

Implementing LIES (Location Independent Email Service)

Pamela Vogel
Brown University
Box 1885 - CIS
Providence, RI 02912
401-863-7228

Pamela_Vogel@brown.edu

John Spadaro
Brown University
Box 1885 - CIS
Providence, RI 02912
401-863-7558

John_Spadaro@brown.edu

ABSTRACT

This paper will provide a useful guide to those schools that have not yet implemented IMAP and are considering doing so. With both the systems and user services perspectives, any schools considering implementing this service will get a full view of what is involved in putting out IMAP.

The goal of the LIES (Location Independent Email Service) project was simple: add IMAP services to our existing POP services in a way that would allow people to work from anywhere with their settings, nicknames, and mailboxes intact. The team knew that even if all the goals were not met in the first stage of implementing the new service, it was critical to achieve the location independence capability quickly. Over 500 people were still using email on our mainframe, and the only way to migrate them off was to recreate the capability they had with their mainframe accounts, complete location independence, while giving them new services (such as the ability to read attachments). Furthermore, with a large number of the POP users making use of the "Leave mail on server" option, it was clear that even those users were looking for a better solution.

The project team was able to start with a server that already provided IMAP service because the server software had been replaced late in 1999 for Y2K reasons. System support still was needed to implement SSL support, quota management, and backup procedures. Client selection (both desktop and web based), new documentation, and marketing strategies were also components of the project.

Keywords

IMAP email rollout ACAP quota email clients.

1. INTRODUCTION

LIES turned out to be a good project name for our IMAP implementation. What we wanted could not be done within the

budget we had available to us. We did learn a lot along the way, though, and should the money suddenly show up, we know how to put out an IMAP service the way it should be done.

2. EMAIL SERVICES

Last fall, we were just beginning to think about implementing an IMAP service. Our email service project team had been meeting regularly during the spring and summer to determine priorities in our email services. Our first priority was upgrading the mail server; it was about 8 years old, and was a potential Y2K problem. We decided in upgrading that we needed to find a server that supported both POP and IMAP so that we would be able to move to IMAP without requiring another upgrade. Our systems staff had been working with freeware, supported by the net, for many years, and the manager of systems, John Spadaro, decided that this time, we were actually going to buy software to support.

2.1 Server Selection

We selected Messaging Direct's mail server, primarily because we were able to get modifications made to it to support some of our local peculiarities (hyphens and apostrophes in email addresses, and Kerberos for password security), and it met the technical requirements for our environment, including:

- Running on Solaris 2.6 and 2.7

- Ability to manage 15k users without needing to split onto multiple servers

- Support for both POP and IMAP, as well as standards support for extensions such as ACAP or IMSP

- Working with our existing backup mechanism (Solstice Backup/Legato Networker)

- Ease of initial setup, upgrades, and patch installations

- Scriptability for life cycle maintenance of mailboxes, for quota increases, and ACL management

- Appropriate handling of quota exceeded conditions

Other servers that we evaluated included:

- Cyrus from CMU

- The University of Washington IMAP server

- Innosoft's PMDF MessageStore

Because John felt strongly that we needed to move toward commercial products with technical support, we looked at the two

free options as last resorts. Had Messaging Direct not been willing to make the modifications necessary to support our mailbox names, we would have had to look more closely at those free options.

2.2 Dealing with Y2K

The upgrade to the Messaging Direct server took place over Thanksgiving weekend, with mail services turned off for about 36 hours. Because Messaging Direct is not an American company, but a Canadian one, that Thursday and Friday was no problem for them, and we felt very strongly that we wanted to take care of upgrading the server well in advance of the end of the year.

2.3 Enhancing Services

Once that problem was solved, we had to look at what else we could do in the short and long term. One large-scale project was started, LIES (Location Independent Email Service), and a few smaller scale projects were started as well. The first was removing a potential security hole where we allowed passwords to be sent over the network in clear text. The second was making our mail server more robust by enhancing the hardware platform and eventually having a failover machine designated should the server fail. Other items that were raised as possible email projects were developing voice/fax/email integration, signed and sealed email, and enabling of PDA access to our services. None of these were scheduled.

3. DEFINING THE IMAP SERVICE

3.1 Why LIES?

Our large-scale project, LIES, was intended to be more than just an IMAP service. In order to achieve real location independent email, we felt that it was important to have the capability to store nicknames and settings in a server location as well. There were two factors behind our desire to have true location independence. Students and others often work on more than one machine on a regular basis and keeping nicknames and settings up to date in that environment is difficult. We also have about 500 people still using our mainframe based email (RiceMail). We wanted a service that would replicate what they had on the mainframe (nicknames and settings with you where ever you go), while also providing an enhancement (capability to deal with attachments appropriately) to encourage them to migrate.

3.2 IMAP functions

As we started to think about how to work with IMAP, Pamela attended the Fall SIGUCCS meeting, and had an opportunity to talk with several schools that had already done it. From that we learned about three issues that were important to think about with IMAP: quotas, clients, and shared mailboxes. It was clear we had a lot of decisions to make.

3.2.1 Quotas

Quotas are more of an issue with IMAP services because potentially all of a person's email could be stored on the server. Take a population of 15,000, and you have a potential disaster of disk usage. The presentation by Pitt at that SIGUCCS indicated that they had implemented 10 meg quotas. Research on the web sites of other schools that had implemented IMAP indicates that

this quota size, or smaller, was the typical sizing. Some allowed you to buy more, some allowed you to apply for more, but, basically, the standard among schools was 10 megs or less.

Our project team had higher goals in mind. With mail services that allows a single mail message (with attachments) to be 10 megs, a quota of that size did not seem reasonable. Looking at our campus population, we felt that while undergraduates are typically at Brown for only 4 years, staff, faculty and graduate students can spend from 6 years to decades at the campus. To limit their mail store to just 10 megs was not going to address the problems we were trying to solve. Our campus customers wanted access not only to their new mail, but also to at least some (if not all) of the mail they had already received (and sent).

Our proposed quota limits were set at 25 megs for undergraduates and 50 megs for staff, faculty and graduate students. Each population could, from a web page, request their quota to be doubled, with no oversight. With those kinds of quotas, we were looking at disk requirements of a terabyte or more. That would require that we put in disk arrays and a new backup system, well above the costs that we had originally estimated for the project.

3.2.2 New Clients

Clients were another sticking point. We had supported Eudora as our mail client since we had first implemented POP services in 1991. Eudora supports Kerberos password protection, but we had always had an unadvertised clear text port that could be used by mail clients without Kerberos support. As more and more fully featured email clients appeared in the marketplace, many of our customers no longer used Eudora. Therefore, we wanted to *enable* a variety of email clients to work in a secure way, while still only *supporting* a single client.

3.2.3 Sharing Mailboxes

With over 300 departmental IDs on our server already, the features of shared mailboxes looked promising. As part of our security strategy, we wanted to move away from having accounts where multiple people knew and used a password; these departmental IDs were a problem area, and shared mailboxes seemed like a reasonable way to enable the functionality these departments were seeking while closing a security hole.

3.3 Policies and Process

During the spring and summer, the project team moved along through several issues. We set quota policies and process, evaluated clients based on an extensive criteria list, selected desktop clients, reviewed possible web clients, and priced out the disk and backup needs of the project.

Even before our fiscal year started, July 1, it was already clear that this project was not going to be able to go forward this fall as planned. Overall monetary needs in the department had grown (salaries needed to go up to attract employees during this period, for example), and the budget did not grow to accommodate those new expenses, or some large scale projects that needed to be undertaken. We continued to move forward on what we could work on, knowing that implementing IMAP was a high priority, and it was possible that money would be found to implement it.

4. CLIENT SELECTION

4.1 Desktop Clients

When it came time to think about what desktop clients we were going to recommend for this project, we had some useful sources of information. Two web sites, one evaluating Windows IMAP clients, and the other evaluating Mac based ones, provided useful information about reasonable clients for IMAP, as well as some important factors to consider in the evaluation. The site we used for Windows clients was at University of North Carolina (www.unc.edu/dande/eval/imap/index.html). The site that had evaluated Mac IMAP clients was at a company (www.tandb.com.au/email/clients/), but they did an excellent job of listing useful criteria for evaluation.

4.1.1 Criteria

Based on the information we found at those two sites, we developed our own client evaluation form. Some of the factors that we looked at in clients included:

Standards support: LDAP and/or Ph directory links, ACAP or IMSP for settings storage, security mechanisms for password protection, and more.

Dual use: Can the client be used for both POP and IMAP? Can you check email for more than one account on the same machine? Can you change the From: address on a per message basis?

Managing mailboxes: Can mailboxes be moved from the server to the local machine and back again easily? Can you create a new mailbox as you transfer a message?

IMAP feature support: Can you read email in shared mailboxes? Can you change ACLs for your own mailboxes from within the client?

Ease of use: How easy is it to do common functions (sending mail, replying, forwarding, reply all or reply to sender, etc.)? What kind of help, manuals, etc. comes with the client? Can you search within mailboxes, within the entire list, etc. for specific phrases or words? What kind of sorting capability is there? How does deleting messages work?

Special features: Are there signatures available? Is there spell checking (and is it good?) Can you easily show the full headers of a message, or hide them? Can you do client level filtering? Is attaching files an easy task? How can you manage attachments? Are nicknames available and easy to use?

4.1.2 Picking a Client

As part of our project, we evaluated five clients that work on both Macs and Windows machines, Eudora, Outlook Express, Mulberry, Simeon, and Netscape Communicator. We looked briefly at MailDrop from Baylor, but since it seemed to only allow for clear text password connections, we did not bother to evaluate it. We also looked at Outlook for Windows, and while it provided additional functionality beyond what we were specifically looking for, we preferred to focus on clients that would allow us to support the same product on both platforms. An Outlook for Mac does apparently exist, but it only works with Microsoft Exchange mail servers, not those that rely on standard protocols. Based on

our client evaluation, we determined that for Brown, Eudora was the best choice.

At SIGUCCS, many of the schools had chosen Mulberry as the preferred client. Our project team felt that the dramatic difference in user interface from Eudora to Mulberry, combined with what they saw as an inelegant implementation for a mail client pushed us toward the client that our campus was already familiar with, Eudora. We also decided to provide limited configuration documentation for Outlook Express and the Netscape Messenger clients, since those were reasonable alternatives. Those two were also clients we knew to be widely used on campus already, and we wanted to make the transition to IMAP as easy as possible for our existing customers.

4.2 Web clients

Selecting a web client has been a difficult issue for us. We thought that we had the situation well in hand because our selected server vendor provided a web client, as well as the Simeon desktop clients as part of our license. Between the time when we purchased our server and our planned implementation of IMAP for Fall 2000, the vendor told us that they were moving out of the client business, and the web client was no longer supported.

4.2.1 POP Web Client Needs

Another wrinkle came up because of one of the other email projects that had been started. Turning off our clear text email port (only POP services were enabled) meant that we needed to move everyone from clients they used to ones that were capable of working in a secure way, either SSL or Kerberos. While the two security mechanisms worked very differently, what was important to us was that they both protect your password.

Netscape's mail client is a commonly used one on campus, as is a student written web-based client called MailRiot. Neither of these clients supports either password protection mechanism. While Outlook and Outlook Express do support SSL, and those are also commonly used on campus, we wanted to be able to provide a web-based POP client to help those who were using MailRiot or Netscape to migrate easily. There were also quite a few people who used Eudora while on campus, but while travelling or at home for the summer, would use the services that Yahoo and Hotmail provide to read POP mail at other sites. For these users, too, we needed a web based POP client.

So, now our IMAP web client would not be vendor supported (nor net supported since it was a commercial product originally, and not widely used), and we also needed to have a POP web client to take care of the other needs.

4.2.2 Locating Web Clients

From the IMAP web site (www.imap.org), we were able to get a list of IMAP web clients, and some of those had a POP interface as well as IMAP. Those were the ones we targeted first. After a quick view at web sites for the various companies, there was nothing obvious to eliminate any of them from the running for user interface reasons, so it was up to systems to evaluate them for manageability and whether they would work in our environment at all.

That was slow going. The systems person felt that the only way to learn about the various products was to download them and set them up. Once installed, they needed tweaking to work in our environment, and then testing to see if the local Brown quirks (apostrophes and hyphens in mailbox names) would work with the clients. The first few failed the test. As of this writing, we have not identified a client that will work with our quirks and fulfill our preference for a web client that will do both POP and IMAP. That has also held up our efforts to close down the clear text port on our mail server, so we are continuing to move forward on solving this problem.

5. OTHER FEATURES

As we worked through how IMAP services were different from POP, two other possible enhancements to our service were discussed. Neither was specifically scheduled or planned because we did not get close enough to implementation to make a determination about the importance of them, but they may be factors for other schools to consider in planning an IMAP implementation.

The first was a startup issue: migrating mail from desktop folders to the server folders did not seem to be a trivial task for those who store a lot of mail in a lot of folders. This migration of mailboxes in some automated fashion was one possible add on to the project. Migrating mailboxes is a one time pain, and since we are going to continue supporting Eudora, the pain was going to be fairly small. So, we did not see this kind of migration utility as a critical component, but merely a nice to have one.

The second issue was about backups and restores. In a POP environment, everyone is responsible for backing up their own email on their own hard drives. In theory, at least, if they accidentally discard something from their email that they need, they may be able to retrieve it from a backup that they manage. With all or most of a person's email stored on the server, would the service be responsible for delivering that level of restore capability? Could we restore a user's entire directory, or a single mailbox, without adding a lot of overhead to our systems staff or operations (who manage the backups)? While we would like to be

able to deliver this as part of our service, some of the same funding issues have held up a replacement backup system, so it is not clear whether we can deliver it. We already do disaster recovery backups, but those will not allow us to deliver a service that enables user requested restores of their own space. This issue is also on hold until we learn more about the funding issues.

6. CONCLUSIONS

At the moment, this project seems quite frustrating since we are not able to deliver our service for the fall semester as we had expected to. We put a lot of good effort into trying to create a product that would be truly useful to the campus community, rather than just trying to create something cheaply that would deliver only half way. Our evaluation process for the server and the client were more professionally done than we have been able to execute on other projects here. The planning involved in making decisions was critical in making the service we planned to deliver into a fully functioning package.

When the funds become available for the appropriate disk arrays and backup systems, we will deliver a quality service to the campus that should go a long way toward meeting the campus needs for email. We believe that delivering a service with very limited disk available would cause more problems than it would solve at this time. Some of the problems that we foresee in doing this with small quotas include:

- the negative impact on the Help Desk from those whose mail bounces due to quota exceeded conditions

- the unhappy response from those we want to move from mainframe email to IMAP because the service doesn't deliver the kind of storage that they are accustomed to in VM

- the concerns of the systems area that even with a 10 meg quota, we would have difficulty doing appropriate backups and disaster recovery if we do not use better storage mechanisms

While we are not content with not having a service to deliver in the fall, we feel that we made the right decision, and hope to see our service appropriately funded in the near future.