Elm Forms Mode Guide

What Forms Mode is, how to use it to create custom forms, how to reply to forms, and how to use it for AT&T Mail messages

The Elm Mail System
(Version 2.4)

Syd Weinstein, Elm Coordinator
Datacomp Systems, Inc.
3837 Byron Road
Huntingdon Valley, PA 19006-2320

email: elm@DSI.COM or dsinc!elm

© Copyright 1986,1987 by Dave Taylor
© Copyright 1988-1992 by The USENET Community Trust

A Guide to Forms Mode in Elm

(The Elm Mail System, Version 2.4)

October 1, 1992

Syd Weinstein
Datacomp Systems, Inc.
3837 Byron Road
Huntingdon Valley, PA 19006-2320

email: elm@DSI.COM or dsinc!elm

Derived from
''The Elm Mail System, Version 2.0''
by
Dave Taylor
Intuitive Systems
Mountain View, California
email: taylor@intuitive.com or limbo!taylor

While there are a lot of mail systems that allow the transmission of text and primitive pictures, to send and reply to more complex forms is simply not possible. Elm, however, took the forms mode implemented as part of the AT&T Mail package and has expanded on it to be a smoothly fit part of the overall mail system.

Forms mode gives you the ability to send ''template'' files to people and receive the filled-in replies.[1] Let's look at an example right off.

Say we were going to use computer mail as a way to file defects with software. There is a certain amount of information we want to be able to collect when each report is made, and if it is in a specific format we can use programs to file the defects upon receipt.

The form we'll try to emulate starts out looking like:

1. Note that this feature assumes that the person on the other end is also using the Elm mail system, that both yourself and the person on the other end have their user levels set to something other than *Beginner* (**0**) and the **forms** variable set ON in their *.elm/elmrc* files.

Defect Reporting Form

Program: _____ Version: _____
Operating System: _____ Version: _____

Defect Type: _____

Date Found: _____ By Whom: _____
Date Reported: _____ Phone: _____

Description: _____
_____
_____

This form can actually be created almost exactly as listed above in the Elm mail system by using your standard editor and can then be mailed about as needed.

Let's say that we want a bit more information, however, especially with fields like *Defect Type*, we want to list all the recommended answers. To create the actual form, we need merely to replace the underlines in the above form with spaces. The multi-line comments can simply be indicated by a ':' by itself on a line:

Defect Reporting Form

Program: Version:
Operating System: Version:

(Valid Defect Types are: user-error, doc-error, fatal, other)
Defect Type:

Date Found: By Whom:
Date Reported: Phone:

Description
:

Thank you for filling in this form.

As you can see, it is quite simple to create forms!!

Now that you have an idea what we're talking about, let's actually officially define the system.

Forms Mode Specification

[Note that this is all taken from the document *Standard for Exchanging Forms on AT&T Mail*, Version 1.9 of 6/7/86, from AT&T.]

The forms mode is really quite simple. Simple enough that it is amazing that it hadn't been implemented before AT&T Mail came along!!

In a nutshell, each field is delimited by a ':' followed by a number of blank spaces or tabs that represent the valid size for that field. That is, if we have a line in the form like:

```
Phone (area-code):     Number:
```

The area-code field is limited to three characters and the number to nine (this is kind of hard to see with the proportionally spaced formatted copy, alas). The only exception to the rule is that a ':' by itself on a line represents a field that is as large as the user entering the data desires.

The actual form that is transmitted, in AT&T Mail parlance, is a ''SIMPLE'' forms handler message (as opposed to the ''ADVANCED'' handler). This means that it contains three sections:

The Message Header
```
[OPTIONS-SECTION]
***
[FORMS-IMAGE]
***
[RULES-SECTION]
```

Elm generates form messages with the *OPTIONS-SECTION* filled out, but ignores it when receiving mail. The filled out *OPTIONS-SECTION* is:

```
WIDTH=80
TYPE=SIMPLE
OUTPUT=TEXT
```

The *FORMS-IMAGE* section is that described above, i.e. prompting text followed by a ':', followed by spaces or tabs. The *RULES-SECTION* can contain explicit rules about the possible values of each field, but this is currently ignored by Elm, being a ''SIMPLE'' forms mode mail system.

Forms also have the header ''Content-Type: mailform'' to indicate to the mail system (either Elm or AT&T Mail) that a form is being sent.

Elm further indicates that a form has been received by having an 'F' as the status character in the header display section (instead of 'N' for new, etc).

Composing and Sending a Form

The first step to enable sending forms is to change the setting of the variable **forms** in your *.elm/elmrc* file to ON:

```
forms = ON
```

The next step is to send the message to someone using the **m** (**mail**) command, which drops you into an editor. Type in the form as indicated above, with appropriate colons and comments, and end the entry by leaving the editor.

The prompt is now:

```
Choose: E)dit msg, edit H)eaders, M)ake form, S)end or F)orget : @
```

so we choose **m** — **make form**. Elm then either rewrites the prompt without the ''M)ake form'' option, indicating

that the form has been accepted, or indicates the problem and gives you a chance to correct it.

Once it has been accepted, simply use the **s — send message —** command and it's off!

Note that you cannot reply to a message with a form.

Replying to a Form

Let's reply to the form message we generated now. The header page of the Elm mail system indicates that the message is a form by having an 'F' next to it. So we use **r** to reply and the screen is immediately cleared and we're prompted, field by field, for the data requested. Each field has underscores in the input area to indicate the size field that is expected.

After answering all the questions we'll have a screen that looks like:

```
                              Defect Reporting Form

Program:  The Elm Mail System_____________________
Version:  1.5_____________________________________
Operating System:  HP-UX__________________________________
Version:  5.141 C_________________________________

(Valid Defect Types are: user-error, doc-error, fatal, other)
Defect Type:  fatal_______________________________________________________________

Date Found:  10/9/86______________________________________
By Whom:  Dave Taylor_____________________________________
Date Reported:  10/9/86___________________________________
Phone:  (415) 857-6887___________________________________

Description
(Enter as many lines as needed, ending with a '.' by itself on a line)
  When running it on a CPM system I cannot compile successfully.
  .
Thank you for filling in this form.
                    Choose: E)dit form, edit H)eaders, S)end or F)orget : @
```

Quite simple. Notice, however, that the order of prompting is left-to-right on each line, so the fields, *By Whom:* and *Phone:*, although placed in what seems like a logical place on the form, turn out to be confusing when filling in the received form since it isn't clear what *Phone:* is being asked for because of the intervention of the *Date Reported:* field.

The message that is actually sent out from this has the fields in a more acceptable format:

```
WIDTH=80
TYPE=SIMPLE
OUTPUT=TEXT
                            Defect Reporting Form
***

Program: The Elm Mail System                                  Version: 1.5
Operating System: HP-UX                                       Version: 5.141 C

(Valid Defect Types are: user-error, doc-error, fatal, other)
Defect Type: fatal

Date Found: 10/9/86                                  By Whom: Dave Taylor
Date Reported: 10/9/86                               Phone: (415) 857-6887

Description

   When running it on a CPM system I cannot compile successfully.

Thank you for filling in this form.
***
```

Comments on Forms Mode

As was said at the beginning, this way of sending about forms could prove to be very helpful and useful in a variety of contexts. On the other hand, until a more sophisticated forms language is used for the forms, this should be sufficient to embody the power of the idea.

I welcome any comments and thoughts on this system and also welcome possible enhancements.

I also gratefully thank Dale DeJager of AT&T Information Systems for sending me more information on AT&T Mail than I could possibly digest in any finite amount of time.