



IMAP Tools User Guide

Version 1.23

October 03, 2016

The IMAP Tools User Guide is a guide to the set of tools from <http://www.athensfbc.com> for working with e-mail folders and messages using the IMAP protocol. The guide describes how to obtain, configure, and use the IMAP Tools.

Table of Contents

1. ***Introduction***
2. ***List of the IMAP Tools***
3. ***Frequently Asked Questions (FAQ)***
4. ***Using the IMAP Tools***
 1. *General information*
 2. *imapcopy.pl*
 3. *migratelMAP.pl*
 4. *pop3tolMAP.pl*
 5. *imapsync.pl*
 6. *imapfilter.pl*
 7. *imapdump.pl*
 8. *dumptolMAP.pl*
 9. *imap_to_maildir.pl*
 10. *maildir_to_imap.pl*
 11. *IMAPtoMbox.pl*
 12. *MboxtolMAP.pl*
 13. *delIMAPdups.pl*
 14. *mbxIMAPsync.pl*
 15. *purgeMbx.pl*
 16. *thunderbird_to_imap.pl*
 17. *trash.pl*
 18. *delete_imap_mailboxes.pl*
 19. *list_imap_folders.pl*
 20. *imap_audit.pl*
 21. *imapPing.pl*
 23. *imapCapability*
 24. *list_account_sizes.pl*
 24. *imap_search.pl*
5. ***Change History.***
6. ***Troubleshooting Notes***

7. *Notes on Gmail*
8. *imapcopy Web Interface*
9. *OAUTH2 logins*

1. Introduction

What are the **IMAP Tools**? It is a set of Perl programs for use with IMAP servers. They enable you to do many things including the following:

- Copying a user's folders and messages to a new server
- Bulk migration of hundreds or thousands of users to a new hosting server
- Moving messages between folders based on a set of rules
- Backing up and restoring IMAP accounts
- Synchronizing accounts on two different IMAP servers
- Purging duplicate messages
- Loading messages from Maildir and Mbox servers into IMAP and vice-versa
- and many others

2. List of the IMAP Tools

The IMAP Tool set consists of the following scripts:

Name	Description	Section
imapcopy.pl	Copies messages and mailboxes from one IMAP server to another.	4.2
migrateIMAP.pl	Mass migration of messages for a set of users from one IMAP server to another.	4.3
pop3toimap.pl	Copies POP3 messages to an IMAP server.	4.4
imapsync.pl	Synchronizes accounts on two different servers	4.5
imapfilter.pl	Moves messages from one IMAP mailbox to another based on a set of rules.	4.6
imapdump.pl	Writes IMAP messages to local files.	4.7
dumptoIMAP.pl	Loads messages from imapdump.pl to a mailbox on an IMAP server.	4.8
imap_to_maildir.pl	Copies messages from an IMAP server to a user's Maildir.	4.9
maildir_to_imap.pl	Copies messages from Maildir to IMAP server.	4.10
IMAPtoMbox.pl	Copies messages from IMAP server to Mbox format.	4.11
MboxtoIMAP.pl	Copies messages from Mbox format to IMAP server.	4.12
delIMAPdups.pl	Deletes duplicate IMAP messages.	4.13
mbxIMAPsync.pl	Synchronizes mbx and IMAP accounts.	4.14
purgeMbx.pl	Purges an IMAP mailbox.	4.15
Thunderbird_to_imap.pl	Copies Thunderbird messages to an IMAP server.	4.16

trash.pl	Moves messages marked "Deleted" to trash mailbox.	4.17
delete_imap_mailboxes.pl	Deletes one or more mailboxes.	4.18
list_imap_folders.pl	Displays a list of the user's folders.	4.19
imap_audit.pl	Compares the messages in a user's account on one IMAP server with another.	4.20
imapPing.pl	Reports how long it takes to get a response from an IMAP server	4.21
imapCapability.pl	Lists which services an IMAP service supports.	4.22
list_account_sizes.pl	Builds a report on the size of each user's account	4.23
search_imap.pl	Searches messages for specified keywords and values	4.24

3. Frequently Asked Questions (FAQ)

1. What are the IMAP Tools?
2. Where can I get the IMAP Tools?
3. What is IMAP?
4. What operating systems do they run on?
5. What language are they written in?
6. OK, where can I get Perl?
7. What do I get with the IMAP Tools?
8. Which E-Mail servers are supported? What about GMail
9. Can I see an example?
10. Is SSL supported?
11. Is there a mailing list ?
12. Got a question not answered in this document?

1. What are the IMAP Tools?

It is a set of programs which make it easy for you to do a lot of things such as copying a user's messages from one server to another, synchronizing accounts on two servers, mass-migrating hundreds or thousands of users, purging duplicate messages, backing up and restoring IMAP accounts, moving messages based on a set of rules you define, loading messages from Maildir and Mbox servers into IMAP, and many other things.

2. Where can I get the IMAP Tools?

Go to <http://www.athensfbc.com/imap-tools>.

3. What is IMAP?

IMAP is a protocol for accessing e-mail folders and messages on a server. See [IMAP](#) for details.

4. What operating systems do the IMAP tools run on?

The IMAP tools will run on any server or PC on which a Perl interpreter is installed. Linux (all of the various flavors), Unix, Windows, and Apple are all supported.

5. OK, where can I get Perl?

One source is ActiveState which offers both free and commercial editions of their ActiveState Perl product for Windows, Unix/Linux, and Mac OS X. See <http://www.activestate.com/activeperl/downloads>.

6. What language are the tools written in?

Perl.

7. What do I get when I order the IMAP Tools set?.

When you place an order you will receive an e-mail message with an attachment containing the full set of tools (or the individual tool you ordered). You get support for questions, problems, and advice from the developer along with upgrades when new versions are released, bug fixes, and enhancements.

8. Which E-mail servers are supported? What about Gmail?

Virtually all e-mail servers are supported by IMAP Tools. IMAP Tools will work correctly with any server which supports the IMAP4 protocol. Gmail is definitely one of those although it has certain characteristics which make it unique. See Section 7 ("*Notes on Gmail*").

10. Is SSL supported?

Yes. You'll need it to talk to any server which does not support in-the-clear connections, such as Gmail which accepts only SSL connections. Please see the notes on SSL operations later in Section 4.1.4 ("*SSL Mode*").

11. Is there a mailing list for IMAP Tools?

Yes. To subscribe send a message to imap-tools-join@athensfbc.com or click here: [Subscribe to IMAP-Tools mailing list](#). To read the list archives you can go to <http://athensfbc.com/pipermail/imap-tools> athensfbc.com.

12. Got a question?

Send your question to rf9999@earthlink.net or post it to the IMAP Tools mailing list. If you have a question please don't hesitate to ask.

4. Using the IMAP Tools

This section provides instructions on how to use the IMAP Tools, including how to configure them for basic operation as well as describing the advanced options.

4.1. General information

4.1.1 Passwords

In order to access a user's account through IMAP you must supply two pieces of information: his username and his password. If you are going to copy messages between accounts on two different servers that means you need to provide the username/password on the Source server and the username/password on the Destination server. However many of the IMAP tools support an Admin mode where you only have to provide an authorized Admin's password to gain access to user accounts. Please refer to *4.1.6 Administrator Mode* for more information.

4.1.2 Mailboxes versus Folders

The IMAP RFCs prefer the term mailbox over folder. Each collection of messages is referred to as a mailbox, such as Inbox, Trash, Drafts, etc rather than as a folder. However, many people think of them as folders and the entire set of folders as a mailbox. This document uses mailbox and folder interchangeably.

4.1.3 Specifying the Port number

The default IMAP port is 143 but you can override it by including the port number in the hostname or IP. For example: `myhost:1430` will cause the IMAP tools to connect to myhost on port 1430 instead of port 143.

4.1.4 SSL Mode

Most of these utilities support SSL. SSL support is implemented using [openSSL](#) and the [IO::Socket::SSL](#) Perl module. Both must be installed on the system where you are running these tools. The site at <http://gnuwin32.sourceforge.net/packages/openssl.htm> has proved helpful in getting SSL working.

To put any of the IMAP tools into SSL mode (as required by Gmail for example) the hostname should include the SSL port number (993). For instance:

```
-S imap.gmail.com:993/<user>/<password>
```

This will cause the tool to make an IMAP connection over SSL connection on port to Gmail.

If you specify port 993 (995 for POP3) then an SSL connection is initiated. If the port number is 143 (110 for POP3) then it will try a non-SSL connection. With any other value the port will be tested to see if it supports SSL. If so, SSL will be used to make the connection; otherwise a non-SSL connection will be made.

4.1.5 CRAM-MD5 login

Many of the tools support the [CRAM-MD5](#) login method. To invoke a CRAM-MD5 login append /CRAM-MD5 after the host/username/password string. For example:

```
-s host/john/welcome/CRAM-MD5
```

4.1.6 Administrator Mode

Generally you must supply a username and password for each user in order to access his account. When migrating lots of users this can be inconvenient to say the least.

Some of the IMAP tools, such as `imapcopy`, `migrateIMAP`, `imapfilter`, and `imapsync`, support an “admin” mode where you only have to provide a valid administrator username and password to access users’ accounts.

These IMAP tools use the AUTHENTICATION=PLAIN login method to provide admin access. The advantage this provides is that you don't have to supply the user's password, just the name and password of an administrator's account which is authorized to assume a user's identify. This is particularly useful when doing mass migrations.

Please see the configuration notes on Administrator Mode for the individual tool you want to use.

Not all servers support AUTHENTICATION = PLAIN and of those that do some support it only over a secure connection (eg SSL).

4.2 imapcopy.pl

imapcopy is used for copying a user's folders and messages from one IMAP server to another. The folder structure, the IMAP flags (such READ/UNREAD status), and the original date of messages are preserved.

imapcopy logs into the “source” server with the user’s name & password, builds a list of the folders and the messages in them, logs into the “destination” server with the user’s name & password on that machine, and then copies the folders and messages from the source to it.

Here is an example of a simple imapcopy run:

Usage: `imapcopy.pl -S host1/user1/password1 -D host2/user2/password2`

```
imapcopy.pl starting  
  
Connected to host1 on port 143  
Connected to host2 on port 143  
Number of mailboxes to process: 7  
  Copied 60 messages to Big-Box  
  Copied 31 messages to Drafts  
  Copied 6025 messages to INBOX  
  Copied 32 messages to NOTES/2010  
  Copied 900 messages to NOTES/2011  
  Copied 525 messages to NOTES/2012/DRAFTS  
  Copied 633 messages to Sent  
Copied 9226 total messages  
Done
```

There are a number of optional arguments which can be used to control how imapcopy works.

Optional Command-line Arguments

Debugging options

When problems are suspected the best way to troubleshoot them is to enable a logfile and turn on debugging. for which there are 3 arguments:

```
-L <logfile>  
-d          (debug mode)  
-v          (verbose, displays info about the copying process)  
-I          (logs IMAP commands and responses)
```

IMAP traffic can be difficult to interpret if you are unfamiliar with the IMAP protocol but it is absolutely essential to figuring out certain problems. If you are reporting a problem to the IMAP Tools developer turning on IMAP logging is going to be the first thing he requests. ☺

Controlling which folders are copied

By default imapcopy will copy all folders on the source to the destination. But you can select a single folder or a subset of the folders with the `-m <mailbox list>` argument.

-m <mailbox list> (for example, "Inbox,Drafts,Sales 2010")
-R include submailboxes when used with -m

The `-e <mailbox list>` permits you to exclude one or more folders from the copy process. Like `-m` the `-e` argument accepts a comma-delimited list of folder names.

-e <mailbox list> (for example, "Trash,Wastebasket")

`-g <mailbox list>` works the same way as `-e` except that the mailbox names on the list are evaluated as regular expressions. Where `-e` expects exact matches (including the full path to subfolders, eg `Sales/old/2011`) `-g` permits you to do something like `-g '^Sales/old'`. Make sure you put single quotes around the value of `-g` to prevent the operating system from interpreting the regular expressions itself.

The folder names are case sensitive meaning the names and the include and exclude lists must match in upper/lower case.

Update Mode

Normally `imapcopy` will copy every message on the source to the destination. This means if you run it more than once you will wind up with duplicate messages on the destination. To prevent this you can put `imapcopy` into Update mode where it only copies a message from the source if it does not already exist on the destination. In other words update mode makes it safe to run `imapcopy` multiple times without the risk of duplicated messages.

To run in update mode add the `-U` argument to the command line. If you want to remove messages from the destination which no longer exist on the source then the `-s` argument will do this.

Changing folder names

`imapcopy` usually creates folders on the destination with the same names as on the source. However, there are cases where this may not be desirable such as often happens with the **Sent** folder. On many servers the folder is named **Sent Mail** or **Sent Items**.

`imapcopy` provides a way to rename folders on the fly during the copy process using a mailbox mapping file. Here's what it looks like:

```
# Mailbox map file for imapcopy. This file is used to change the name of a mailbox
# when it is copied from the source system to the destination system.
#
# The format is SRC MAILBOX: DST MAILBOX
#
# Nested mailboxes must be specified as a series of terms with '/' separating
# each mailbox. For example: OLDMAIL/NOTES/2007: MAIL/OLD_NOTES/2007.
#
# NOTE: Because IMAP mailboxes which contain National characters (eg non-ASCII)
# must be encoded in the Modified UTF-7 character set it is required
# that the Unicode::IMAPUTF7 Perl module be installed. This can be obtained
# from the CPAN archive or the ActiveState Perl site.

Sent: Sent Items
Trash: Wastebasket
```

To use the mailbox mapping feature add -M <file of mailbox name mappings> to the command line.

Copying Messages by Date(s)

imapcopy copies all messages regardless of their dates. However, if you want to copy only messages before, after, or in a date range, imapcopy provides a way to do it.

The -a <date> argument directs imapcopy to copy only those messages after the indicated date. In the same fashion -b <date> says to copy only the ones prior to that date. The format of those arguments is:

-a <DD-MMM-YYYY> for example, 01-MAR-2012
-b <DD-MMM-YYYY> for example, 31-DEC-2011

They can be used together to specify a date range. For instance, -a 01-JAN-2012 -b 31-DEC-2013 will copy only messages from 2013.

Parallel Mode

For higher performance you can run imapcopy in Parallel mode where several child processes will be created to split the work of copying folders among themselves.

Use the -Y <n> argument to select how many child processes imapcopy should employ. When choosing the number please keep in mind that as the number increases so does contention between the children for resources. The optimum number depends on a number of factors so it's a good idea to do some testing with various numbers to arrive at the one that works best for you.

Parallel mode requires the Perl Parallel::ForkManager module to be installed on your server. Parallel::ForkManager is available from CPAN.

Note: Parallel mode is not available on Windows servers since that operating system does not support the fork() feature.

Administrator Access

imapcopy supports *AUTHENTICATION=PLAIN* logins which permits you to avoid using the user's password. Instead of the user's password you supply an authorized administrator's username and password. Instead of *host/user/password* you would use *host/user:admin_user/admin_password*. For example:

"-S (or -D) host/tom:postmaster/mysecret".

XOAUTH2 tokens

To indicate that the password is actually an XOAUTH2 token prefix it with **oauth2:**. For

example: -S host/user/oauth2:password

CRAM-MD5 login

If imapcopy is to do a CRAM-MD5 login then add **/CRAM-MD5** to the -S or -D argument as shown here: -D host/user/password/**CRAM-MD5**

Other Command-Line Arguments

- d debug
- I show IMAP protocol exchanges
- L logfile
- m mailbox list (eg "Inbox, Drafts, Notes". Default is all mailboxes)
- R include submailboxes when used with -m
- e exclude mailbox list (using exact matches)
- g exclude mailbox list (using regular expressions)
- C remove msgs from source mbx after copying
- p <mailbox> put copied mailboxes under a root mailbox
- A <mailbox> copy to local mailbox from scrmbx
- o <mailbox> put all messages in this mbx on the destination
- x <mbx delimiter [mbx prefix]> source (eg, -x '. INBOX.'
- y <mbx delimiter [mbx prefix]> destination
- i initialize mailbox (remove existing messages first)
- M <file> mailbox map file. Maps src mbxs to dst mbxs. Each line in the file should be 'src mbx:dst mbx'
- q quiet mode (still writes to the logfile)
- t <timeout in seconds>
- T copy custom flags (eg, \$Label1,\$MDNSent,etc)
- a <DD-MMM-YYYY> copy only messages after this date
- b <DD-MMM-YYYY> copy only messages before this date
- X <megabytes> Skip any message exceeding this size
- U update mode, don't copy messages that already exist
- s In update mode delete messages from the destination which don't exist on the source
- B <msgnum> Starting point for message fetch
- E <msgnum> Ending point for message fetch
- u Don't copy unread (Unseen) messages
- H copy message headers only
- Z <directory> Record the msgids for copied messages in a DBM file at this location (eg -Z </var/tmp>). Used to prevent copying dups messages.
- j <n> Display count every <n> msgs
- z use an IMAP SEARCH filter expression to select messages (eg "OR (SINCE 01-mar-2001 SEEN)(before 01-mar-2001 UNSEEN)"
- G source is Gmail, strip the '[Gmail]' prefix from mailbox names
- c destination is Cyrus, fix the line terminator characters
- Y <n> number of processes to run in parallel
- f In Update mode remove messages from the source if they exist on the destination
- l Duplicate messages on the source will not be copied
- N Copy Gmail '[Gmail]/All Mail' mailbox (normally skipped)
- v Mark copied messages as SEEN (READ)
- P Don't copy source msgs which are marked for Delete

Web Browser Interface

In addition to executing imapcopy as a command-line tool it can be invoked through a Web browser. Please refer to Section 8 for the details.

4.3 migrateIMAP.pl

migrateIMAP is similar in functionality to *impcopy* except that the latter is used to copy a single user account whereas *migrateIMAP* is designed for bulk migration, eg to do a mass migration of dozens, hundreds, or even thousands of accounts in one pass. Faster migration is obtained by forking a number of child processes to migrate the users in parallel instead of doing one at a time.

The ***maxChildren*** argument (-n) indicates how many child processes will be executed in parallel (the default is 2). The optimum number of children depends on the system's resources but usually 4 or 5 maximizes the performance. Using more than that causes too much contention for resources and diminishing

Note: On Windows migrateIMAP limits the number of processes to one since Windows does not support the fork() system call necessary to run multiple, simultaneous child processes for higher performance.

The -i <file> argument tells migrateIMAP to read a file of users and passwords with each line as *source_user:source_password:destination_user:destination_password*. For example:

```
john:mypass:john:mypass
mary:secret:mary_jones:secret
don.smith:123ddA:don.smith:123ddA
etc
```

If a username or password contains a colon then you must escape it like this: my\:password.

To execute migrateIMAP use the following command:

```
migrateIMAP.pl -S host1 -D host2 -i <users file>
```

Update Mode

When migrating a large number of users it may be necessary or desirable to run migrateIMAP multiple times. For example, you might want to do a “pre-migration” in advance of the cutover to a new server. That will reduce the number of messages that have to be copied when the cutover day arrive because most will have been migrated earlier.

That’s where ***Update Mode*** comes in. If you run in update mode (-U) migrateIMAP only copies messages which do not exist on the destination. So when you do the final migration the number of messages to be copied is much smaller than it would otherwise have been. If desired, you can even do one last migration after the users have been moved over to the new server to pick any straggler messages that came in to the old server while the DNS changes are taking effect. With that in mind it’s usually a good idea prior to the migration to set the DNS TTL (Time-to-live) parameter to a low value so as to speed up the process.

Note that in update mode messages which exist on the destination but not the source are not deleted from the destination. There are situations where you might be using migrateIMAP to copy messages to the destination but if messages are being delivered to the destination by the local mail server you would not want deletions to be done. If you do want them to be deleted add the `-X` argument. That tells migrateIMAP to remove such messages.

(migrateIMAP uses the Message-ID to determine if a particular message already exists on the destination.)

Optional arguments:

```
-n      maxChildren (if omitted it defaults to 2)
-d      debug
-I      log IMAP commands and responses
-a      <DD-MMM-YYYY> Migrate only messages after this date.
-b      <DD-MMM-YYYY> Migrate only messages before this date.
-E      <mbx1,mbx2,...mbxn> Exclude these mailboxes (exact matches)
-R      <mbx1,mbx2,...mbxn> Exclude these mailboxes (using regular expressions)
-x      <delimiter [prefix]> Specify the src mbx delimiter and prefix (if server does not support NAMESPACE
command). For example:
-y      <delimiter [prefix]> Specify the dst mbx delimiter and prefix (if server does not support NAMESPACE
command)
-M      <file of mailbox name mappings>. See description of imapcopy.pl for details.
-L      logfile
-U      (update mode)
-u      Copy only Unseen (unread) messages
-o      Copy only Seen (read) messages
-X      (delete msgs from the destination that aren't on the source – Update mode)
-Y      Delete message from source after it has been successfully copied to the dest
-j      source Groupwise key <trustedAppName> <trustedAppKey>
-g      destin Groupwise key <trustedAppName> <trustedAppKey>
-e      sourceAdminName:sourceAdminPassword
-f      destAdminName:destAdminPassword
-w      destination is an Exchange server with a 9-mailbox-creation-per-IMAP-session
-G      Source user passwords are XOAUTH2 tokens
-J      Destination user passwords are XOAUTH2 tokens
-s      <size in MB>. Don't copy messages larger than this size
-A      <source Kerio master password>
-C      <destination Kerio master password>
```

To use CRAM-MD5 for logins add CRAM-MD5 to the host name like this: `-S host/CRAM-MD5`

Administrator Mode

Like imapcopy migrateIMAP supports AUTHENTICATION=PLAIN which permits you to supply an administrator name and password instead of having to provide each user's password. To put migrateIMAP into admin mode supply the source and destination Admin usernames and passwords using the `-e` and `-f` command line arguments:

```
-e sourceAdminName:sourceAdminPassword
-f destinationAdminName:destinationAdminPassword
```

In the users files leave the source and destination passwords null:

```
Rick::Rick:
Mary::Mary:
Donald::Donald.Smith:
James.Thomas::James.Thomas
```

XOAUTH2 Tokens

migrateIMAP supports the AUTHENTICATION XOAUTH2 login method. Please see https://developers.google.com/gmail/xoauth2_protocol and <https://developers.google.com/accounts/docs/OAuth2Login> for the details on how to create a project, register it, and create client credentials to make use of tokens for accessing Gmail accounts.

GroupWise Admin access

migrateIMAP supports administrator access to user accounts using GroupWise Trusted Application keys. This eliminates the need to supply user passwords since a Trusted Application key provides the application with access to user accounts without having to use passwords. Two arguments are used to supply migrateIMAP with the key information, one if the source server is GroupWise and the other for a GroupWise destination server.

```
-j source Groupwise key "<trustedAppName> <trustedAppKey>"
-g destin Groupwise key "<trustedAppName> <trustedAppKey>"
```

For example: `-g "trustedapp F8D11810E84095454544350AA9F8D11710048519C1CF4"`

4.4 pop3toimap.pl

pop3toimap.pl is used to copy a user's messages from a POP3 server to an IMAP server.

pop3toimap.pl makes a POP connection to the POP host and logs in with the user's name and password. Then it makes an IMAP connection to the IMAP host and logs in with the user's IMAP username and password. pop3toimap.pl fetches each message from the Inbox on the POP server and copies it to the user's Inbox on the IMAP server.

The usernames and passwords are supplied via a text file named in the -u argument. The format of the file of users is:

```
popUsername password imapUsername password
```

Usage: `pop3toimap.pl -p POP3host -i IMAPhost -u input_file`

Optional arguments:

```
-u file of users (popUser popPwd imapUser imapPwd on each line)
-n notify e-mail address
-r delete POP3 message after copying to IMAP
-t timeout in seconds
-m <mailbox> Default INBOX. Specify as mbx1/mbx2/mbx3.
-d debug mode
-I show IMAP commands and responses
-L <logfile>
-R <range of messages to copy, eg 200-333>
-A amin_user:admin_password
```

-t timeout (seconds)
-h print this message

You can also process a single user without having to create a file. Just supply the username and password on the command line using the -i and -p arguments as shown below:

-p pophost[:port]/popuser/poppwd -i imaphost[:port]/imapuser/imappwd

For example: *-p pop.server/rick/mypass -i imap.server.net/rick/secretpass*

4.5 imapsync.pl

imapsync is designed for synchronizing user accounts on a pair of IMAP servers called the "source" and "destination" hosts. The source account is the primary and the destination account is the secondary or mirrored account.

imapsync scans the account on the source host and does the following.

1. Adds to the destination any messages on the source which aren't on the destination
2. Deletes any messages from the destination which aren't on the source
3. Sets the message flags on the destination to match the source's flags

Please note that this is not two-way (bidirectional) synchronization. Messages added to the destination will not be copied to the source and messages deleted from the destination are not deleted from the source.

How imapsync works

1. Log into the source and destination hosts
2. Get a list of all mailboxes and messages on the source
3. Get a list of all mailboxes and messages on the destination
4. Compare the two lists
5. For each message on the source see if the message exists on the destination. If not, add it to the destination.
7. For each message on the destination see if the message exists on the source. If not, delete it from the destination.
8. For each message on the source compare the message flags with the message flags on the destination (eg \Deleted, \Answered, etc). If they don't match update the flags on the destination.

Usage: *imapsync -S sourcehost/username/password -D desthost/username/password*
Or

imapsync -S sourcehost -D desthost -u <users file>

users file format: *sourceUser:sourcePassword:destinationUser:destinationPassword*

Some notes:

- a. By default all of the user's mailboxes are processed but you can limit that by providing a list of mailboxes in which case only those will be examined (eg -m "Inbox, Drafts, Actions").

- a. The Message-Id is used for identifying messages in an IMAP account. A consequence of this is that if you have multiple copies of a message in a source mailbox only one copy will be reproduced on the destination host.

Other (optional) arguments are:

```
-L      <logfile>      (if omitted the output goes to STDOUT)
-u      <file of users>  Format srcUser:srcPwd:dstUser:dstPwd
-s      <DD-MM-YYYY or offset in days, eg 60> Include only messages since this date in the sync
-d      debug mode
-v      verbose mode
-q      Quiet mode. Display less details on the screen while logging everything to logfile
-I      log IMAP commands and responses
-m      <include mailbox list> eg, "Inbox,Drafts,Sales/2012"
-e      <exclude mailbox list>
-x      <delimiter [prefix]> Specify the src mbx delimiter and prefix (if server does not support NAMESPACE
command)
-y      <delimiter [prefix]> Specify the dst mbx delimiter and prefix (if server does not support NAMESPACE
command)
-n      Do not delete messages from the destination that do not exist on the source
-E      source administrator user and password
-F      destination administrator user and password
-t      Test mode. Do a "dry-run" and report what would have been done but don't actually do it.
-Y      use CRAM-MD5 login method on the source
-Z      use CRAM-MD5 login method on the destination
-R      collapse spaces in mbx names to 1 space (Gmail doesn't accept multiples
-W      wrap lines longer than 1,000 characters
```

Admin mode:

To activate admin mode where you use an admin user and password instead of the user's password, set the -E argument for a source user (-F for a destination user) and leave the user's password blank in the users file.

4.6 imapfilter.pl

imapfilter enables you to move selected messages to different folders based on a set of rules. For example you can create a rule which moves all messages whose Subject contains the phrase "Personal and Confidential". Any field in the message header can be used in a filter rule, for example, From, Subject, Date, In-Reply-To, etc.

The syntax of the rules file is **Header<tab>match<tab> source-mailbox<tab> destination-mbx**. Note that the fields must be separated by a <tab> character. Mailbox names are case-sensitive but filter criteria are not.

imapfilter supports the use of regular expressions in the matching rules. For instance, the following rule will move all messages from a sender whose name starts with "Jack" from the inbox to the wastebasket.

```
From ^Jack* Inbox Wastebasket
```

Some other examples of filter rules:

Subject test\$ Inbox Junk
Importance High junk Inbox
*From *abc.com Inbox Wastebasket*
X-Priority 3 Inbox Wastebasket
Size >50000 Inbox Big_Messages

Usage: imapfilter -S host/user/password -r <rules file> -L <logfile>

Optional arguments:

- E <admin_user:admin_pwd>
- I Log IMAP commands and responses
- c <number> the number of messages processed at a time (default 500)
- t test mode (don't actually move the messages)
- X <trash_mbx> expunge the Trash mbx, eg -X Trash

To use CRAM-MD5 for logins add /CRAM-MD5 like this:

-S server/user/password/CRAM-MD5

IMAP SEARCH syntax

Beginning in version 1.41 imapfilter supports the IMAP SEARCH syntax specified in RFC3501.

IMAP SEARCH provides a way to execute more sophisticated searches than was possible in previous versions which were restricted to a single keyword/value search like 'Subject "Test Message"'. For example:

OR (FROM george) (FROM sam) (FROM linda)
OR (SUBJECT Invoices) (TO someone@gmail.com)
NOT (TO john.smith)

When the prefix operator is omitted AND is implied:

(SUBJECT sanders) (SUBJECT guide)
(FROM test) (FROM george) (FROM smith)
(FROM john) (TO tom)

A number of other keywords and operators are available such as ones that work on dates like SENTSINCE, SENTBEFORE, SENTON.

To use the IMAP SEARCH syntax you must include the keyword ISEARCH as the first term in a rule. (This informs imapfilter that IMAP SEARCH syntax is being used) instead of the traditional imapfilter rule syntax. For instance:

ISEARCH "(from rick) (subject guide)" INBOX SEARCH_RESULTS

You can include a mix of IMAP SEARCH and traditional searches in the rules file:

```
Subject "[imap-tools\]\s*Fixed bug" Trash INBOX
Subject "requires approval" INBOX Trash
ISEARCH "(SUBJECT sanders) (SUBJECT guide)" INBOX OLD-DOCS
ISEARCH "(FROM test) (FROM george) (FROM smith)" INBOX SAVED_MSGS
```

 Note that the IMAP SEARCH function does not support regular expressions.

Please refer to **RFC3501 Section 9 ("Formal Syntax")** for the rules on composing a valid IMAP SEARCH. Following is an excerpt from the RFC.

```
search      = "SEARCH" [SP "CHARSET" SP astring] 1*(SP search-key)
              ; CHARSET argument to MUST be registered with IANA

search-key   = "ALL" / "ANSWERED" / "BCC" SP astring /
              "BEFORE" SP date / "BODY" SP astring /
              "CC" SP astring / "DELETED" / "FLAGGED" /
              "FROM" SP astring / "KEYWORD" SP flag-keyword /
              "NEW" / "OLD" / "ON" SP date / "RECENT" / "SEEN" /
              "SINCE" SP date / "SUBJECT" SP astring /
              "TEXT" SP astring / "TO" SP astring /
              "UNANSWERED" / "UNDELETED" / "UNFLAGGED" /
              "UNKEYWORD" SP flag-keyword / "UNSEEN" /
              ; Above this line were in [IMAP2]
              "DRAFT" / "HEADER" SP header-fld-name SP astring /
              "LARGER" SP number / "NOT" SP search-key /
              "OR" SP search-key SP search-key /
              "SENTBEFORE" SP date / "SENTON" SP date /
              "SENTSINCE" SP date / "SMALLER" SP number /
              "UID" SP sequence-set / "UNDRAFT" / sequence-set /
              "(" search-key *(SP search-key) ")"
```

If your search criteria does not comply with the syntax you'll get an error like this one and the rule will be ignored.

BAD command syntax error: NOT (XXXXTO jay.anderson)

Moving messages to a remote server

Normally the source and destination mailboxes are on the same server. However, you can configure imapfilter to put filtered messages into a mailbox on a remote server. In the following two examples the first rule moves messages to a local mailbox belonging to another user while the second moves messages to a remote server.

```
Subject *home* INBOX localhost:INBOX
Subject *testing* INBOX remotehost.xyz.com:INBOX
```

When configuring imapfilter to move messages to a remote server you must supply access control info in the rules file as shown below:

RemoteServer: localhost/user1/password1
RemoteServer: remotehost.xyz.com/user2/password2
RemoteServer: remote.acme.com:1430/user2/password2

Please note that IMAP SEARCH rules do not currently support moving messages to remote servers. This may change in future versions.

Global rules

Normally filter rules apply to individual mailboxes. However, you can specify a '*' character in place of the local mailbox name in a rule. This causes imapfilter to apply the rule to all mailboxes. For example:

*Subject *sales* * filtered_messages*

Date rules

imapfilter permits you to filter on dates which are earlier, later, or the same as a specified date. The date in the rules must be in RFC822 Mail date format:

DD MMM YYYY HH:MM:SS <GMT offset>

For example: 15 Dec 2009 18:00:05 +0500

Syntax of Date Rules:

>12 Nov 2009 12:45:10 +0500 selects messages whose date is more recent
<01 Jan 2013 08:14:00 +0500 selects messages whose date is earlier

You can also specify a relative date:

>30 selects message within the past 30 days
<30 selects messages more than 30 days old.

Some examples:

Date ">22 Dec 2008 15:00:00 +0000" INBOX MOVED
Date "<15 Jan 2009 00:00:00 +0500" INBOX MOVED
Date "=25 Dec 2009 08:00:00 +0500" INBOX MOVED
Date ">60" INBOX OLD
*Date "*2009*" INBOX 2009_messages*

In order to use the date comparison feature your Perl installation must have the following modules:

DateTime
DateTime::Format::Mail
DateTime::Format::DateParse Perl

Other command-line Arguments

```

-r      <rules file>
-L      <logfile>
-d      debug
-I      show IMAP commands and responses
-t      test mode, show what would have been done but don't move any messages
-f      first match. Apply the rule that matches first and ignore the others
-c      <number> Number of messages to process at a time. Default is 500.
-u      <file of users>, each line with user:password
-E      <admin_user:admin_password>, used with list of users.

```

The `-f` argument ("first match") causes `imapfilter` to honor the first rule that matches and to ignore any subsequent matches. Without `-f` the last matching rule is the one that gets applied.

4.7 imapdump.pl

`imapdump.pl` is used to export messages from an IMAP server and write them to files on the local server.

`imapdump` can be used for a number of purposes including getting a dump of the messages so that they be archived, or in order to extract information or attachments from them, or other things. One of the primary goals is to provide an easy way to backup an IMAP account and to provide the means of restoring one or all of the messages. To facilitate backup and restore the *`dumptoimap.pl`* script is designed to work with the files produced by `imapdump`. Please see Section 4.8 *`dumptoimap.pl`* for more information.

`imapdump` creates one file for each IMAP message using the IMAP message number as the name of the file to which it is written. For instance we can configure `imapdump.pl` to write the message files to `/var/imap_backup/rick`. Let's see how that looks.

```

/var/imap_backup/rick/INBOX
  1 2 3 4 5 .....
/var/imap_backup/rick/drafts/
  1 2
/var/imap_backup/rick/2002
  1 2 3 4 5 6 7 8 9 a b c d e f 10 11 ...
/var/imap_backup/rick/2003
  1 2 3
etc

```

`/var/imap_backup/rick/INBOX/1` would contain message number 1 from the INBOX;
`/var/imap_backup/rick/INBOX/2` would have the contents of message number 2, and so on.

Usage: `imapdump.pl -S host/user/password -f <local directory>`

"local directory" is the location where the messages are to be dumped, eg `/var/backups`

Optional command-line arguments

-i <file of users>
-g Extract attachments as separate files. Use with -x EML to set the file extension to EML
-n <number of simultaneous dump processes>
-F <flags> Dump only the messages with matching flags. <flags> is a comma-separated list of either standard or custom IMAP flags. Messages with flags matching any values in the list will be dumped.
-A admin_user:admin_password
-d debug mode
-l log IMAP commands and responses
-L <logfile>
-m <mailbox list> For example "Inbox,Drafts". Dump only the selected folders.
-x <extension>. Append an extension to the filenames, "txt" for instance
-a <DD-MMM-YYYY> Dump only the messages which arrived after this date
-r Remove messages from the IMAP server after dumping them.
-U Update mode, don't dump a message if it already exists in the dump directory
-k Remove messages from the dump if they no longer exist on the server
-u Don't dump a message if we have previously dumped it. Used with -D <dbm directory> which tells imapdump where to put a DBM directory database holding Message-IDs previously dumped.
-e <exclude mailbox list>. List of mbxs to skip, must be exact match on names
-E <exclude mailbox list>. List of mbxs to skip using regular expressions
-c Insert '\n' line termination characters
-s Flag SEEN msgs by appending 'S' to filenames (dump2imap will set SEEN flag when importing them).
-w sync flags
-C include custom (nonstandard) flags

To use CRAM-MD5 for logins add /CRAM-MD5 like this:

-S server/user/password/CRAM-MD5

4.8 dump2imap.pl

dump2imap.pl will load a set of RFC822 messages into an IMAP mailbox. The messages can be ones exported by imapdump or any other utility which produces messages in the correct format. dump2imap was designed as a companion script to imapdump. Used together imapdump and dump2imap provides a way to backup and restore a user's account.

Usage: ./dump2imap.pl -i host/username/password -D /work/dump/<user>/Inbox

The -D argument points to the directory where the message files are located. This directory should not have any files other than the messages. The optional -m argument can be used to limit the mailboxes to be loaded rather than to the list supplied rather than loading all of them. Mailboxes can be nested, /work/dump/rick/Sales/2013.

Optional Arguments

-m <list of mailboxes to load> For example -m "Inbox,Sales,Notes"
-A admin_user:admin_password
-d debug

- I log IMAP commands and responses
- m <mbx1,mbx2,...,mbxn> copy only the listed mailboxes
- x <extension> Import only files with this extension
- s Update mode. Copy only messages which do not exist in the IMAP mailbox

To use CRAM-MD5 for logins add /CRAM-MD5 like this:

-i server/user/password/CRAM-MD5

4.9 imap_to_maildir.pl

imap_to_maildir.pl is used to copy the messages in an IMAP account to a user's Maildir.

The **Maildir** e-mail format is a common way of storing e-mail messages, where each message is kept in a separate file with a unique name, and each folder is a directory. Please see <http://en.wikipedia.org/wiki/Maildir> for more information on Maildir.

Usage: imap_to_maildir.pl -S Host/User/Password -u <user> -M <maildir>

For example: ./imap_to_maildir.pl -S localhost/rich/pwd -u rick -M /users/rick/Maildir

The value of the `-u <user>` argument is used to set the ownership of the maildir created by `imap_to_maildir` with the *chown* function (Unix/Linux only).

Optional arguments:

- d debug mode
- I log IMAP commands and responses
- e <list of mailboxes to exclude> eg, "Inbox,Drafts,Trash"
- L <logfile>
- m mailbox list (eg "Inbox,Drafts,Notes". Default is all mailboxes)
- a <DD-MMM-YYYY> copy only messages after this date
- e <mailboxes to exclude> Regular expression, eg -e "\"Sales(.*)\"[Drafts|^Notes]\"";
- A <admin_user:admin_pwd>
- G source is Gmail; strip [Gmail] from folder names

To use CRAM-MD5 for logins add /CRAM-MD5 like this:

-S host/user/password/CRAM-MD5

4.10 maildir_to_imap.pl

maildir_to_imap.pl is used to copy the messages in a Maildir to an account on an IMAP server.

Usage: maildir_to_imap.pl -i <users list> -D <imapHost[:port]>

Each line in the user list must contain a pointer to the user's maildir, his IMAP username, and his IMAP password with the values separated by commas. For example:

Format: maildir,IMAP user,IMAP password

```
/mhub4/maildirs/jsampson@abc.net,jessi.sampson,secret
/mhub4/maildirs/rsmith@abc.net,rick.smith,wouldntyouliketoknow
/mhub4/maildirs/jjones@abc.net,jane.jones,nip123
```

Optional arguments:

-L logfile
-A admin_user:admin_password
-n <integer> (The number of child processes to run, default is 1)
-d debug
-I log IMAP commands and responses
-M <maildir_folder_name:IMAP_mailbox_name,...,maildir_folder_name:IMAP_mailbox_name>

To login with CRAM-MD5 add CRAM-MD5 like this:

-D imaphost/CRAM-MD5

The -M argument enables you to change the name of the maildir folder to something different in IMAP.

4.11 IMAP_to_Mbox.pl

IMAPtoMbox copies mailboxes and messages from an IMAP server to the local server (or PC) in the mbox format.

mbox is a generic term for a family of related file formats used for holding collections of messages. All messages in an mbox mailbox are concatenated and stored as plain text in a single file. The beginning of each message is indicated by a line whose first five characters consist of "From" followed by a space (the so named "From_ line" or "'From ' line" or simply "From line") and the return_path e-mail address. A blank line is appended to the end of each message. For a while, the mbox format was popular because text processing tools can be readily used on the plain text files used to store the e-mail messages.

For more information on mbox please see <http://en.wikipedia.org/wiki/Mbox>.

To run IMAPtoMbox you supply the name of the IMAP server, your username & password along with a pointer to the directory on the local host where the mailboxes are to be created. Each local mailbox is given the same name as the corresponding IMAP mailbox. For example:

Usage: IMAPtoMbox -i host/user/pwd -m mailbox_directory

For example: */IMAPtoMbox -i mail.abc.com/rick.sanders/mypassword -m /home/rfs*

The -i argument supplies the imap server name / the username / user's password and -m gives the local directory into which to create the mbox.

Optional arguments:

- d debug mode
- A admin_user:admin_password
- I Log IMAP commands and responses
- M <mbox list> Copy only select mailboxes, eg "Drafts,Trash,Notes"
- r Used with -M to do a recursive copy of all subfolders
- n Update mode, don't copy messages that already exist
- o <user> sets ownership of mailfile
- L logfile

To use CRAM-MD5 for the logins add this:
-i server/user/password/CRAM-MD5

4.12 MboxtoIMAP.pl

MboxtoIMAP.pl is used to copy the contents of Unix mailfiles to IMAP mailboxes. It parses the mailfiles into separate messages which are inserted into the corresponding IMAP mailbox.

mbox is a generic term for a family of related file formats used for holding collections of messages. All messages in an mbox mailbox are concatenated and stored as plain text in a single file. The beginning of each message is indicated by a line whose first five characters consist of "From" followed by a space (the so named "From_ line" or "From ' line" or simply "From line") and the return_path e-mail address. A blank line is appended to the end of each message. For a while, the mbox format was popular because text processing tools can be readily used on the plain text files used to store the e-mail messages.

For more information on mbox please see <http://en.wikipedia.org/wiki/Mbox>.

You can supply a directory name where one or more mailfiles reside or a complete filepath and the name of a single mailbox to be imported.

Usage: MboxtoIMAP.pl -m mailfiles -i server/user/pwd

For example:

```
MboxtoIMAP.pl -m /users/mboxes -i localhost/rick/pwd    (multiple mailboxes)
Or
MboxtoIMAP.pl -f /mhub4/mbox/00_toDo.mbx -i localhost/rick/pwd    (individual mailbox)
```

Optional arguments:

- f <mailfile> full filespec of individual mailfile
- A admin_user:admin_password
- n <name of IMAP mailbox> Used with -f argument

- U Update mode, don't copy messages if they already exist
- L <logfile>
- d debug mode
- I Log IMAP commands and responses
- r <range of messages to copy> for example, 1-500 or 25-75
- R remove messages from mailfile after copying to the IMAP server
- p <root mbx> put all mailboxes under the root mbx

To use CRAM-MD5 for the logins add this:
 -i server/user/password/CRAM-MD5

4.13 delIMAPdups.pl

delIMAPdups is used to remove duplicate messages. It identifies duplicates by finding messages that have the same Message-ID and removes all but one of them. If you prefer another field a different criteria to determine which messages are duplicates you may choose any header field such as Date, Subject, and so on. See the description of Optional arguments for details.

When a duplicate message is found the DELETED flag is set. If the -p argument has been supplied then an EXPUNGE operation is executed on the mailbox in which the message resides to remove it.

delIMAPdups does not check for duplicate copies of messages across multiple folders since it is often useful to cross-file messages.

Usage: delIMAPdups -S host/user/password

Optional Arguments:

- i <list of users and passwords> (format: user password)
- A admin_user:admin_password
- m <mailbox list> Only process the mailboxes in the list
- p delete the duplicates
- M <mailbox> Move duplicate msgs to this mailbox instead of deleting them.
- u include the date in determining if a message is unique, not just the message-id
- I log IMAP commands and responses
- H Use an MD5-digest hash of the message body instead of the Message-Id
- F <header field> Instead of the Message-Id use a different header field to determine uniqueness, eg Subject or From.
- L <logfile>
- d debug

4.14 mbxIMAPsync.pl

mbxIMAPsync is used to synchronize the contents of a Unix mailfile with an IMAP mailbox. The user supplies the location and name of the Unix mailbox (eg /var/mail/rfs) and the hostname, username, and password of the IMAP account along with the name of the IMAP mailbox. For example:

Usage: mbxIMAPsync.pl -f /var/mail/rfs -i host/user/pass -m INBOX

mbxIMAPsync compares the messages in the mailfile with those in the IMAP mailbox by Message-Id and adds the ones in the mailfile which are not in the IMAP mailbox. Then it looks for messages in the IMAP mailbox which are not in the mailfile and removes them from the IMAP mailbox.

Optional arguments:

-d debug
-L <logfile>

4.15 purgeMbx.pl

purgeMbx.pl deletes all of the messages in a user's IMAP mailbox. It requests a list of all the messages in the mailbox, sets the Deleted flag, and then expunges the mailbox to remove them.

Usage: ./purgeMbx.pl -s host/user/password -m mailbox

Note that the mailbox name is case-sensitive.

Optional arguments:

-d debug mode
-A admin_user:admin_password
-I log IMAP commands and responses
-L <logfile>

To use CRAM-MD5 for logins add /CRAM-MD5 like this:

-s server/user/password/CRAM-MD5\

4.16 thunderbird_to_imap.pl

thunderbird_to_imap copies messages from Thunderbird to an IMAP server. You supply the script with the name of the IMAP server along with your username and password. The -m argument is used to indicate the root directory where the Thunderbird folders reside on the PC. For example:

"/DOCUMENTS AND SETTINGS/RICK/APPLICATION
DATA/Thunderbird/Profiles/default/dlss3fw9.slt/Mail/mail.earthlink.net"

Usage: thunderbird_to_imap.pl -i host/username/password -m <thunderbird root>

Optional arguments:

-L <logfile>
-A admin_user:admin_password
-d debug mode
-M <folder list> eg folder1,folder2,etc. Copy only these folders
-E <exclude folder list> folder1,folder2,
-U update mode, copy only message which do not already exist in IMAP
-I log IMAP commands and responses
-W Thunderbird folders have ^M line terminators
-R <start:end> range of message numbers to copy
-r <root mailbox> Put folders under this root in IMAP

To use CRAM-MD5 for logins add /CRAM-MD5 like this:
-i server/user/password/CRAM-MD5

4.17 trash.pl

trash.pl checks a user's IMAP mailboxes for deleted messages which it moves to the trash mailbox. trash.pl opens a connection to an IMAP server, logs in with a username and password, searches the folders for messages marked for deletion, and moves them to the trash folder. The name of the trash folder is supplied with the -t argument.

If the -e argument is used then trash.pl empties the trash folder.

Usage: trash.pl -S server/user/pwd -t "Wastebasket"

Optional arguments:

-i <user list> (file of users and passwords, format "user password")
-a <admin_user:admin_pwd).
-m mailbox_list (eg "Inbox, Drafts, Notes". Default is all mailboxes)
-e empty the trash folder
-L <logfile>
-d debug

Example:

```
Checking mailboxes for deleted messages...
  Moved 2 messages from Drafts to Wastebasket
  Moved 4 messages from Sent Items to Wastebasket
  Moved 45 messages from INBOX to Wastebasket

51 messages were moved to Wastebasket
The Wastebasket mailbox has been emptied
```

4.18 delete_imap_mailboxes.pl

delete_imap_mailboxes is used to one or more mailboxes based on a regular expression supplied by the `-m` argument. For example:

```
-m "^thunder|^real|AA|^MOVED$|XXXX|zas"
```

delete_imap_mailboxes pulls a list of the user's mailboxes from the IMAP server and compares each one with the regular expression looking for matches. For each match it issues an IMAP delete <mailbox name> command to remove it.

It is advisable to do a test-run first to make sure that the mailboxes selected by the regular expression are the ones expected before actually deleting them. delete_imap_mailboxes lets you do this with the `-t` argument ("test" mode) by showing you which ones would have been deleted.

Some examples:

```
-m "books" will delete books, old_books, books2010
-m "^books" will delete all mailboxes starting with "books"
-m "2010$" will delete all mailboxes ending in 2010
```

You can combine the criteria into a single filter:

```
-m "books|2010$|Old folders|^My personal folder$"
```

Usage: delete_imap_mailboxes.pl -S host/user/password -m <regex>

Optional arguments:

-t	test mode (show what would have been deleted but don't delete it)
-A	admin_user:admin_password
-d	debug
-I	log IMAP commands and responses
-L	<logfile>

To login with CRAM-MD5 add /CRAM-MD5 like this:

```
-S host/user/password/CRAM-MD5
```

4.19 list_imap_folders.pl

list_imap_folders.pl will display a list of a user's folders.

Usage: list_imap_folders.pl -S host/user/mypassword [-O output file]

Example output:

```
AAA
```

BOGUS
Drafts
INBOX
INBOX/MOVED
INBOX/MOVED/Where
International

If you include the `-s` argument (“statistics”) then the number of messages and bytes in each folder will be included. For example:

AAA/FSM (29 msgs, 1.61 MB)
ADMIN (1 msgs, 15.96 MB)
Archive (5 msgs, 0.07 MB)
BOGUS (1542 msgs, 12.09 MB)
Big.Box (7 msgs, 0.06 MB)
Chats (1 msgs, 0.00 MB)
Deleted Messages (24 msgs, 1.33 MB)
Drafts (32 msgs, 0.90 MB)
FOLDER&NEW (0 msgs, 0.00 MB)
FSM (245 msgs, 12.51 MB)
Firmy (48 msgs, 1.49 MB)
Freelancer (15 msgs, 0.11 MB)
INBOX (9592 msgs, 434.87 MB)

Optional arguments:

`-s` Include count of messages and bytes in each folder
`-U <bytes>` Write a file messages larger than this size (large_msgs_report.txt)
`-I` Log IMAP commands and responses
`-O` <output file> Write the results to a file
`-A` admin_user:admin_password
`-l` <file of users> one user per line
`-B` Brief summary
`-N` Don't use IMAP QUOTA command
`-I` Log IMAP commands and responses

The users file consists of one user per line:

John:secret
Joan:password
Tom:password

If you supply an admin username and password then user passwords are not required.

4.20 imap_audit.pl

imap_audit can be used to compare the contents of a user's account on one IMAP server with an account on another IMAP server. It is often useful after migrating a user to another server to be able to check whether all messages were successfully copied.

For example:

```
imap_audit starting...
Mailbox EMC sms
  There are 32 messages in EMC sms on the source
  There are no missing messages on the destination
Mailbox Items
  There are 0 messages in Items on the source
  There are no missing messages on the destination
Mailbox Junk E-mail
  There are 0 messages in Junk E-mail on the source
  There are no missing messages on the destination
Mailbox INBOX
  There are 2420 messages in INBOX on the source
  There are no missing messages on the destination
Mailbox Artwork
  There are 9 messages in Artwork on the source
  There are no missing messages on the destination
[ ... ]

Total messages  9405
Missing messages 0
```

Usage: imap_audit.pl -S host1/user1/pwd1 -D host2/user2/pwd

Optional arguments

```
-d      debug
-I      Log MAP commands/responses
-L      <logfile>
-u      <file of users> format srcuser:srcpwd:dstuser:dstpwd
-E      source_admin_user:source_admin_password
-F      destination_admin_user:destination_admin_password
-m      <mailbox list> comma-separated list of mbxs to check
-n      Report message counts in each folder.
-Y      use CRAM-MD5 login method on the source
-Z      use CRAM-MD5 login method on the destination
-a      source Kerio master password
-c      destination Kerio master password
-B      Before date dd-mmm-yyyy
-A      After date dd-mm-yyyy
-u      <file of users> format srcuser:srcpwd:dstuser:dstpwd
```

4.21 imapPing.pl

imapPing is a tool to do a rough measurement of the response times from an IMAP server. It connects to the server and then performs some basic IMAP operations on a user's account and displays the time as each one is executed. The operations are:

1. Connect to the IMAP server
2. Log in with a username and password
3. Get a list of mailboxes
4. Select the INBOX

5. Get a list of messages in the INBOX
6. Log off the server

Usage: `imapPing.pl -h <host> -u <user> -p <password>`

To use CRAM-MD5 for logins add `/CRAM-MD5` like this:
`-h hostname/CRAM-MD5`

4.22 imapCapability.pl

`imapCapability` queries an IMAP server and reports which IMAP services it supports.

Usage: `imapCapability.pl -h <host> -u <username> -p <password>`

For example:

```
# ./imapCapability.pl -h localhost:1430 -u rfs9999 -p mysecret
```

The server supports the following IMAP capabilities:

```
IMAP4 IMAP4REV1 ACL NAMESPACE UIDPLUS IDLE LITERAL+ QUOTA
ID MULTIAPPEND LISTEXT CHILDREN BINARY LOGIN-REFERRALS
UNSELECT STARTTLS AUTH=LOGIN AUTH=PLAIN AUTH=CRAM-MD5
AUTH=DIGEST-MD5 AUTH=GSSAPI AUTH=MSN AUTH=NTLM
```

Optional arguments:

```
-d      debug
-m      list the user's folders
-A      admin_user:admin_password
```

To use CRAM-MD5 for logins add `/CRAM-MD5` like this:
`-i server/user/password/CRAM-MD5`

Please refer to the [IMAP RFCs](#) for an explanation of these capabilities.

4.23 list_account_sizes.pl

`list_account_sizes.pl` can be used as a pre-migration planning tool to determine the size of each account to be migrated as well as the total number of MBs which will be copied. It produces a report that looks like this:

```
SIZE (MB)    USER
=====
```


1,583.27	jessi
1,429.61	tom
1,260.20	john
691.07	jane
45.18	bob

Totals

=====

Users 5

Bytes 5,009.33 MB

The command to execute list_account_sizes.pl is:

```
./list_account_sizes.pl -S <host> -u <list of users> [-A admin:password]
```

If you supply the name and password for an administrator who can log into a user's account then the user list is just account names; otherwise you must supply the users's passwords in the list:

```
john:mypass
mary:herpassword
etc
```

4.24 imap_search.pl

imap_search.pl is used to search a user's mailbox for messages matching a set of search criteria. By default all folders are searched but you can limit the search to a single folder if desired.

For example, let's say you want to find all messages sent by john.lombard:

```
./imap_search.pl -S host/user/password I -s "from john.lombard@aol.com"
```

Searching for from "john.lombard@aol.com" in all folders

Folder	Date	Subject
Gmail	18 Aug 2012 09:31:09	property 13251984
INBOX	03 Dec 2013 11:10:01	subscribe test_list
INBOX	07 Sep 2012 01:11:05	property 33951800
INBOX	07 Sep 2012 01:12:25	RE: meeting tomorrow
TRASH	03 Dec 2013 18:10:01	hello

You can combine filter expressions as well:

```
OR from=tom from=charles
AND subject=test from=marie
"subject=sales meeting" NOT from=linda
from=Mary subject=notification
```

```
subject=password NOT subject=new
sentsince=10-Nov-2013
sentbefore=1-Jan-2001
on=5-Dec-2013
"body=agenda for today"
```

Note that the SEARCH command is not a very sophisticated command as specified by the IMAP RFCs so you will want to keep your searches fairly simple. If your search expression is not acceptable to IMAP you'll get a syntax error. After some testing you will find what works on your IMAP server (some fields aren't searchable on some servers while they work on others. It seems to be implementation-specific).

In some cases you have to insert 'header' to tell the IMAP server to search the ENVELOPE rather than the content header: header Received=xxxxx.

Usage: ./imap_search.pl
-S host/user/pwd
-s <IMAP SEARCH EXPRESSION>
[-m <folder list>] folder1, folder2, etc. Default is all folders
[-r] If list of folders supplied with -m then also get submailboxes under each
[-L <logfile>] default is imap_search.log
[-I] print IMAP commands and responses
[-d] debug mode

Add CRAM-MD5 for CRAM-MD5 logins like this: host/user/pwd/CRAM-MD5

Administrator mode

It is a pain having to supply a user's password with each search so imap_search supports admin mode where you can give the administrator's username and pwd instead. For example:
host/user:admin_user/admin_password

Single folder search

By default imap_search searches all folders but you may want to narrow the search to save time, for example by searching just the inbox. To limit the search place the name of the folder at the end of the access string like this:

```
host/user/password:Inbox
-or-
user:Drafts
```

Notes:

- Searches are case-insensitive
- Mailbox names are case-sensitive
- Regular expressions are not permitted
- Wildcards are implied, eg subject=test matches every subject which has the string "test" in it
- Search values containing spaces must be quoted: susubject="this is the day"
- "Body" searches take longer since the body of each message has to be searched

- See the IMAP RFC for a listing of all IMAP SEARCH keywords or visit:

http://www.afterlogic.com/mailbee-net/docs/MailBee.ImapMail.Imap.Search_overload_1.html

Administrator Login

migrateIMAP, imapcopy, imapsync, and imapfilter support AUTHENTICATION=PLAIN login. One advantage this provides is that you don't have to supply the user's password, just the name and password of an administrator's account which is authorized to assume a user's identity. This is particularly useful when doing mass migrations.

For imapcopy the syntax is "-S (or -D) host/user:admin_user/admin_password".

For migrateIMAP you have to supply the source admin name and password and/or the destination admin name and password:

```
-e sourceAdminName:sourceAdminPassword
-f destAdminName:destAdminPassword
```

(imapsync uses the -E and -F arguments rather than -e and -f).

In the file of usernames and password you must indicate whether to use AUTH PLAIN by leaving the user's password empty. For a normal login the line is:

```
src_user:src_pwd:dst_user:dst_password
```

But for a source admin login you would omit his password (and the same for the destination user if you want to use an admin login). Some examples:

```
src_user::dst_user:dst_password
src_user:src_password:dst_user:
src_user::dst_user:
```

Note that not all servers support AUTHENTICATION = PLAIN and of those that do some support it only over a secure connection (eg SSL).

5. Change History

When changes are made to the IMAP Tool set the Change history is updated at http://www.athensfbc.com/imap-tools/change_history.html. For example:

- 01/11/2013 - imapdump.pl. Added update mode (-U) which copies only messages that don't exist in the dump directory.
- 01/10/2013 - imapsync.pl. Added support for AUTHENTICATION=PLAIN.

- 01/05/2013 - migrateIMAP.pl. Added support for AUTHENTICATION=PLAIN.
- 01/04/2013 - imapcopy.pl. Added support for AUTHENTICATION=PLAIN. Among other benefits is the ability to use an administrator's ID to access a user's account. This avoids the need to supply the user's password, just the admin's password. The syntax is -S (or -D) host/user:admin_user/admin_password. See imapcopy.pl notes on main page for details.
- 01/03/2013 - imap_to_maildir.pl. Fixed a couple of bugs. The maildir filename was missing a dot before the hostname and the rfc822_size was parsed incorrectly when working with at least one IMAP server. Also fixed the -m argument. \
- 12/08/2012 - imapcopy.pl. Added the option to display a progress counter while copying messages. -j <1000> will log 'Copied 1000 of 14963 messages, Copied 2000 of 14963 messages', etc.
- 12/01/2012 - imap_audit.pl. New tool for comparing a newly migrated account. It compares the user's mailboxes on the old IMAP server with the ones on the new IMAP server. This can be used following a migration to check that everything was copied over.

5.1 Announcements of updates and new versions

A note about each change is also mailed to the IMAP Tools Mailing list. Subscribing to the mailing list is the easiest way to stay in touch with bug fixes, enhancements and the release of new versions of the tools.

6. Troubleshooting Notes

This section lists some hints on what to try when you run into problems.

6.1 *Can't connect to a server*

First determine whether you can make a basic network connection by pinging the host (*ping <host>*). If ping fails double-check the host or IP address to make sure it is correct. If it is right then there may be a firewall blocking the connection.

If ping works but the IMAP Tool can't access the host then the next step is to check whether a firewall might be blocking the port the connection is on. The standard IMAP port is 143 (110 for POP). The secure IMAP port is 993 (995 for POP). The easiest way to test the port is with the telnet utility. An example:

```
# telnet imap.server.net 143
Connected to imap.server.net
Escape character is '^'.
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE STARTTLS
AUTH=PLAIN] Dovecot ready.
```

If you don't get something like this check the firewall. If you are testing port 993 (or 995) with telnet be aware that the server will not send clear text like in the previous example (because they are SSL ports). Look instead for just 'Connected' as proof the port is not blocked.

6.2 IMAP Tracing and Debugging

The `-d` argument will log additional information about what the tool is doing which may be helpful in diagnosing problems.

Most of these scripts accept the `-I` argument which causes logging of the IMAP commands and responses between the script and the IMAP server. This can be useful when troubleshooting problems but you probably don't want to use it during normal operations because of the amount of data that gets written to the logfile.

For example:

```
11-08-2007.03:28:18 2114 >> 1 EXAMINE "Inbox"
11-08-2007.03:28:18 2114 << * FLAGS (\Answered \Flagged \Deleted \Seen \Draft $MDNSent $
Hidden $Media $Forwarded Junk $Label1 $Label2 $Label3)
11-08-2007.03:28:18 2114 << * OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Dra
ft $MDNSent $Hidden $Media $Forwarded Junk $Label1 $Label2 $Label3)] limited
11-08-2007.03:28:18 2114 << * 86 EXISTS
11-08-2007.03:28:18 2114 << * 0 RECENT
11-08-2007.03:28:18 2114 << * OK [UIDNEXT 360] predicted next UID
11-08-2007.03:28:18 2114 << * OK [UIDVALIDITY 298417783] UIDs valid
11-08-2007.03:28:18 2114 << 1 OK [READ-ONLY] EXAMINE completed
```

Be aware that with IMAP logging turned on usernames and passwords will be written to the logfile. So if you will be submitting the logfile to get help in diagnosing problems you will want to edit the logfile to remove passwords.

6.3 Reporting problems

You can report problems with the IMAP Tools either directly to rfs9999@earthlink.net or by posting a note in the IMAP Tools mailing list. Either way will result in the same prompt response from the IMAP Tools developer.

7. Notes on Gmail

Gmail has some special characteristics which are unique to it. This section contains information that may prove useful in migrating users to Gmail or just copying large numbers of messages.

7.1 Throttling

Gmail places limitations on how fast you can copy data to their servers using IMAP. If you are migrating a large number of messages to a Gmail account you may experience *throttling* (or rate-limiting). When Gmail starts throttling a connection the throughput may decrease until basically nothing is being transferred. Eventually Gmail will disconnect the IMAP connection. This is done to protect Gmail from being overloaded by IMAP migrations.

If you are running one of the IMAP tools and you have the -I argument set you'll know that throttling is in effect when Gmail sends this response: **OK SUCCESS [THROTTLING]**.

Here's what Google's Dan Hovey posted in November 2011 concerning Gmail throttling:

"We currently implement quotas for IMAP, which prevent abnormally intensive access to IMAP services by a single user account. Although most users and applications do not ever come close to these limits, they are necessary to ensure a good experience for our users."

Gmail cannot currently sustain permanent, intensive IMAP request rates for user accounts. This causes users to experience unresponsiveness on the web and on mobile devices. These issues are not easily traced back to individual applications. In order to mitigate these problems, we're beginning to throttle requests on a per-user basis."

"When your requests are throttled for a particular user, the IMAP response will include '[THROTTLED]'. In this case, your application should back off and perform fewer requests for that user."

Gmail starts throttling after a certain amount of IMAP traffic. The throttle is not a fixed amount; it increases as the IMAP connection continues to move messages. That means you might be getting 100 msgs/second and then 50 msgs/sec, and then 1 per second. At some point Gmail disconnects the session.

Another comment from Google regarding IMAP migration.

*"In general, we have an upload limit, but that limit should rise with the number of messages uploaded. I'm not sure how well that lock-step works, we haven't really tested it since **IMAP migrations aren't technically supported**."*

The following actions will trigger throttling by Gmail (this information is from 2011 and it may have changed since then).

- 1) 8 IMAP commands/second sustained over 1 hour
- 2) 2.5GB/download/24 hours plus extra capacity up to the user's quota that gets renewed every 2 months (it is designed to allow a user to download all of their mail 5 times in 2-3 months).

7.2 Folders versus Labels

In Gmail messages aren't in folders, they're in labels. Gmail puts all messages into a folder named "All Mail" and maps the messages to folders for the sake of IMAP using labels. To IMAP it appears that messages are in both All Mail and in individual folders but there is really only one copy and it is in All Mail.

For that reason imapcopy will ignore the '[Gmail]/All Mail' folder when copying messages from Gmail to another server because that would result in duplicate messages getting copied. It would also require copying twice the actual number of messages in the Gmail account and would be a waste of time and bandwidth.

7.3 No duplicate messages in Gmail

Because of the way that Gmail works it will not store multiple copies of the same message. Even if you run imapcopy several times you will find just a single copy of each message. In other words, there won't be any duplicate messages in a Gmail account.

8. imapcopy Web Browser Interface

Although primarily designed for execution on the command line imapcopy also can be invoked from a Web browser. This section describes how to configure imapcopy for that usage.

The web browser mode in imapcopy is designed for an Apache Webserver; it may work with other web servers but only Apache has been tested with imapcopy.

Installation and configuration

1. There are four files in browser mode: `imapcopy.pl`, `imapcopy.cgi`, `imapcopy.html`, and `imapcopy.cf`.
2. Copy ***imapcopy.cgi*** and ***imapcopy.cf*** to the cgi-bin directory on the Apache server and set the owner and permissions so they can be accessed by the Apache server process.
3. Copy the ***imapcopy.html*** form to the htdocs directory.
4. Copy ***imapcopy.pl*** to some location on the server. It does not have to be located in the cgi-bin directory (you may prefer it somewhere else if you use it in command-line mode as well). `imapcopy.cf` will tell `imapcopy.cgi` where to find it.
5. Edit `imapcopy.cf` and change the values if necessary:

```
IMAPCOPY: /usr/local/apache/cgi-bin/imapcopy.pl
LOGFILE: /usr/local/apache/cgi-bin/imapcopy.log
DEBUG: 0
SHOWIMAP: 0
```

6. If desired, you can modify the html form (`imapcopy.html`) to set the name of the source and destination servers, move the fields around, add help text, insert a company logo, etc. The default form looks like this:

IMAPCOPY

Source server	<input type="text"/>		
Destination server	<input type="text"/>		
Source username	<input type="text"/>	Source password	<input type="text"/>
Destination username	<input type="text"/>	Destination password	<input type="text"/>
Copy only these folders	<input type="text"/>	folder1, folder2, ...	
Exclude these folders	<input type="text"/>	folder1, folder2, ...	
After date	<input type="text"/>	DD-MM-YYYY	
Before Date	<input type="text"/>	DD-MM-YYYY	
Update Mode	<input type="checkbox"/>		
<input type="button" value="Submit"/> <input type="button" value="Clear form"/>			

After clicking on Submit the copy process will start. Depending on the size of your account it will take a few minutes or more to copy everything over. When it finishes you will receive an e-mail notifying of the results.

7. Create imapcopy.cf

imapcopy.cf is the configuration file for imapcopy in browser mode. It also permits the administrator to control the number of simultaneous instances of imapcopy permitted to execute.

```
# Config parameters for imapcopy browser app
#
IMAPCOPY: /usr/local/apache/cgi-bin/imapcopy.pl
LOGFILE: /usr/local/apache/cgi-bin/imapcopy.log
PROCESS_LIMIT: 8
DEBUG: 0
SHOWIMAP: 0
ARGUMENTS: -m Inbox,Drafts,Notes -U -j 500
```

PROCESS_LIMIT controls how many users can be copying their mail at any given time. It allows you to prevent imapcopy from overloading the IMAP server when too many users try to use it at the same time. When the limit is reached the next user will see this message in his browser:

The maximum number of IMAP copies is already running. Please try again later.

ARGUMENTS permits you to pass command-line arguments to imapcopy.pl. Any optional argument support by imapcopy.pl can be supplied using the ARGUMENTS keyword.

This completes the installation and configuration. To access imapcopy point your browser to `http:<server>/imapcopy.html` where <server> is the hostname or IP of the Apache server.

Notes:

- Ensure that the Apache process has read/write privilege to the location you want imapcopy to write its logfile.
- When the user clicks on "Submit" the imapcopy process is launched and ***Your copy job has been started. You will be notified when it has completed*** is displayed on the screen.
- When imapcopy completes copying the user's messages it inserts a notification in the inbox on the source and destination with a summary of the message counts. For example:

```
Subject: IMAP COPY completed at Sat Dec 7 19:12:11 2013
From: imapcopy <imapcopy>
Date: 7:12 PM
To: tomcat

Here is a summary of the job:
Completed at Sat Dec 7 19:12:11 2013

Msgs Copied      Folder
-----
42      I6AOQA9gD8-
32      2010 Sent
32      2010 Sent Mbx
8      AAAAAAAAAA
3      Archives
20     Archives.2010
```

8. OAUTH2 Logins

OAUTH2 enables an application to access a user's account without having to use a password. This is considered to be more secure than any method in which the user supplies his password. Google's policy is that all other access methods are "less secure" and they now require a user to enable "less secure" methods in their account settings or a password-based login will be disallowed.

Support for OAUTH2 logins was added to the IMAP Tools added support in version 1.300 which was released on 10/17/2014.

Please visit <http://en.wikipedia.org/wiki/OAuth> (Wikipedia) and https://developers.google.com/gmail/xoauth2_protocol (Gmail) for more information.

The article at <https://developers.google.com/accounts/docs/OAuth2Login> explains how to create a project in Gmail, register it, and create client credentials to make use of tokens for accessing Gmail accounts. As explained in that article a user must grant permission to the application before it can use tokens to access the user's account. Tokens are good only for a short period of time, typically one hour, but an application can issue a "refresh" command and get a fresh copy of the token.

Using OAUTH2 tokens with IMAP Tools

To use an OAUTH2 token with the IMAP Tools simply specify the token in place of the password with the prefix "oauth2:" which will tell the tool that it is a OAUTH2 token rather than a password.

For example, let's say that you have obtained a token ---
ya29.oQDDSxPm3ciwisd1cw3Mxb32WyoMUsbxa5l5TYfqjooMb7cPNtX6Chbl --- and
you are going to use it with imapcopy:

```
imapcopy.pl -S  
imap.gmail.com:993/username/oauth2:ya29.oQDDSxPm3ciwisd1cw3Mxb32WyoMUsbxa5l5T  
YfqjooMb7cPNtX6Chbl
```