Wikiprint Book

Title: Basics ? what is e-mail?

Subject: YAM - Yet Another Mailer - Documentation/What is e-mail

Version: 12

Date: 04/02/15 09:59:24

Table of Contents

Languages: 3
Basics ? what is e-mail? 3
Make up of a message header 3
Mail protocols 4
Outgoing mail 4
Incoming mail 4
What is MIME? 4
MIME Encoding 4
Content type 5
Encrypting with PGP 5
E-mail netiquette 6

Languages:

- **English**
- Español
- Français

Basics ? what is e-mail?

E-mail was the first service to be implemented on the Internet and is still the most popular method of communication over the Internet. A substantial proportion of the people on the Internet use only this service. It is predicted that in the next few years e-mail will replace the traditional forms of communication like letters and fax for many purposes. At the present time one can in a matter of minutes reach several million people world-wide by e-mail. Users of other networks, such as Compuserve and T-Online, are connected to the Internet through "gateways".

Make up of a message header

Every e-mail starts with a header which is separated from the actual message body by a blank line. YAM constructs this header automatically following the settings specified in the configuration.

The header of an e-mail is divided into fields which each start on a new line and have the general form 'Field: contents'. Fields which are too long to fit in a single line can be split to run over several lines. The majority of the header lines can be omitted but are added to provide the recipient's mail program with additional information about the message or to give data needed to check for errors caused by transmission problems. A few of the important fields are explained by means of the following example.

```
Return-Path: <just@zfn.uni-bremen.de>
```

This field is added by the recipient's mail server and contains the e-mail address of the sender to allow the recipient's computer to send a reply by e-mail.

```
Received: from ina.zfn.uni-bremen.de by atlantica.access.ch
          (8.8.5/INA-1.05pri) id XAA29100;
          Tue, 23 Dec 1997 23:40:45 +0100 (MET)
Received: from moritz37.zfn.uni-bremen.de by
          ina.zfn.uni-bremen.de (AIX 3.2/UCB 5.64/ZFNserver) id AA26355;
          Tue, 23 Dec 1997 23:40:13 +0100
```

Each computer which sends the message on the next stage of its journey, and also the recipient's own system, adds on a "Received:" field to the header to say when the message arrived and where it came from. This allows one to reconstruct transmission problems which may have affected the message along its route.

```
From: Christian Just <just@example.net>
```

E-mail address and real name of the sender.

```
Reply-To: just@example.org
```

The address to which any reply to this message should be sent. This is used if the message is despatched from a computer which the sender cannot use to read mail, for whatever reason. Then he can use this field to say where a reply should be sent. In the absence of a **Reply-To:** field, replies go to the address given in the **From:** field.

```
To: "Marcel Beck (Yet another Mailer-author)" <mbeck@example.com>
```

The address of the recipient; additional names can be given, separated by commas. This field may contain simply the address in the form name@domain or else it may be prefaced by an additional comment, such as the recipient's name -- in this case, the address must be enclosed in pointed brackets.

```
Date: Tue, 23 Dec 1997 23:28:30 +0200
```

Date and time of despatch of the message. The figure after the time represents the timezone, expressed as the difference from Greenwich Mean Time. Here the first two digits show the hours, and the next two the minutes.

```
Message-ID: <YAM7296.1549.122414920@zfn.uni-bremen.de>
```

An unique identifier, created automatically on despatch. Using this it is possible unequivocally to cancel any particular message.

```
In-Reply-To: <YAM7296.263.121102896@mail.access.ch>
```

A precise reference to the message being replied, such as its Message-ID

```
X-Mailer: YAM 2.0beta4 - Amiga Mailer by Marcel Beck - http://yam.ch
```

Name and Version of the sender's mail program.

```
Subject: Re: YAM2beta5
```

This field should state concisely the topic of the message. Giving a clear and informative Subject is a mark of good style Netiquette.

```
 Mime-Version: 1.0
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain; charset=iso-8859-1
```

These fields indicate that the message is set out in MIME format. In this example, it contains plain text in the ISO-Latin-1 character set and characters which cannot be represented using 7 bits are encoded as 'Quoted-printable'.

Mail protocols

YAM uses the Simple Mail Transfer Protocol (SMTP) in order to send your mail to the SMTP server, which also uses SMTP to transmit your mail across the whole world. Incoming mail arrives at your Post Office Protocol (POP) server, where it waits until YAM downloads it to your computer, using the POP3 (POP version 3) protocols. The messages which YAM sends and receives conform to the conventions set out in RFC 822 and RFC 1521 (MIME).

Outgoing mail

If you want to send an e-mail message to anyone, YAM transfers the message by SMTP to your local SMTP server. This computer forwards the message to the recipient's computer, generally also by SMTP.

Why does YAM not deal directly with the recipient's server ? Firstly, it would take quite a long time for your Amiga to get a connection to one particular computer and then transmit the message. Secondly, many computers are hard to find; it is much better to let the mail server look for the address, instead of burdening your Amiga. Thirdly, quite frequently the recipient's server will not be available at the time you want to send the mail. The SMTP server solves these problems, holding back the message until the other computer is ready to receive it.

Incoming mail

If someone sends you email, the other computer transfers it using the SMTP protocol as far as your POP server. This stores the message in a sort of mailbox, where it remains until YAM collects it. When you look for new mail, YAM downloads the message to your Amiga using POP3.

Why doesn't YAM use SMTP for incoming mail? SMTP works best if both computers are ready to receive messages. Unless you run YAM and your Amiga 24 hours a day and seven days a week, SMTP would not be particularly suitable for you.

What is MIME?

MIME stands for *Multipurpose Internet Mail Extensions*. MIME serves two main purposes: it allows one mail application to tell another what sort of data is contained in a message, and it also provides standardised rules by which mail applications can encode data, so that it can be sent through the Internet mail system.

MIME Encoding

The Internet uses the SMTP protocol to move mail around. SMTP is limited to the US-ASCII character set. This is a problem for people who speak languages other than American English and so need accented characters or non-American letters, or for people who want to use special symbols like the

bullet. Even more difficult is the transmission of binary files, as it is often the case with attachments. MIME provides a way around this restriction by offering two encodings: **quoted-printable** and **base64**.

These encodings use US-ASCII character codes to represent any sort of data you like, including special characters or even non-text data. Quoted-printable is used for data that is mostly text, but has special characters for very long lines.

Quoted-printable looks just like regular text, except when a special character is used -- the special character is replaced with a "**=**" (dash) and two more characters that represent the (hexadecimal) character code of the special character. Thus, a bullet in quoted-printable looks like **=95DA**.

No line in quoted-printable is allowed to be more than 76 characters long. If your mail has some line longer than 76 characters, the quoted-printable encoding will break your line in two, and put a "**=**" at the end of the first line, to notify the mail reader at the other end that the two lines are really supposed to be one.

Base64 encoding is another way to protect binary data from the SMTP mail system. However, Base64 makes no attempt to be readable, and is more appropriate for non-text data. It is equivalent to the older UUencode, but more reliable in use.

Content type

The other important function of MIME is to allow mail programs to exchange information about what kind of data is in a message (or part of a message). The primary mechanism used for this is the **Content-Type:** header. The main content types are:

```
 text        readable text
image       pictures and graphics
audio       sound
video       animations
message     messages or parts of messages
multipart   several different kinds of data in a single message
```

The subtype gives additional information about the type of data:

```
 text/plain   plain text
text/html    text in HTML format
image/gif    image in GIF format
etc.
```

By looking at the **Content-Type:** header, a mail program can select the most suitable utility to display an attached file.

Encrypting with PGP

In order to ensure that the e-mail cannot be read by anyone other than the recipient, it is necessary to encrypt the transmission. Is this important? Sometimes very much so! It is not possible to say in advance what route electronic mail will take through the Net and along the way it is possible for someone to read your mail unauthorised, admittedly with more trouble than one would normally bother to take. In particular, encrypting e-mail is a wise precaution if you want to send passwords, credit card numbers or some such over the Net. Such encrypted data is then often transmitted more safely than if sent by normal letter post. A simple, effective and widely used tool for this sort of encryption is **PGP**, short for **Pretty Good Privacy**.

PGP was developed by Phil Zimmermann and employs the *public key* method. Using this PGP program, one can be sure that the message is the one actually written by the sender, and that only the intended recipient can read it. The so called *public keys* offer the highest possible level of security.

There are two kinds of key:

- One is a private key, used on your computer and never revealed elsewhere.
- The other sort is the public key. You can make as many copies of this as you like, and send the copies to other users so that they can send you encrypted mail.

You need both types of key, public and private, because they are inherently connected together. You can distribute your public key as often as needed, but it will only work when matched up with its exact counterpart. Hence, both public and private keys are involved in locking and (generally) unlocking information.

PGP keys are used in two distinct ways:

i. Another person can encrypt information using your public key and send the encrypted file to you, to decipher with your private key.

i. You can encrypt information with your private key and send it safely over the Net. Anyone in possession of your public key can read your communication. The recipient can be sure that the communication is genuinely from you (your digital signature proves its authenticity) and that it has not been altered.

PGP is obtainable as freeware and the International PGP homepage is easy to find on the Internet at ?http://www.pgpi.org/

Related topics:

- Installing PGP 2.6.x
- Installing PGP 5.0i

E-mail netiquette

- Keep your messages short and to the point.
- Give the message a concise and meaningful **Subject:** header, so that it can easily be found again.
- Put a signature at the end of the message. This should contain your name and e-mail address and should not be longer than five lines. Signatures often also give the postal address, telephone number, website and instructions about sending PGP messages.
- Only write in block capitals if you want to give particular emphasis to a point. *Stars* are also used for emphasis (YAM interprets this by using bold type). Remember, block capitals are generally taken as equivalent to SHOUTING.
- Set up YAM to use a line length of less than 80 characters and don't use any control characters.
- Don't use non-ASCII characters unless you are sure that the recipient's software interprets them correctly.
- It is regarded as extremely impolite to forward a private message to a mailing list without the permission of the original sender.
- Abbreviations and *TLAs* (Three Letter Acronyms) can be useful provided that they don't make the message unintelligible. The following are often used:

  IMHO (in my humble opinion)
  BTW (by the way)
  FYI (for your information)
  AFAIK (as far as I know)
  ASAP (as soon as possible)
  CU (see you)

- Use the smiley (winky?) to denote a touch of irony which could otherwise easily be misinterpreted.
- Be tolerant of people's failings, e.g. in spelling, grammar, expression or familiarity with e-mail.
- If you join a mailing list, read the messages for a while to get a feel of the general style, what questions are asked and what is not suitable.
- Study the *FAQs* (lists of Frequently Asked Questions). It is annoying for the members of a group if the same questions are being asked repeatedly.
- If someone posts a message which is "off-topic", (i.e. it does not belong in the mailing list), reply privately and NOT in the list.
- If you quote another message, then cut all the bits which are not relevant to your answer. No-one wants to read the same message three or four times, especially when all that is added is "Yes, me too".
- Resist the temptation to flame in the mailing list. ("Flame" = write abusively, generally when provoked by idiocy.) Remember that the list is public and meant for constructive discussion. Do as you would be done by!
- If you are replying to a message coming from a mailing list, think carefully whether to send it direct to the sender or to the list, and check that the *To:* address is correct. It can be very irritating when a letter addressed to a particular person appears on the list, unintentionally.
- When replying to a message coming from a mailing list, it is usually a good idea to mention the person who wrote that message as part of your welcome phrase, to avoid confusion.