of representing these search filters in a human-readable form. This document defines a human-readable string format for representing LDAP search filters.

LDAP Search Filter Definition

An LDAP search filter is defined in [1] as follows:

```
Filter ::= CHOICE {
    and             [0] SET OF Filter,
    or              [1] SET OF Filter,
    not             [2] Filter,
    equalityMatch   [3] AttributeValueAssertion,
    substrings      [4] SubstringFilter,
```

```
        greaterOrEqual  [5] AttributeValueAssertion,
        lessOrEqual     [6] AttributeValueAssertion,
        present         [7] AttributeType,
        approxMatch     [8] AttributeValueAssertion
    }
    SubstringFilter ::= SEQUENCE {
        type    AttributeType,
        SEQUENCE OF CHOICE {
            initial         [0] LDAPString,
            any             [1] LDAPString,
            final           [2] LDAPString
        }
    }
    AttributeValueAssertion ::= SEQUENCE
        attributeType   AttributeType,
        attributeValue AttributeValue
    }
    AttributeType ::= LDAPString
    AttributeValue ::= OCTET STRING
    LDAPString ::= OCTET STRING
```

where the LDAPString above is limited to the IA5 character set. The AttributeType is a string representation of the attribute object identifier in dotted OID format (e.g., "2.5.4.10"), or the shorter string name of the attribute (e.g., "organizationName", or "o").

String Search Filter Definition

The string representation of an LDAP search filter is defined by the following BNF. It uses a prefix format.

```
<filter> ::= '(' <filtercomp> ')'
<filtercomp> ::= <and> | <or> | <not> | <item>
<and> ::= '&' <filterlist>
<or> ::= '|' <filterlist>
<not> ::= '!' <filter>
<filterlist> ::= <filter> | <filter> <filterlist>
<item> ::= <simple> | <present> | <substring>
<simple> ::= <attr> <filtertype> <value>
<filtertype> ::= <equal> | <approx> | <greater> | <less>
<equal> ::= '='
<approx> ::= '~='
<greater> ::= '>='
<less> ::= '<='
<present> ::= <attr> '=*'
<substring> ::= <attr> '=' <initial> <any> <final>
<initial> ::= NULL | <value>
<any> ::= '*' <starval>
<starval> ::= NULL | <value> '*' <starval>
<final> ::= NULL | <value>
```

Request for Comments: 1740
MIME Encapsulation of Macintosh Files - MacMIME

This memo describes the format to use when sending Apple Macintosh files via MIME. The format is compatible with existing mechanisms for distributing Macintosh files, while allowing non-Macintosh systems access to data in standardized formats.

Files on the Macintosh consists of two parts, called forks:

- DATA FORK

  The actual data included in the file. The Data fork is typically the only meaningful part of a Macintosh file on a non-Macintosh computer system. For example, if a Macintosh user wants to send a file of data to a user on an IBM-PC, she would only send the Data fork.

- RESOURCE FORK

  Contains a collection of arbitrary attribute/value pairs, including program segments, icon bitmaps, and parametric values.

Additional information regarding Macintosh files is stored by the Finder in a hidden file, called the "Desktop Database". Because of the complications in storing different parts of a Macintosh file in a non-Macintosh filesystem that only handles consecutive data in one part, it is common to convert the Macintosh file into some other format before transferring it over the network.

The two styles of use are:

- AppleSingle

  Apple's standard format for encoding Macintosh files as one byte stream.

- AppleDouble

  Similar to AppleSingle except that the Data fork is separated from the Macintosh-specific parts by the AppleDouble encoding.

AppleDouble is the preferred format for a Macintosh file that is to be included in an Internet mail message, because it provides recipients with Macintosh computers the entire document, including Icons and other Macintosh specific information, while other users easily can extract the Data fork (the actual data) as it is separated from the AppleDouble encoding.

MIME format for Apple/Macintosh-specific file information

- APPLICATION/APPLEFILE

MIME type-name:	APPLICATION
MIME subtype name:	APPLEFILE
Required parameters:	none
Optional parameters:	NAME, which must be a "value" as defined in RFC-1521.
Encoding considerations:	The presence of binary data will typically require use of Content-Transfer-Encoding: BASE64

Security considerations: See separate section in the document
Published specification: Apple-single & Apple-double
Rationale: Permits MIME-based transmission of data with Apple/Macintosh specificinformation, while allowing general access to non-specific user data.

• MULTIPART/APPLEDOUBLE

MIME type-name: MULTIPART
MIME subtype name: APPLEDOUBLE
Required parameters: none
Optional parameters: NAME, which must be a "value" as defined in RFC-1521.
Encoding considerations: none
Security considerations: See separate section in the document
Published specification: Apple-single & Apple-double
Rationale: Permits MIME-based transmission of data with Apple/Macintosh specific information, while allowing general access to non-specific user data.

• Detail specific to MIME-based usage

Macintosh documents do not always need to be sent in a special format. Those documents with well-known MIME types and non-existent or trivial resource forks can be sent as regular MIME body parts, without use of AppleSingle or AppleDouble. Documents which lack a data fork must be sent as AppleSingle. Unless there are strong reasons not to, all other documents should normally be sent as AppleDouble. This includes documents with non-trivial resource forks, and documents without corresponding well-known MIME types. It may be valuable in some cases to allow the user to choose one format over another, either because he disagrees with the implementor's definition of "trivial" resource forks, or for reasons of his own.

AppleSingle

An AppleSingle, version 2 file, is sent as one consecutive stream of bytes. The format is described in [APPL90] with a brief summary in Appendix A. The one and only part of the file is sent in an application/applefile message.

The first four bytes of an AppleSingle header are, in hexadecimal:
00, 05, 16, 00.

The AppleSingle file is binary data. Hence, it may be necessary to perform a Content-Transfer-Encoding for transmission, depending on the underlying email transport environment. The safest encoding is Base64, since it permits transfer over the most restricted channels. Even though an AppleSingle file includes the original Macintosh filename, it is recommended that a name parameter be included on the Content-Type header to give the recipient a hint as to what file is attached. The value of the name parameter must be a "value" as defined by RFC-1521. Note that this restricts the value to seven-bit US-ASCII characters.

AppleDouble

An AppleDouble, version 2, file is divided in two parts:
• Header

including the Macintosh resource fork and desktop information and

- Data fork

  containing the Macintosh data fork.

The AppleDouble file itself is sent as a multipart/appledouble MIME body-part, which may have only two sub-parts. The header is sent as application/applefile and the data fork as whatever best describes it. For example, is the data for is actually a GIF image, it should be sent as image/gif. If no appropriate Content-Type has been registered for the data type, it should be sent as an application/octet-stream.

The first four bytes of an AppleDouble header are, in hexadecimal:
00, 05, 16, 07.

The AppleDouble header is binary data. Hence, it may be necessary to perform a Content-Transfer-Encoding for transmission, depending on the underlying email transport environment. The safest encoding is Base64, since it permits transfer over the most restrictive channels. Even though an AppleDouble file includes the original Macintosh filename, it is recommended that a name parameter be included on the Content-Type header of both the header and data parts of the AppleDouble file to give the recipient a hint as to what file is attached. The value of the name parameter must be a "value" as defined by RFC-1521. Note that this restricts the value to seven-bit US-ASCII characters.

Request for Comments: 1741
MIME Content Type for BinHex Encoded Files

This memo describes the format to use when sending BinHex4.0 files via MIME. The format is compatible with existing mechanisms for distributing Macintosh files. Only when available software and/or user practice dictates, should this method be employed. It is recommended to use application/applefile for maximum interoperability.

Files on the Macintosh consists of two parts, called forks:

- DATA FORK

  The actual data included in the file. The Data fork is typically the only meaningful part of a Macintosh file on a non-Macintosh computer system. For example, if a Macintosh user wants to send a file of data to a user on an IBM-PC, she would only send the Data fork.

- RESOURCE FORK

  Contains a collection of arbitrary attribute/value pairs, including program segments, icon bitmaps, and parametric values.

Additional information regarding Macintosh files is stored by the Finder in a hidden file, called the "Desktop Database". Because of the complications in storing different parts of a Macintosh file in a non-Macintosh filesystem that only handles consecutive data in one part, it is common to convert the Macintosh file into some other format before transferring it over the network.

MIME format for BinHex4.0

MIME-base Apple information is specified by:

MIME type-name:	APPLICATION
MIME subtype name:	MAC-BINHEX40
Required parameters:	none
Optional parameters:	NAME, which must be a "value" as defined in RFC-1521.
Encoding considerations:	none
Security considerations:	See separate section in the document
Published specification:	Appendix A
Rationale:	Permits MIME-based transmission of data with Apple Macintosh file system specific information using a currently popular, though platform specific, format.

BinHex

BinHex 4.0 is a popular means of encoding Macintosh files for archiving on non-Macintosh file systems and for transmission via Internet mail. (See Appendix A for a brief description of the BinHex 4.0 format.)

The content-type application/mac-binhex40 indicates that the body of the mail is a BinHex4.0 file. Even though the BinHex encoding consists of characters which are not the same as those used in Base64 (those regarded as safe according to RFC-1521 [BORE93]) a transportation encoding should not be done. Even though a BinHex file includes the original Macintosh filename, it is recommended that a name parameter be included on the Content-Type header to give the recipient a hint as to what file is attached. The value of the name parameter must be a "value" as defined by RFC-1521 [BORE93]. Note that this restricts the value to seven-bit US-ASCII characters.

Request for Comments: 1777
Lightweight Directory Access Protocol

The protocol described in this document is designed to provide access to the X.500 Directory while not incurring the resource requirements of the Directory Access Protocol (DAP). This protocol is specifically targeted at simple management applications and browser applications that provide simple read/write interactive access to the X.500 Directory, and is intended to be a complement to the DAP itself.

Key aspects of LDAP are:

- Protocol elements are carried directly over TCP or other transport, bypassing much of the session/presentation overhead.
- Many protocol data elements are encoding as ordinary strings (e.g.,Distinguished Names).
- A lightweight BER encoding is used to encode all protocol elements.

The general model adopted by this protocol is one of clients performing protocol operations against servers. In this model, this is accomplished by a client transmitting a protocol request describing the operation to be performed to a server, which is then responsible for performing the

necessary operations on the Directory. Upon completion of the necessary operations, the server returns a response containing any results or errors to the requesting client. In keeping with the goal of easing the costs associated with use of the Directory, it is an objective of this protocol to minimize the complexity of clients so as to facilitate widespread deployment of applications capable of utilizing the Directory. Note that, although servers are required to return responses whenever such responses are defined in the protocol, there is no requirement for synchronous behavior on the part of either client or server implementations: requests and responses for multiple operations may be exchanged by client and servers in any order, as long as clients eventually receive a response for every request that requires one.

Consistent with the model of servers performing protocol operations on behalf of clients, it is also to be noted that protocol servers are expected to handle referrals without resorting to the return of such referrals to the client. This protocol makes no provisions for the return of referrals to clients, as the model is one of servers ensuring the performance of all necessary operations in the Directory, with only final results or errors being returned by servers to clients. Note that this protocol can be mapped to a strict subset of the directory abstract service, so it can be cleanly provided by the DAP.3. Mapping Onto Transport Services This protocol is designed to run over connection-oriented, reliable transports, with all 8 bits in an octet being significant in the data stream. Specifications for two underlying services are defined here, though others are also possible.

Transmission Control Protocol (TCP)

The LDAPMessage PDUs are mapped directly onto the TCP bytestream. Server implementations running over the TCP should provide a protocol listener on port 389.

Connection Oriented Transport Service (COTS)

The connection is established. No special use of T-Connect is made. Each LDAPMessage PDU is mapped directly onto T-Data.

Elements of Protocol

For the purposes of protocol exchanges, all protocol operations are encapsulated in a common envelope, the LDAPMessage, which is defined as follows:

```
LDAPMessage ::=
    SEQUENCE {
        messageID       MessageID,
        protocolOp      CHOICE {
                            bindRequest         BindRequest,
                            bindResponse        BindResponse,
                            unbindRequest       UnbindRequest,
                            searchRequest       SearchRequest,
                            searchResponse      SearchResponse,
                            modifyRequest       ModifyRequest,
                            modifyResponse      ModifyResponse,
                            addRequest          AddRequest,
                            addResponse         AddResponse,
                            delRequest          DelRequest,
                            delResponse         DelResponse,
                            modifyRDNRequest ModifyRDNRequest,
                            modifyRDNResponseModifyRDNResponse,
                            compareDNRequestCompareRequest,
                            compareDNResponseCompareResponse,
```

```
                abandonRequest     AbandonRequest
            }
    }
    MessageID ::= INTEGER (0 .. maxInt)
```

The function of the LDAPMessage is to provide an envelope containing common fields required in all protocol exchanges. At this time the only common field is a message ID, which is required to have a value different from the values of any other requests outstanding in the LDAP session of which this message is a part.

Request for Comments: 1779
A String Representation of Distinguished Names

The OSI Directory uses distinguished names as the primary keys to entries in the directory. Distinguished Names are encoded in ASN.1. When a distinguished name is communicated between to users not using a directory protocol (e.g., in a mail message), there is a need to have a user-oriented string representation of distinguished name. This specification defines a string format for representing names, which is designed to give a clean representation of commonly used names, whilst being able to represent any distinguished name.

Why a notation is needed

Many OSI Applications make use of Distinguished Names (DN) as defined in the OSI Directory, commonly known as X.500. This specification assumes familiarity with X.500, and the concept of Distinguished Name. It is important to have a common format to be able to unambiguously represent a distinguished name. This might be done to represent a directory name on a business card or in an email message. There is a need for a format to support human to human communication, which must be string based (not ASN.1) and user oriented. This notation is targeted towards a general user oriented system, and in particular to represent the names of humans. Other syntaxes may be more appropriate for other uses of the directory. For example, the OSF Syntax may be more appropriate for some system oriented uses. (The OSF Syntax uses "/" as a separator, and forms names in a manner intended to resemble UNIX filenames).

A notation for Distinguished Name

The following goals are laid out:

- To provide an unambiguous representation of a distinguished name
- To be an intuitive format for the majority of names
- To be fully general, and able to represent any distinguished name
- To be amenable to a number of different layouts to achieve an attractive representation.
- To give a clear representation of the contents of the distinguished name

This notation is designed to be convenient for common forms of name. Some examples are given. The author's directory distinguished name would be written:

```
CN=Steve Kille,
O=ISODE Consortium, C=GB
```

This may be folded, perhaps to display in multi-column format. For example:

```
CN=Steve Kille,
```

O=ISODE Consortium,
C=GB

Another name might be:
CN=Christian Huitema, O=INRIA, C=FR

Semicolon (";") may be used as an alternate separator. The separators may be mixed, but this usage is discouraged.
CN=Christian Huitema; O=INRIA; C=FR

In running text, this would be written as
<CN=Christian Huitema;
O=INRIA; C=FR>.

Another example, shows how different attribute types
are handled:
CN=James Hacker,
L=Basingstoke,
O=Widget Inc,
C=GB

Here is an example of a multi-valued Relative Distinguished Name, where the namespace is flat within an organization, and department is used to disambiguate certain names:
OU=Sales + CN=J. Smith, O=Widget Inc., C=US

The final examples show both methods quoting of a comma in an Organization name:
CN=L. Eagle, O="Sue, Grabbit and Runn", C=GB
CN=L. Eagle, O=Sue\, Grabbit and Runn, C=GB

Formal definition

A formal definition can now be given. The structure is specified in a BNF grammar in Figure 1. This BNF uses the grammar defined in RFC 822, with the terminals enclosed in "<>". This definition is in an abstract character set, and so may be written in any character set supporting the explicitly defined special characters. The quoting mechanism is used for the following cases:

- Strings containing ",", "+", "=" or """ , <CR>, "<", ">", "#", or ";".
- Strings with leading or trailing spaces
- Strings containing consecutive spaces

There is an escape mechanism from the normal user oriented form, so that this syntax may be used to print any valid distinguished name. This is ugly. It is expected to be used only in pathological cases.

There are two parts to this mechanism:

- Attributes types are represented in a (big-endian) dotted notation. (e.g., OID.2.6.53).
- Attribute values are represented in hexadecimal (e.g. #0A56CF). Each pair of hex digits defines an octet, which is the ASN.1 Basic Encoding Rules value of the Attribute Value.

The keyword specification is optional in the BNF, but mandatory for this specification. This is so that the same BNF may be used for the related specification on User Friendly Naming. When this specification is followed, the attribute type keywords must always be present.

A list of valid keywords for well known attribute types used in naming is given in Table 1. Keywords may contain spaces, but shall not have leading or trailing spaces. This is a list of keywords which must be supported. These are chosen because they appear in common forms of name, and can do so in a place which does not correspond to the default schema used. A register of valid keywords is maintained by the IANA.

Request for Comments: 1806 Communicating Presentation Information in Internet Messages: The Content-Disposition Header

This memo provides a mechanism whereby messages conforming to the RFC 1521 ("MIME") specification can convey presentational information. It specifies a new "Content-Disposition" header, optional and valid for any RFC 1521 entity ("message" or "body part"). Two values for this header are described in this memo; one for the ordinary linear presentation of the body part, and another to facilitate the use of mail to transfer files. It is expected that more values will be defined in the future, and procedures are defined for extending this set of values.

RFC 1521 specifies a standard format for encapsulating multiple pieces of data into a single Internet message. That document does not address the issue of presentation styles; it provides a framework for the interchange of message content, but leaves presentation issues solely in the hands of mail user agent (MUA) implementors. Two common ways of presenting multipart electronic messages are as a main document with a list of separate attachments, and as a single document with the various parts expanded (displayed) inline. The display of an attachment is generally construed to require positive action on the part of the recipient, while inline message components are displayed automatically when the message is viewed. A mechanism is needed to allow the sender to transmit this sort of presentational information to the recipient; the Content-Disposition header provides this mechanism, allowing each component of a message to be tagged with an indication of its desired presentation semantics. Tagging messages in this manner will often be sufficient for basic message formatting. However, in many cases a more powerful and flexible approach will be necessary. The definition of such approaches is beyond the scope of this memo; however, such approaches can benefit from additional Content-Disposition values and parameters, to be defined at a later date.

In addition to allowing the sender to specify the presentational disposition of a message component, it is desirable to allow her to indicate a default archival disposition; a filename. The optional "filename" parameter provides for this.

The Content-Disposition Header Field

Content-Disposition is an optional header; in its absence, the MUA may use whatever presentation method it deems suitable. It is desirable to keep the set of possible disposition types small and well defined, to avoid needless complexity. Even so, evolving usage will likely require the definition of additional disposition types or parameters, so the set of disposition values is extensible.
In the extended BNF notation of RFC 822, the Content-Disposition header field is defined as follows:

disposition := "Content-Disposition" ":"
disposition-type

*(";" disposition-parm)
disposition-type := "inline"
/ "attachment"
/ extension-token
; values are not case-sensitive
disposition-parm := filename-parm / parameter
filename-parm := "filename" "=" value;

`Extension-token', `parameter' and `value' are defined according to RFC 822 and RFC 1521.

Request for Comments: 1823
The LDAP Application Program Interface

This document defines a C language application program interface to the lightweight directory access protocol (LDAP). The LDAP API is designed to be powerful, yet simple to use. It defines compatible synchronous and asynchronous interfaces to LDAP to suit a wide variety of applications. This document gives a brief overview of the LDAP model, then an overview of how the API is used by an application program to obtain LDAP information. The API calls are described in detail, followed by an appendix that provides some example code demonstrating the use of the API.

Overview of the LDAP Model

LDAP is the lightweight directory access protocol. It can provide a lightweight frontend to the X.500 directory, or a stand-alone service. In either mode, LDAP is based on a client-server model in which a client makes a TCP connection to an LDAP server, over which it sends requests and receives responses.

The LDAP information model is based on the entry, which contains information about some object (e.g., a person). Entries are composed of attributes, which have a type and one or more values. Each attribute has a syntax that determines what kinds of values are allowed in the attribute (e.g., ASCII characters, a jpeg photograph, etc.) and how those values behave during directory operations (e.g., is case significant during comparisons).

Entries are organized in a tree structure, usually based on political, geographical, and organizational boundaries. Each entry is uniquely named relative to its sibling entries by its relative distinguished name (RDN) consisting of one or more distinguished attribute values from the entry. At most one value from each attribute may be used in the RDN. For example, the entry for the person Babs Jensen might be named with the "Barbara Jensen" value from the commonName attribute. A globally unique name for an entry, called a distinguished name or DN, is constructed by concatenating the sequence of RDNs from the root of the tree down to the entry. For example, if Babs worked for the University of Michigan, the DN of her U-M entry might be "cn=Barbara Jensen, o=University of Michigan, c=US". Operations are provided to authenticate, search for and retrieve information, modify information, and add and delete entries from the tree. The next sections give an overview of how the API is used and detailed descriptions of the LDAP API calls that implement all of these functions.

Overview of LDAP API Use

An application generally uses the LDAP API in four simple steps.

- Open a connection to an LDAP server. The ldap_open() call returns a handle to the connection, allowing multiple connections to be open at once.
- Authenticate to the LDAP server and/or the X.500 DSA. The ldap_bind() call and friends support a variety of authentication methods.
- Perform some LDAP operations and obtain some results. ldap_search() and friends return results which can be parsed by ldap_result2error(), ldap_first_entry(), ldap_next_entry(), etc.
- Close the connection. The ldap_unbind() call closes the connection.

Operations can be performed either synchronously or asynchronously. Synchronous calls end in _s. For example, a synchronous search can be completed by calling ldap_search_s(). An asynchronous search can be initiated by calling ldap_search(). All synchronous routines return an indication of the outcome of the operation (e.g, the constant LDAP_SUCCESS or some other error code). The asynchronous routines return the message id of the operation initiated. This id can be used in subsequent calls to ldap_result() to obtain the result(s) of the operation. An asynchronous operation can be abandoned by calling ldap_abandon().

Request for Comments: 1891 SMTP Service Extension for Delivery Status Notifications

This memo defines an extension to the SMTP service, which allows an SMTP client to specify (a) that delivery status notifications (DSNs) should be generated under certain conditions, (b) whether such notifications should return the contents of the message, and (c) additional information, to be returned with a DSN, that allows the sender to identify both the recipient(s) for which the DSN was issued, and the transaction in which the original message was sent.

The SMTP protocol requires that an SMTP server provide notification of delivery failure, if it determines that a message cannot be delivered to one or more recipients. Traditionally, such notification consists of an ordinary Internet mail message, sent to the envelope sender address (the argument of the SMTP MAIL command), containing an explanation of the error and at least the headers of the failed message. Experience with large mail distribution lists indicates that such messages are often insufficient to diagnose problems, or even to determine at which host or for which recipients a problem occurred. In addition, the lack of a standardized format for delivery notifications in Internet mail makes it difficult to exchange such notifications with other message handling systems. Such experience has demonstrated a need for a delivery status notification service for Internet electronic mail, which:

- is reliable, in the sense that any DSN request will either be honored at the time of final delivery, or result in a response that indicates that the request cannot be honored,
- when both success and failure notifications are requested, provides an unambiguous and nonconflicting indication of whether delivery of a message to a recipient succeeded or failed,
- is stable, in that a failed attempt to deliver a DSN should never result in the transmission of another DSN over the network,
- preserves sufficient information to allow the sender to identify both the mail transaction and the recipient address which caused the notification, even when mail is forwarded or gatewayed to foreign environments, and

- interfaces acceptably with non-SMTP and non-822-based mail systems, both so that notifications returned from foreign mail systems may be useful to Internet users, and so that the notification requests from foreign environments may be honored.

Among the requirements implied by this goal are the ability to request non-return-of-content, and the ability to specify whether positive delivery notifications, negative delivery notifications, both, or neither, should be issued.

Framework for the Delivery Status Notification Extension

The following service extension is therefore defined:

- the name of the SMTP service extension is "Delivery Status Notification";
- the EHLO keyword value associated with this extension is "DSN", the meaning of which is defined in section 4 of this memo;
- no parameters are allowed with this EHLO keyword value;
- two optional parameters are added to the RCPT command, and two optional parameters are added to the MAIL command.
- no additional SMTP verbs are defined by this extension.

The Delivery Status Notification service extension

An SMTP client wishing to request a DSN for a message may issue the EHLO command to start an SMTP session, to determine if the server supports any of several service extensions. If the server responds with code 250 to the EHLO command, and the response includes the EHLO keyword DSN, then the Delivery Status Notification extension (as described in this memo) is supported. Ordinarily, when an SMTP server returns a positive (2xx) reply code in response to a RCPT command, it agrees to accept responsibility for either delivering the message to the named recipient, or sending a notification to the sender of the message indicating that delivery has failed. However, an extended SMTP ("ESMTP") server which implements this service extension will accept an optional NOTIFY parameter with the RCPT command. If present, the NOTIFY parameter alters the conditions for generation of delivery status notifications from the default (issue notifications only on failure). The ESMTP client may also request (via the RET parameter) whether the entire contents of the original message should be returned (as opposed to just the headers of that message), along with the DSN.

In general, an ESMTP server which implements this service extension will propagate delivery status notification requests when relaying mail to other SMTP-based MTAs which also support this extension, and make a "best effort" to ensure that such requests are honored when messages are passed into other environments.

In order that any delivery status notifications thus generated will be meaningful to the sender, any ESMTP server which supports this extension will attempt to propagate the following information to any other MTAs that are used to relay the message, for use in generating DSNs:

- for each recipient, a copy of the original recipient address, as used by the sender of the message.
- for the entire SMTP transaction, an envelope identification string, which may be used by the sender to associate any delivery status notifications with the transaction used to send the original message.

Request for Comments: 1892
The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages

The Multipart/Report MIME content-type

The Multipart/Report MIME content-type is a general "family" or "container" type for electronic mail reports of any kind. Although this memo defines only the use of the Multipart/Report content-type with respect to delivery status reports, mail processing programs will benefit if a single content-type is used to for all kinds of reports.

The Multipart/Report content-type is defined as follows:

MIME type name: multipart
MIME subtype name: report
Required parameters: boundary, report-type
Optional parameters: none
Encoding considerations: 7bit should always be adequate
Security considerations: see section 4 of this memo.

When used to send a report, the Multipart/Report content-type must be the top-level MIME content type for any report message. The report-type parameter identifies the type of report. The parameter is the MIME content sub-type of the second body part of the Multipart/Report.

User agents and gateways must be able to automatically determine that a message is a mail system report and should be processed as such. Placing the Multipart/Report as the outermost content provides a mechanism whereby an auto-processor may detect through parsing the RFC 822 headers that the message is a report.

The Multipart/Report content-type contains either two or three sub-parts, in the following order:

- [required] The first body part contains human readable message. The purpose of this message is to provide an easily-understood description of the condition(s) that caused the report to be generated, for a human reader who may not have an user agent capable of interpreting the second section of the Multipart/Report. The text in the first section may be in any MIME standards-track content-type, charset, or language. Where a description of the error is desired in several languages or several media, a Multipart/Alternative construct may be used. This body part may also be used to send detailed information that cannot be easily formatted into a Message/Report body part.
- [required] A machine parsable body part containing an account of the reported message handling event. The purpose of this body part is to provide a machine-readable description of the condition(s) which caused the report to be generated, along with details not present in the first body part that may be useful to human experts.
- [optional] A body part containing the returned message or a portion thereof. This information may be useful to aid human experts in diagnosing problems. (Although it may also be useful to allow the sender to identify the message which the report was issued, it is hoped that the envelope-id and original-recipient- address returned in the Message/Report body part will replace the traditional use of the returned content for this purpose.) Return of content may be wasteful of network bandwidth and a variety of implementation strategies can be used. Generally the sender should choose the appropriate strategy and inform the recipient of the required level of returned content required. In the absence of an

explicit request for level of return of content such as that provided in [DRPT], the agent which generated the delivery service report should return the full message content. When data not encoded in 7 bits is to be returned, and the return path is not guaranteed to be 8-bit capable, two options are available. The original message MAY be reencoded into a legal 7 bit MIME message or the Text/RFC822-Headers content-type MAY be used to return only the original message headers.

The Text/RFC822-Headers MIME content-type

The Text/RFC822-Headers MIME content-type provides a mechanism to label and return only the RFC 822 headers of a failed message. These headers are not the complete message and should not be returned as a Message/RFC822. The returned headers are useful for identifying the failed message and for diagnostics based on the received: lines.

The Text/RFC822-Headers content-type is defined as follows:

MIME type name: Text
MIME subtype name: RFC822-Headers
Required parameters: None
Optional parameters: none
Encoding considerations: 7 bit is sufficient for normal RFC822 headers, however, if the headers are broken and require encoding, they may be encoded in quoted-printable.
Security considerations: see section 4 of this memo.

The Text/RFC822-headers body part should contain all the RFC822 header lines from the message which caused the report. The RFC822 headers include all lines prior to the blank line in the message. They include the MIME-Version and MIME Content- headers.

Request for Comments: 1893
Enhanced Mail System Status Codes

There currently is not a standard mechanism for the reporting of mail system errors except for the limited set offered by SMTP and the system specific text descriptions sent in mail messages. There is a pressing need for a rich machine readable status code for use in delivery status notifications. This document proposes a new set of status codes for this purpose.

SMTP error codes have historically been used for reporting mail system errors. Because of limitations in the SMTP code design, these are not suitable for use in delivery status notifications. SMTP provides about 12 useful codes for delivery reports. The majority of the codes are protocol specific response codes such as the 354 response to the SMTP data command. Each of the 12 useful codes are each overloaded to indicate several error conditions each. SMTP suffers some scars from history, most notably the unfortunate damage to the reply code extension mechanism by uncontrolled use.

This proposal facilitates future extensibility by requiring the client to interpret unknown error codes according to the theory of codes while requiring servers to register new response codes. The SMTP theory of reply codes partitioned in the number space such a manner that the remaining available codes will not provide the space needed. The most critical example is the existence of only 5 remaining codes for mail system errors. The mail system classification includes both host

and mailbox error conditions. The remaining third digit space would be completely consumed as needed to indicate MIME and media conversion errors and security system errors. A revision to the SMTP theory of reply codes to better distribute the error conditions in the number space will necessarily be incompatible with SMTP. Further, consumption of the remaining reply-code number space for delivery notification reporting will reduce the available codes for new ESMTP extensions.

The following proposal is based on the SMTP theory of reply codes. It adopts the success, permanent error, and transient error semantics of the first value, with a further description and classification in the second. This proposal re-distributes the classifications to better distribute the error conditions, such as separating mailbox from host errors.

Status Codes

This document defines a new set of status codes to report mail system conditions. These status codes are intended to be used for media and language independent status reporting. They are not intended for system specific diagnostics. The syntax of the new status codes is defined as:

status-code = class "." subject "." detail
class = "2"/"4"/"5"
*subject = 1*3digit*
*detail = 1*3digit*

White-space characters and comments are NOT allowed within a status-code. Each numeric sub-code within the status-code MUST be expressed without leading zero digits. Status codes consist of three numerical fields separated by ".". The first sub-code indicates whether the delivery attempt was successful. The second sub-code indicates the probable source of any delivery anomalies, and the third sub-code indicates a precise error condition.

The codes space defined is intended to be extensible only by standards track documents. Mail system specific status codes should be mapped as close as possible to the standard status codes. Servers should send only defined, registered status codes. System specific errors and diagnostics should be carried by means other than status codes.

New subject and detail codes will be added over time. Because the number space is large, it is not intended that published status codes will ever be redefined or eliminated. Clients should preserve the extensibility of the code space by reporting the general error described in the subject sub-code when the specific detail is unrecognized.

The class sub-code provides a broad classification of the status. The enumerated values the class are defined as:

2.X.X Success

Success specifies that the DSN is reporting a positive delivery action. Detail sub-codes may provide notification of transformations required for delivery.

4.X.X Persistent Transient Failure

A persistent transient failure is one in which the message as sent is valid, but some temporary event prevents the successful sending of the message. Sending in the future may be successful.

5.X.X Permanent Failure

A permanent failure is one which is not likely to be resolved by resending the message in the cur-

rent form. Some change to the message or the destination must be made for successful delivery.

A client must recognize and report class sub-code even where subsequent subject sub-codes are unrecognized. The subject sub-code classifies the status. This value applies to each of the three classifications. The subject sub-code, if recognized, must be reported even if the additional detail provided by the detail sub-code is not recognized. The enumerated values for the subject sub-code are:

- X.0.X Other or Undefined Status

  There is no additional subject information available.

- X.1.X Addressing Status

  The address status reports on the originator or destination address. It may include address syntax or validity. These errors can generally be corrected by the sender and retried.

- X.2.X Mailbox Status

  Mailbox status indicates that something having to do with the mailbox has cause this DSN. Mailbox issues are assumed to be under the general control of the recipient.

- X.3.X Mail System Status

  Mail system status indicates that something having to do with the destination system has caused this DSN. System issues are assumed to be under the general control of the destination system administrator.

- X.4.X Network and Routing Status

  The networking or routing codes report status about the delivery system itself. These system components include any necessary infrastructure such as directory and routing services. Network issues are assumed to be under the control of the destination or intermediate system administrator.

- X.5.X Mail Delivery Protocol Status

  The mail delivery protocol status codes report failures involving the message delivery protocol. These failures include the full range of problems resulting from implementation errors or an unreliable connection. Mail delivery protocol issues may be controlled by many parties including the originating system, destination system, or intermediate system administrators.

- X.6.X Message Content or Media Status

  The message content or media status codes report failures involving the content of the message. These codes report failures due to translation, transcoding, or otherwise unsupported message media. Message content or media issues are under the control of both the sender and the receiver, both of whom must support a common set of supported content-types.

- X.7.X Security or Policy Status

  The security or policy status codes report failures involving policies such as per-

recipient or per-host filtering and cryptographic operations. Security and policy status issues are assumed to be under the control of either or both the sender and recipient. Both the sender and recipient must permit the exchange of messages and arrange the exchange of necessary keys and certificates for cryptographic operations.

Request for Comments: 1894
An Extensible Message Format for Delivery Status Notifications

This memo defines a MIME content-type that may be used by a message transfer agent (MTA) or electronic mail gateway to report the result of an attempt to deliver a message to one or more recipients. This content-type is intended as a machine-processable replacement for the various types of delivery status notifications currently used in Internet electronic mail. Because many messages are sent between the Internet and other messaging systems (such as X.400 or the so-called "LAN-based" systems), the DSN protocol is designed to be useful in a multi-protocol messaging environment. To this end, the protocol described in this memo provides for the carriage of "foreign" addresses and error codes, in addition to those normally used in Internet mail. Additional attributes may also be defined to support "tunneling" of foreign notifications through Internet mail.

This memo defines a MIME content-type for delivery status notifications (DSNs). A DSN can be used to notify the sender of a message of any of several conditions: failed delivery, delayed delivery, successful delivery, or the gatewaying of a message into an environment that may not support DSNs. This memo defines only the format of the notifications.

The DSNs defined in this memo are expected to serve several purposes:

- Inform human beings of the status of message delivery processing, as well as the reasons for any delivery problems or outright failures, in a manner which is largely independent of human language;
- Allow mail user agents to keep track of the delivery status of messages sent, by associating returned DSNs with earlier message transmissions;
- Allow mailing list exploders to automatically maintain their subscriber lists when delivery attempts repeatedly fail;
- Convey delivery and non-delivery notifications resulting from attempts to deliver messages to "foreign" mail systems via a gateway;
- Allow "foreign" notifications to be tunneled through a MIME-capable message system and back into the original messaging system that issued the original notification, or even to a third messaging system;
- Allow language-independent, yet reasonably precise, indications of the reason for the failure of a message to be delivered (once status codes of sufficient precision are defined); and
- Provide sufficient information to remote MTA maintainers (via "trouble tickets") so that they can understand the nature of reported errors. This feature is used in the case that failure to deliver a message is due to the malfunction of a remote MTA and the sender wants to report the problem to the remote MTA administrator.

Format of a Delivery Status Notification

A DSN is a MIME message with a top-level content-type of multipart/report. When a multipart/report content is used to transmit a DSN:

- The report-type parameter of the multipart/report content is "delivery-status".
- The first component of the multipart/report contains a human-readable explanation of the DSN.
- The second component of the multipart/report is of content-type message/delivery-status.
- If the original message or a portion of the message is to be returned to the sender, it appears as the third component of the multipart/report.

Request for Comments: 2045
Multipurpose Internet Mail Extensions (MIME)
Part One: Format of Internet Message Bodies

STD 11, RFC 822 defines a message representation protocol specifying considerable detail about US-ASCII message headers, but which leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow for

- textual message bodies in character sets other than US-ASCII,
- an extensible set of different formats for non-textual message bodies,
- multi-part message bodies, and
- textual header information in character sets other than US-ASCII.

Since its publication in 1982, RFC 822 has defined the standard format of textual mail messages on the Internet. Its success has been such that the RFC 822 format has been adopted, wholly or partially, well beyond the confines of the Internet and the Internet SMTP transport defined by RFC 821. As the format has seen wider use, a number of limitations have proven increasingly restrictive for the user community.

RFC 822 was intended to specify a format for text messages. As such, non-text messages, such as multimedia messages that might include audio or images, are simply not mentioned. Even in the case of text, however, RFC 822 is inadequate for the needs of mail users whose languages require the use of character sets richer than US-ASCII. Since RFC 822 does not specify mechanisms for mail containing audio, video, Asian language text, or even text in most European languages, additional specifications are needed.

One of the notable limitations of RFC 821/822 based mail systems is the fact that they limit the contents of electronic mail messages to relatively short lines (e.g. 1000 characters or less [RFC-821]) of 7bit US-ASCII. This forces users to convert any non-textual data that they may wish to send into seven-bit bytes representable as printable US-ASCII characters before invoking a local mail UA (User Agent, a program with which human users send and receive mail).

The limitations of RFC 822 mail become even more apparent as gateways are designed to allow for the exchange of mail messages between RFC 822 hosts and X.400 hosts. X.400 [X400] specifies mechanisms for the inclusion of non-textual material within electronic mail messages.

This document describes several mechanisms that combine to solve most of these problems without introducing any serious incompatibilities with the existing world of RFC 822 mail. In particular, it describes:

- A MIME-Version header field, which uses a version number to declare a message to be conformant with MIME and allows mail processing agents to distinguish between such messages and those generated by older or non-conformant software, which are presumed to lack such a field.
- A Content-Type header field, generalized from RFC 1049, which can be used to specify the media type and subtype of data in the body of a message and to fully specify the native representation (canonical form) of such data.
- A Content-Transfer-Encoding header field, which can be used to specify both the encoding transformation that was applied to the body and the domain of the result. Encoding transformations other than the identity transformation are usually applied to data in order to allow it to pass through mail transport mechanisms which may have data or character set limitations.
- Two additional header fields that can be used to further describe the data in a body, the Content-ID and Content-Description header fields.

All of the header fields defined in this document are subject to the general syntactic rules for header fields specified in RFC 822. In particular, all of these header fields except for Content-Disposition can include RFC 822 comments, which have no semantic content and should be ignored during MIME processing. Finally, to specify and promote interoperability, RFC 2049 provides a basic applicability statement for a subset of the above mechanisms that defines a minimal level of "conformance" with this document.

Request for Comments: 2046 Multipurpose Internet Mail Extensions(MIME) Part Two: Media Types

STD 11, RFC 822 defines a message representation protocol specifying considerable detail about US-ASCII message headers, but which leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow for

- textual message bodies in character sets other than US-ASCII,
- an extensible set of different formats for non-textual message bodies,
- multi-part message bodies, and
- textual header information in character sets other than US-ASCII.

The first document in this set, RFC 2045, defines a number of header fields, including Content-Type. The Content-Type field is used to specify the nature of the data in the body of a MIME entity, by giving media type and subtype identifiers, and by providing auxiliary information that may be required for certain media types. After the type and subtype names, the remainder of the header field is simply a set of parameters, specified in an attribute/value notation. The ordering of parameters is not significant.

In general, the top-level media type is used to declare the general type of data, while the subtype specifies a specific format for that type of data. Thus, a media type of "image/xyz" is enough to tell

a user agent that the data is an image, even if the user agent has no knowledge of the specific image format "xyz". Such information can be used, for example, to decide whether or not to show a user the raw data from an unrecognized subtype -- such an action might be reasonable for unrecognized subtypes of "text", but not for unrecognized subtypes of "image" or "audio". For this reason, registered subtypes of "text", "image", "audio", and "video" should not contain embedded information that is really of a different type. Such compound formats should be represented using the "multipart" or "application" types. Parameters are modifiers of the media subtype, and as such do not fundamentally affect the nature of the content. The set of meaningful parameters depends on the media type and subtype. Most parameters are associated with a single specific subtype. However, a given top-level media type may define parameters which are applicable to any subtype of that type. Parameters may be required by their defining media type or subtype or they may be optional. MIME implementations must also ignore any parameters whose names they do not recognize.

MIME's Content-Type header field and media type mechanism has been carefully designed to be extensible, and it is expected that the set of media type/subtype pairs and their associated parameters will grow significantly over time. Several other MIME facilities, such as transfer encodings and "message/external-body" access types, are likely to have new values defined over time. In order to ensure that the set of such values is developed in an orderly, well-specified, and public manner, MIME sets up a registration process which uses the Internet Assigned Numbers Authority (IANA) as a central registry for MIME's various areas of extensibility. The registration process for these areas is described in a companion document, RFC 2048.

Definition of a Top-Level Media Type

The definition of a top-level media type consists of:

- a name and a description of the type, including criteria for whether a particular type would qualify under that type,
- the names and definitions of parameters, if any, which are defined for all subtypes of that type (including whether such parameters are required or optional),
- how a user agent and/or gateway should handle unknown subtypes of this type,
- general considerations on gatewaying entities of this top-level type, if any, and
- any restrictions on content-transfer-encodings for entities of this top-level type.

Overview Of The Initial Top-Level Media Types

The five discrete top-level media types are:

- text -- textual information.

  The subtype "plain" in particular indicates plain text containing no formatting commands or directives of any sort. Plain text is intended to be displayed "as-is". No special software is required to get the full meaning of the text, aside from support for the indicated character set. Other subtypes are to be used for enriched text in forms where application software may enhance the appearance of the text, but such software must not be required in order to get the general idea of the content. Possible subtypes of "text" thus include any word processor format that can be read without resorting to software that understands the format. In particular, formats that employ embedded binary formatting information are not considered directly readable. A very simple and portable subtype, "richtext", was defined in RFC 1341, with a further revision in RFC 1896 under the name "enriched".

- image -- image data.

"Image" requires a display device (such as a graphical display, a graphics printer, or a FAX machine) to view the information. An initial subtype is defined for the widely-used image format JPEG subtypes are defined for two widely-used image formats, jpeg and gif.

• audio -- audio data.

"Audio" requires an audio output device (such as a speaker or a telephone) to "display" the contents. An initial subtype "basic" is defined in this document.

• video -- video data.

"Video" requires the capability to display moving images, typically including specialized hardware and software. An initial subtype "mpeg" is defined in this document.

• application -- some other kind of data, typically either uninterpreted binary data or information to be processed by an application.

The subtype "octet-stream" is to be used in the case of uninterpreted binary data, in which case the simplest recommended action is to offer to write the information into a file for the user. The "PostScript" subtype is also defined for the transport of Post Script material. Other expected uses for "application" include spreadsheets, data for mail-based scheduling systems, and languages for "active" (computational) messaging, and word processing formats that are not directly readable.

The two composite top-level media types are:

• multipart -- data consisting of multiple entities of independent data types.

Four subtypes are initially defined, including the basic "mixed" subtype specifying a generic mixed set of parts, "alternative" for representing the same data in multiple formats, "parallel" for parts intended to be viewed simultaneously, and "digest" for multipart entities in which each part has a default type of "message/rfc822".

• message -- an encapsulated message.

A body of media type "message" is itself all or a portion of some kind of message object. Such objects may or may not in turn contain other entities. The "rfc822" sub type is used when the encapsulated content is itself an RFC 822 message. The "partial" subtype is defined for partial RFC 822 messages, to permit the fragmented transmission of bodies that are thought to be too large to be passed through transport facilities in one piece. Another subtype, "external-body", is defined for specifying large bodies by reference to an external data source. It should be noted that the list of media type values given here may be augmented in time, via the mechanisms described above, and that the set of subtypes is expected to grow substantially.

Request for Comments: 2047
MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text

STD 11, RFC 822, defines a message representation protocol specifying considerable detail about US-ASCII message headers, and leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow for

- textual message bodies in character sets other than US-ASCII,
- an extensible set of different formats for non-textual message bodies,
- multi-part message bodies, and
- textual header information in character sets other than US-ASCII.

These documents are based on earlier work documented in RFC 934, STD 11, and RFC 1049, but extends and revises them. Because RFC 822 said so little about message bodies, these documents are largely orthogonal to (rather than a revision of) RFC 822.

This particular document is the third document in the series. It describes extensions to RFC 822 to allow non-US-ASCII text data in Internet mail header fields.

RFC 2045 describes a mechanism for denoting textual body parts which are coded in various character sets, as well as methods for encoding such body parts as sequences of printable US-ASCII characters. This memo describes similar techniques to allow the encoding of non-ASCII text in various portions of a RFC 822 [2] message header, in a manner which is unlikely to confuse existing message handling software. Like the encoding techniques described in RFC 2045, the techniques outlined here were designed to allow the use of non-ASCII characters in message headers in a way which is unlikely to be disturbed by the quirks of existing Internet mail handling programs. In particular, some mail relaying programs are known to (a) delete some message header fields while retaining others, (b) rearrange the order of addresses in To or Cc fields, (c) rearrange the (vertical) order of header fields, and/or (d) "wrap" message headers at different places than those in the original message. In addition, some mail reading programs are known to have difficulty correctly parsing message headers which, while legal according to RFC 822, make use of backslash-quoting to "hide" special characters such as "<", ",", or ":", or which exploit other infrequently-used features of that specification.

While it is unfortunate that these programs do not correctly interpret RFC 822 headers, to "break" these programs would cause severe operational problems for the Internet mail system. The extensions described in this memo therefore do not rely on little-used features of RFC 822.

Instead, certain sequences of "ordinary" printable ASCII characters (known as "encoded-words") are reserved for use as encoded data. The syntax of encoded-words is such that they are unlikely to "accidentally" appear as normal text in message headers. Furthermore, the characters used in encoded-words are restricted to those which do not have special meanings in the context in which the encoded-word appears.

Generally, an "encoded-word" is a sequence of printable ASCII characters that begins with "=?", ends with "?=", and has two "?"s in between. It specifies a character set and an encoding method,

and also includes the original text encoded as graphic ASCII characters, according to the rules for that encoding method. A mail composer that implements this specification will provide a means of inputting non-ASCII text in header fields, but will translate these fields (or appropriate portions of these fields) into encoded-words before inserting them into the message header. A mail reader that implements this specification will recognize encoded-words when they appear in certain portions of the message header. Instead of displaying the encoded-word "as is", it will reverse the encoding and display the original text in the designated character set.

Request for Comments: 2252 Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions

The Lightweight Directory Access Protocol (LDAP) requires that the contents of AttributeValue fields in protocol elements be octet strings. This document defines a set of syntaxes for LDAPv3, and the rules by which attribute values of these syntaxes are represented as octet strings for transmission in the LDAP protocol. The syntaxes defined in this document are referenced by this and other documents that define attribute types. This document also defines the set of attribute types which LDAP servers should support.

This document defines the framework for developing schemas for directories accessible via the Lightweight Directory Access Protocol. Schema is the collection of attribute type definitions, object class definitions and other information which a server uses to determine how to match a filter or attribute value assertion (in a compare operation) against the attributes of an entry, and whether to permit add and modify operations.

Syntaxes

This section of the RFC defines general requirements for LDAP attribute value syntax encodings. All documents defining attribute syntax encodings for use with LDAP are expected to conform to these requirements. The encoding rules defined for a given attribute syntax must produce octet strings. To the greatest extent possible, encoded octet strings should be usable in their native encoded form for display purposes. In particular, encoding rules for attribute syntaxes defining non-binary values should produce strings that can be displayed with little or no translation by clients implementing LDAP.

There are a few cases (e.g. audio) however, when it is not sensible to produce a printable representation, and clients MUST NOT assume that an unrecognized syntax is a string representation. In encodings where an arbitrary string, not a Distinguished Name, is used as part of a larger production, and other than as part of a Distinguished Name, a backslash quoting mechanism is used to escape the following separator symbol character (such as "", "$" or "#") if it should occur in that string. The backslash is followed by a pair of hexadecimal digits representing the next character. A backslash itself in the string which forms part of a larger syntax is always transmitted as '\5C' or '\5c'. Syntaxes are also defined for matching rules whose assertion value syntax is different from the attribute value syntax.

Binary Transfer of Values

This encoding format is used if the binary encoding is requested by the client for an attribute, or if the attribute syntax name is "1.3.6.1.4.1.1466.115.121.1.5". The contents of the LDAP

AttributeValue or AssertionValue field is a BER-encoded instance of the attribute value or a matching rule assertion value ASN.1 data type as defined for use with X.500. (The first byte inside the OCTET STRING wrapper is a tag octet. However, the OCTET STRING is still encoded in primitive form.)

All servers MUST implement this form for both generating attribute values in search responses, and parsing attribute values in add, compare and modify requests, if the attribute type is recognized and the attribute syntax name is that of Binary. Clients which request that all attributes be returned from entries MUST be prepared to receive values in binary (e.g. userCertificate;binary), and SHOULD NOT simply display binary or unrecognized values to users.

Index

E

F

H

I

L

M

N

O

P

Q

R

S

T

V

W