

Mailman

Discussion Groups

Jerry Stratton

April 6, 2005

Creating a Discussion Group

The web page for creating a USD mailing list is at <http://www.sandiego.edu/ins/spam/mailman.php>. This page contains information about how lists work at USD, and, if we are having any issues with mailing lists, should contain a summary of the error and when we expect it to be fixed.

You will need to sign in with your USD UNet account and password in order to request a mailing list.

Authorized Contacts

Your authorized contact is the person who is authorized to request password changes, request that the list be removed, and request that authorized contacts be changed.

The authorized contacts are not the same as the list administrators or list moderators. Authorized contact information is kept on file at ITS, and must be changed with ITS. If the authorized contacts are unavailable, the process of determining who should be an authorized contact will take considerable time; the list may experience significant downtime if the authorized contacts that ITS has on file are unavailable. So, please, keep your authorized contacts up to date.

You should have at least two authorized contacts, so that one authorized contact is always available. At least one authorized contact must have a USD UNet account. We recommend that all authorized contacts have USD UNet accounts.

More information about authorized contacts is available at <http://www.sandiego.edu/webdev/contacts.php>.

Mailing List Name

Your mailing list's name is the part of the list's e-mail address on the left side of the "@" symbol. So, if you choose "musketeers" as your list's name, your list's address will be `musketeers@lists.sandiego.edu`.

More Information about Mailman

You can get more information about Mailman from the official Mailman web site at <http://www.gnu.org/software/mailman/>.

Regular Expressions

Several of the options available to you accept what are called “regular expressions”. Regular expressions are a means of specifying multiple options at once. For example, if you want to block all messages that contain the phrases “make money now”, “make cash now”, “make simolas now”, you could use the regular expression “make .* now”. Any phrase that contains the letters “make” followed by a space, followed by any characters of any kind, and then followed by a space and the letters “now”, will be “matched” by that regular expression.

Mailman is written in a programming language called *Python*, so it uses Python’s regular expressions. You can get an in-depth tutorial on Python’s regular expressions at <http://www.amk.ca/python/howto/regex/>.

Regular expressions can be intimidating, and are not at all necessary for managing a mailing list. If you want to use them, however, they can make some tasks easier. You do have to be careful, however, or you can block more messages than you expected, or let more messages through than you expected.

We’ll cover only the simplest regular expressions here.

The asterisk

The asterisk character (*) *repeats* the previous character zero to any number of times. The regular expression “le*” will match “l”, “le”, “lee”, “leee”, and so on. The asterisk immediately follows the character “e”, so any number of “e” characters, including zero “e” characters, will match.

The plus

The plus character (+) repeats the previous character one or more times. The regular expression “le+” will match “le”, “lee”, “leee”, and so on, but will not match “l”.

The question mark

The question mark character (?) repeats the previous character zero or more times. The regular expression “le?x” will match “lx” or “lex” but not “leex”.

The dot

The dot, or period (.), will match any character once. The regular expression “li.a” will match “lina”, “liza”, “lisa”, “lira”, as well as nonsensical combinations such as “li9a” or “li%a”.

The dot is useful in combination with the asterisk or plus. The regular expression “li.+a”, for example, will match not only “lisa” but also “lissa” or “linda”. It will also match such

strange things as “li%9a” or “li5t5adeda3a”. It matches one or more of any characters between the “li” and the “a”.

Grouping characters

You can group a set of characters using parentheses. The regular expression within parentheses acts as a single character from the perspective of outside the parentheses.

For example, if you want to match both Lisa and Lisanne, you could use the regular expression “Lisa(nne)?”. This will match the characters “Lisa” and either zero or one occurrences of “nne”: “Lisa” or “Lisanne”.

A list of characters

You can list specific characters using square brackets. The regular expression “[sndz]” will match either ‘s’, ‘n’, ‘d’, or ‘z’.

The list of characters is a lot like the dot. It only matches one character, but that character may be any of the characters listed in the square brackets. Like the dot, the character list is very useful with the plus and the asterisk.

For example, the regular expression “li[sndz]+a” will match “lisa”, “lissa”, “Linda”, “lisza”, or any combination of “li”, the characters ‘s’, ‘n’, ‘d’, or ‘z’ in any combination, and the letter “a”. It will, for example, also match “lindndnsza”.

Placing a caret at the start of the list means any character *except* these characters. The expression “[^sndz]” would match any character *except* s, n, d, or z.

Limiting the number of characters

Finally, you can limit the number of characters using curly brackets. The limit is two numbers separated by commas. The previous character must repeat no fewer times than the first number, and no more times than the second number.

The regular expression “lis{1,2}a” will match “lisa” and “lissa” but not “lia” or “lissaa”.

Like the plus and the asterisk, the character limit is very useful with the dot and the list of characters. For example, the regular expression “li[sndz]{1,2}a” will match “lisa”, “linda”, “lissa”, or liza” but will not match “lindndna or any other combination of more than two characters.

Not

Sometimes it is easier to say what you don’t want than what you do want. This requires what is called a “negative lookahead”. Basically, surround your regular expression with parentheses, and then after the opening parentheses, put a question mark and an exclamation point.

This will mean that the target will match if that regular expression does *not* match the target.

For example, the regular expression `(?!jerry)` will match any set of characters that does *not* contain the word ‘jerry’ or the characters ‘jerry’ in that order.

Using them in Mailman

In most cases, you need to put your regular expression on a line that begins with a caret symbol. The caret at the beginning of a line tells Mailman that this line is a regular expression.

For example, the “List of non-member addresses whose postings will be automatically discarded” takes both simple addresses and regular expressions. If you want to block all messages that do not come from an address at USD, you could use:

```
^(?!.*sandiego.edu)
```

The `^` matches the beginning of the address (as well as letting Mailman know that this is a regular expression). The `!` tells Mailman that we want everything that *doesn't* match this regular expression to be what it takes action on—in this case, be automatically discarded. The `.` matches any character, and the `*` matches any number of those, so this matches any number of any characters. But whatever those characters are, they must end in ‘@sandiego.edu’.

The sharp-eyed may notice that `.` in `sandiego.edu`. This does really mean ‘sandiego’, any character, and ‘edu’. If you specifically want a period, you need to precede it with a backslash:

```
^(?!.*sandiego\.edu)
```

You can see what items can take regular expressions by clicking on the “more details” link for that item. If it accepts regular expressions, it will say something like “start the line with a `^` character to designate a regular expression match”.

Other items *expect* a regular expression. With those, you do not need the initial caret character. For example, “Hold posts with header value matching a specified regexp” expects a regular expression. All of the lines listed are regular expressions whether you use a caret or not.

Error Messages

E-mail error messages tend to have a lot of incomprehensible text. When you receive an error message after sending a message to your list, you will need to search the error message for the comprehensible text. Most of the time there is an error that tells you exactly what went wrong.

There are two things you're looking for. First, did the message fail while going to the list, or did it fail while going to a member of the list? If it failed on a member, then all the other members received the message. If, on the other hand, your message was rejected before it even made it to the list, then nobody received it.

Second, why did the error occur? What will you need to change for the message to go out if you send it again?

Stale NFS Handle

All of the Mailman files are stored on an NFS server, but the NFS server appears to occasionally lose itself. If that happens when you are making a change, you will receive this error message.

```
Bug in Mailman version 2.1.1
We're sorry, we hit a bug!

If you would like to help us identify the problem, please email a copy of this
page to the webmaster for this site with a description of what happened.
Thanks!

Traceback:

Traceback (most recent call last):
  File "/home/mailman/scripts/driver", line 87, in
run_main
    main()
  File "/home/mailman/Mailman/Cgi/admin.py", line 162, in
main
    mlist.Lock()
  File "/home/mailman/Mailman/MailList.py", line 145, in
Lock
    self.__lock.lock(timeout)
  File "/home/mailman/Mailman/LockFile.py", line 282, in
lock
    elif self.__read() == self.__tmpfname:
  File "/home/mailman/Mailman/LockFile.py", line 424, in
__read
    filename = fp.read()
IOError: [Errno 70] Stale NFS file handle
```

The important part of this error message is the phrase “Stale NFS handle”. When you receive this error, copy it and e-mail it to me, then wait a few seconds and try again.

We're working on this one. We hope to have this fixed in a couple of weeks when we move away from NFS for the mailing list server.

This is a very rare message. I see it only about once a month, and I use the mailing list administration pages every day.

Not allowed to post

This one usually happens to list administrators. When you add an address to the list of administrators, you are not adding them to the membership list.

You are not allowed to post to this mailing list, and your message has been automatically rejected. If you think that your messages are being rejected in error, contact the mailing list owner at educ184_284-owner@lists.sandiego.edu.

Received: from mx1.sandiego.edu (mx1.sandiego.edu [192.55.87.54])
 i8FJlscY025733 for <educ184_284@lists.sandiego.edu>;
 Wed, 15 Sep 2004 12:18:54 -0700 (PDT)

Received: from [199.106.158.125] (USD-C1-02.acusd.edu
 [199.106.158.125])
 i8FJlqIS000172 for <educ184_284@lists.sandiego.edu>;
 Wed, 15 Sep 2004 12:18:53 -0700 (PDT)

Mime-Version: 1.0
 X-Sender: viviana@pop.sandiego.edu
 Message-Id: <a05200f0abd6e3bbc4efb@[199.106.158.125]>
 Date: Wed, 15 Sep 2004 12:18:52 -0700
 To: educ184_284@lists.sandiego.edu
 From: Viviana Alexandrowicz <viviana@sandiego.edu>
 Subject: reading assignment
 Content-Type: text/plain; charset="us-ascii" ; format="flowed"

When you receive an error that you are not allowed to post (or one of your moderators or administrators receives this error), verify that they are in fact a member as well as a manager/administrator.

Bad Subjects

Occasionally we block subjects at the mail server. These messages never reach the mailing list. Some of the subjects we have blocked in the past include “test” and “hello”. The rejection message is extremely unhelpful. All it will say is “The following addresses had permanent fatal errors”. It will give no indication that the error was the subject line and not the addresses.

```

-----Original Message-----
From: Mail Delivery Subsystem <MAILER-DAEMON@BlackBerry.NET>
Date: Wed, 3 Mar 2004 15:46:30
To: <jonbmaher@tmo.blackberry.net>
Subject: Returned mail: see transcript for details

The original message was received at Wed, 3 Mar 2004 15:46:13 -0500
(EST)
from mailrouter03.bwc.prod.on.blackberry [172.16.147.102]

----- The following addresses had permanent fatal errors -----
<sigmachi@lists.sandiego.edu>
  (reason: 550 5.7.0 Message rejected.)
<jerry@sandiego.edu>
  (reason: 550 5.7.0 Message rejected.)

----- Transcript of session follows -----
... while talking to mx1.sandiego.edu.:
DATA
<<< 550 5.7.0 Message rejected.
554 5.0.0 Service unavailable

Sent via BlackBerry from T-Mobile.Reporting-MTA: dns; BlackBerry.NET
Received-From-MTA: DNS; mailrouter03.bwc.prod.on.blackberry
Arrival-Date: Wed, 3 Mar 2004 15:46:13 -0500 (EST)

Final-Recipient: RFC822; sigmachi@lists.sandiego.edu
Action: failed
Status: 5.7.0
Remote-MTA: DNS; mx1.sandiego.edu
Diagnostic-Code: SMTP; 550 5.7.0 Message rejected.
Last-Attempt-Date: Wed, 3 Mar 2004 15:46:30 -0500 (EST)

Final-Recipient: RFC822; jerry@sandiego.edu
Action: failed
Status: 5.7.0
Remote-MTA: DNS; mx1.sandiego.edu
Diagnostic-Code: SMTP; 550 5.7.0 Message rejected.
Last-Attempt-Date: Wed, 3 Mar 2004 15:46:30 -0500 (EST)
From: "Jonathan B. Maher" <jonathan.maher@host1.com>
Date: Wed Mar 3, 2004 1:00:16 PM US/Pacific
To: "Sigma Chi" <sigmachi@lists.sandiego.edu>
Cc: "Jerry Straton" <jerry@sandiego.edu>
Subject: Test

Sent via BlackBerry from T-Mobile.

```

There really isn't a canonical way of telling that your subject is a bad subject. The real clue is that there is no error message other than "Service unavailable" and "Message rejected". When you receive a message like this, and your subject is one you've seen on spam or viruses, try a different subject. Most subjects that are blocked are ones with no specific meaning.

User unknown

Another common error occurs when a member is subscribed with the wrong address, or a member changes their address without changing it on the list. You can tell these messages by the presence of “User unknown” or “unknown user” or some variation, and because the error message does not come from lists.sandiego.edu. The example here comes from the main mail server at USD, for example.

```

Date: Tue, 24 Jun 2003 11:51:43 -0700 (PDT)
From: Mail Delivery Subsystem <MAILER-DAEMON@SanDiego.edu>
To: <jtarbox@SanDiego.edu>
Subject: Returned mail: see transcript for details
Auto-Submitted: auto-generated (failure)

The original message was received at Tue, 24 Jun 2003 11:51:43 -0700
(PDT)
from hannah [192.55.87.48]

----- The following addresses had permanent fatal errors -----
<scales@is.acusd.edu>
  (reason: 550 RCPT TO:<scales@is.acusd.edu> User unknown)
<tarboxj@is.acusd.edu>
  (reason: 550 RCPT TO:<tarboxj@is.acusd.edu> User unknown)

----- Transcript of session follows -----
... while talking to is.acusd.edu.:
DATA
<<< 550 RCPT TO:<tarboxj@is.acusd.edu> User unknown
550 5.1.1 <tarboxj@is.acusd.edu>... User unknown
<<< 550 RCPT TO:<scales@is.acusd.edu> User unknown
550 5.1.1 <scales@is.acusd.edu>... User unknown
<<< 554 DATA Transaction failed, no recipients given
Reporting-MTA: dns; mx2.sandiego.edu
Received-From-MTA: DNS; hannah
Arrival-Date: Tue, 24 Jun 2003 11:51:43 -0700 (PDT)

```

Note that when you receive a “User unknown” message, only the listed users did not receive the message. Unknown user errors come from that user’s server, not from the mailing list. Everybody else has received the message.