Как освежить отношения с партнёром по e-mail

Всё, что вы хотели знать о заголовке (X-)Face, но боялись спросить

Shroom [OmskLUG]*

25 сентября 2013 г.†

Статья рассказывает о заголовочных полях сообщений электронной почты X-Face и Face, о способах их генерации и внедрения в письмо. Также затронуты темы фильтрации изображений с помощью утилит из пакетов **netpbm** и **ImageMagick**.

Что такое e-mail message headers?

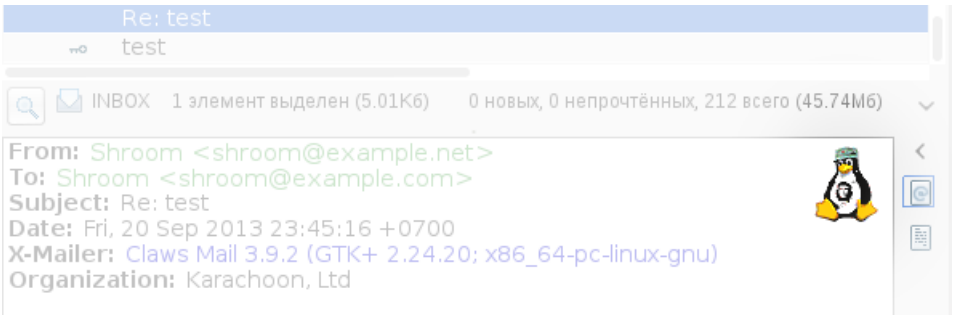
Заголовки (вернее, заголовочные поля) сообщений e-mail — это служебные поля данных, расположенные непосредственно перед телом сообщения и содержащие (как минимум) информацию об отправителе, дате отправки, получателе и пути следования письма[1]. Физически они представляют собой строки, начинающиеся с зарезервированного служебного слова и заканчивающиеся переводом строки. От тела письма заголовок отделяется пустой строкой. В принципе, всё это давно было описано в RFC 822 («Internet Message Format»), который к данному моменту окончательно устарел и был в своё время заменён на RFC 2822, который тоже уже устарел и был заменён на более новый RFC 5322[2].

†Дата обновления: 27 сентября 2013 г.

Однако, если мы внимательно почитаем эти документы, то не найдём в них упоминаний ни о каких полях (X-)Face. В общем-то, это и правильно, поскольку RFC дают лишь то, что необходимо и достаточно для организации той или иной информационной инфраструктуры. А те поля, о которых пойдёт речь дальше, являются расширениями.

Для чего нужны поля (X-)Face?

Если очень коротко, то в этих полях может содержаться портрет отправителя письма или какой-нибудь логотип. Например, в Claws Mail[3] это выглядит примерно так:



А теперь о том, как обстоят дела на самом деле…

А на самом деле о полноценном изображении речи в принципе не идёт. Во-первых, сам по себе размер картинки 48×48 точек не предполагает каких-либо эстетических изысков, поэтому портретное сходство будет весьма общим. Во-вторых, ограничение на количество цветов делает задачу передачи этого самого сходства весьма нетривиальной. Например, формат X-Face определяет возможность передачи *только* чёрно-белой картинки. Поле Face может содержать уже цветную картинку, но количество цветов в ней, скорее всего, не будет превышать 30, поскольку есть очень жёсткие ограничения на объём информации.

Однако, нет поводов для беспокойства. Автор формата X-Face Джеймс Эштон на своей страничке буквально говорит о том, что «сделанные тщательно, с использованием дизеринга, [чёрно-белые] изображения лиц на удивление узнаваемы, даже с их небольшим размером»[4]. На самом деле никто не запрещает вместо портрета использовать скан личной подписи или какой-нибудь символ. А может быть и пару-тройку слов. Тем

более, что изображение может быть цветным… Однако, описание подобных нетривиальных способов использования полей (X-)Face выходит за рамки данной статьи.

Способы создания картинок для поля Face

Поскольку поле X-Face давно можно считать морально устаревшим, сразу перейдём к генерации изображений для Face. Тем же, кто интересуется историческими артефактами, можно посоветовать поставить пакет **compface**[5], который, собственно, и предназначен для создания картинок в старом формате. Если же вы хотите моментально оценить, нужен ли вам X-Face, можно заглянуть на страницу онлайн-конвертора[6] изображений в этот формат[1].

Итак, для начала рассмотрим, какие ограничения накладываются на изображение для поля Face. Во-первых, это картинка в формате PNG. Во-вторых, размер изображения остаётся таким же, как и в X-Face и составляет 48×48 точек. В-третьих, размер файла жёстко ограничен сверху. Человек, предложивший использовать новый формат поля, Ларс Магне Ингебригтсен[7] (по совместительству — автор программы Gnus[8]) даёт довольно подробный расчёт размера этого самого png-файла, исходя из ограничений на размер поля по RFC, BASE64-кодирования и наличия дополнительной служебной информации[9]. В сухом остатке получаем 725 байт максимум… Да, я тоже сначала подумал, что это невозможно. Однако, сейчас мы с вами узнаем, как же можно упихать в эти жалкие 700 байт какую-нибудь прикольную картинку.

Авторский способ

Собственно, Ларс предлагает конвертировать любую картинку в формат Face примерно вот так:

```
djpeg face.jpg | ppmnorm | pnmscale -width 48 -height 48 |
ppmquant 7 | pnmtopng > face.png
```

Единственный параметр, который здесь можно менять, — это количество цветов, до которого ограничивается палитра выходного изображения (параметр для команды `ppmquant`). И то, в большинстве

[1] Благодарности за предоставленную ссылку — alexandrorodrigez

случаев менять его можно только в меньшую сторону, что отнюдь не способствует повышению эстетической ценности результата. Есть, правда, более гибкая команда — `pnmquant`, которой кроме количества уровней квантования можно также задавать алгоритмы, по которым будет производиться уменьшение цветов и предварительная фильтация. Тем не менее, результат оставляет желать лучшего:

Если хотите побаловаться с этими утилитами самостоятельно, их можно найти в пакетах **netpbm**[10] и **libjpeg-progs**.

Продвинутый способ

Для его использования нам понадобится откровенно тяжёлый (во всех смыслах) пакет **ImageMagick**[11]. Однако, и качество результата будет значительно выше. Сразу же замечу, что чем меньше в исходном изображении мелких деталей и различимых цветов, чем больше крупных областей, залитых одним цветом, и чем ближе его размер к нужному нам 48×48 точек, тем меньше будет размер полученной картинки в байтах и тем больше у неё будет сходства с оригиналом.

Однако, ближе к делу! Вот пример командной строки, которая из картинки 225x269 (19Кб) делает картинку объёмом всего 722 байта (обратите внимание, что перед параметрами стоит одиночный знак «–», а не двойной «- -»):

```
convert face_big.jpg -filter gaussian -define filter:blur=0.75
-resize 48x48 -gravity center -extent 48x48 -unsharp 0x1
-normalize -quantize RGB +dither -colors 32 -posterize 6 -strip
face.png
```

Если, как было указано выше, большого количества мелких деталей нет (или линейный размер исходного изображения не более, чем в 3-4 раза превышает целевой), то параметры `-filter gaussian -define filter:blur=0.75` можно опустить. Они используются в качестве ВЧ-фильтра[12] для исключения «шумовой» информации, которая будет

практически неразличима в маленьком формате, но приведёт к увеличению объёма файла.

Параметр `-resize 48x48` используется для изменения размера[13], а `-gravity center -extent 48x48` — для центрирования изображения[14] внутри квадрата 48×48. То есть, если картинка изначально квадратная, её можно не центрировать.

Команды `-unsharp 0x1 -normalize` позволяют несколько увеличить резкость после уменьшения размера[15] и повысить контрастность и насыщенность[16] за счёт нормализации[2] цветов.

Далее пойдёт некоторое количество магии[17], которая, собственно, и позволяет выкинуть бóльшую часть избыточности. Параметр `-quantize RGB`[18] задаёт цветовое пространство для дальнейших преобразования палитры. Параметр `+dither`[19] *отключает* (как это ни покажется странным со значком «+») дизеринг[3]. Параметр `-colors 32`[20] ограничивает палитру изображения 32 цветами (5 бит на пиксель). В данном случае 32 цвета — это отнюдь не догма, а отправная точка. Если объём результирующего файла оказался меньше 650 байт, можно попробовать повысить качество, ограничив палитру не 32, а 64 цветами (6 бит на пиксель). Если же по какой-то причине размер превысил жёсткое ограничение в 725 байт, количество цветов придётся уменьшить. Наконец, параметр `-posterize 6`[21] используется для дополнительного уменьшения количества цветов и увеличения площадей, залитых одним цветом, что способствует лучшему сжатию. Заметьте, что если поменять местами параметры `-posterize` и `-colors`, выигрыш в объёме будет не столь заметен. Это объясняется тем, что такая последовательность сначала пытается покрыть рисунок сеткой изолиний для выявления областей примерно одинакового цвета, а потом уменьшает количество цветов. Это приводит к тому, что найденные смежные области после сокращения палитры в общем случае не будут иметь один и тот же цвет, что приведёт к худшему сжатию. Но если сначала уменьшить количество цветов, а затем выявлять области с похожими цветами, эта стратегия даст значительный прирост количества связных областей, залитых одним и тем же цветом.

Из вышеизложенного следует логичный вопрос: если мы уменьшили

[2] Процесс растягивания гистограммы изображения с переводом 2% самых тёмных точек в чёрный цвет и 1% самых светлых — в белый.

[3] Процесс, который вносит искусственно сформированный шум, скрывая резкие переходы цветов и существенно увеличивая при этом размер файла

количество цветов постеризацией, почему бы дополнительно не снизить количество бит на пиксель до 4 (использовать 16-цветную палитру)? Давайте проверим:

```
convert face_big.jpg -filter gaussian -define filter:blur=0.75
-resize 48x48 -gravity center -extent 48x48 -unsharp 0x1
-normalize -quantize RGB +dither -colors 32 -posterize 6
-colors 16 -strip face.png
```

Получили файл объёмом 683 байта (против 722 с 32 цветами). Весьма неплохо. Если хочется попытаться чуть-чуть улучшить качество, можно постеризовать изображение не до 6, а до 7 или 8 оттенков. Однако, это не всегда приводит к заметному улучшению, зато практически всегда увеличивает размер файла. Если же размер всё ещё больше 725 байт, можно попробовать использовать `-posterize 5` (4 и ниже не рекомендую, поскольку это приведёт к серьёзным искажениям). Если и это не помогло, вероятно, следует выбрать другую картинку.

Да, кстати… Чуть не забыл! Параметр `-strip` не сто́ит недооценивать, даже несмотря на то, что он идёт последним. Он выполняет весьма важную в данном контексте функцию: убирает мета-информацию из файла. А это как минимум дата создания и модификации, которые занимают ещё пару-тройку десятков драгоценных байтов.

Использование

Теперь вернёмся к тому, ради чего, собственно, всё это и затевалось. То есть, к почтовым клиентам, которые поддерживают отображение полей (X-)Face.

Claws Mail [3] отображает эти заголовки самостоятельно, без помощи плагинов.

Gnus [8], естественно, показывает (X-)Face, поскольку автор программы решил «доработать» старый формат поля именно для этого софта.

Thunderbird [22] может показывать их с помощью плагинов; к данному моменту, вероятно, в живых остались только *Mnenhy*[23] и *Display Contact Photo*[24].

KMail [25] тоже умеет, вероятно, достаточно давно[26].

Balsa [27] не афиширует эту возможность, хотя Ларс Ингебригтсен считает[9], что там вроде бы всё в порядке.

The Bat! [28] считает, что это поле не является стандартным и, кроме того, проще прикрепить фотографию к контакту в адресной книге; соответственно, изображение не показывает (хотя был плагин для X-Face)[29].

Если отображение заголовков обычно не является проблемой, то вставить заголовок в письмо не всегда является тривиальной задачей. Хорошо, если почтовый клиент позволяет хотя бы просто добавлять свои строки в дополнительные поля. Тогда можно пропустить полученный png-файл через утилиту `base64` и вставить получившийся ~~мусор~~ закодированный файл в поле ввода, предоставленное клиентом. Но есть и такие, которые позволяют сгенерировать (X-)Face для каждого аккаунта автоматически. В качестве примера рассмотрим процедуру добавления поля Face в двух почтовых клиентах.

Claws Mail

Здесь всё просто. Всего-то нужно:

1. в главном меню программы выбрать пункт «Настройки…»,
2. далее выбрать «Редактировать учётные записи…»,
3. в появившемся окне выбрать нужную учётку (если их несколько, естественно, при необходимости придётся повторить процедуру для каждой),
4. нажать на кнопку «Изменить»,
5. в появившемся окне в списке слева выбрать пункт «Отправить»,
6. установить чекбокс «Добавить заданный заголовок» и нажать на кнопку «Изменить»,
7. в открывшемся окне в поле «Заголовок» выбрать «Face»,

8. для заполнения поля «Значение» нажать кнопку «Просмотр» и выбрать подготовленный по вышеописанной процедуре png-файл,
9. если не было ругани, нажать кнопку «Добавить»,
10. закрыть все открытые окна настроек кнопкой «OK» (или «Закрыть»).

Если после этого отправить письмо с того аккаунта, в настройках которого мы только что загрузили картинку, то в результате должно получиться примерно то же, что и на самом первом рисунке.

Thunderbird

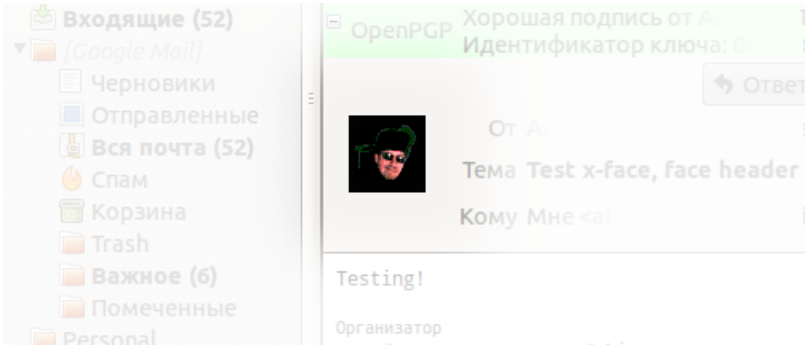
Сначала хотелось бы сказать большое спасибо alexandrorodrigez'y (OmskLUG) за то, что он нашёл время и энергию для определения работающего способа вставки[30] и отображения заголовка (X-)Face для Thunderbird.

Итак, что нужно делать:

1. зайти в Config Editor (Tools → Options → Advanced → General → Config Editor),
2. добавить поле mail.identity.idN.headers (где N — номер почтового ящика) со значением «X-Face, Face» (без кавычек, естественно)[4],
3. создать поле mail.identity.idN.header.X-Face и в качестве значения скопировать в него строку вашего 1-битного изображения, **или**...
4. создать поле mail.identity.idN.header.Face и в качестве значения скопировать в него base64-encoded строку вашего цветного изображения,
5. закрыть Config Editor

Для проверки посылаем письмо с того аккаунта, который мы только что модифицировали. Если никаких специальных плагинов не подключено, в полученном письме мы увидим в числе прочих заполненное нами поле (X-)Face. Если же стоит, например, указанный выше плагин *Display Contact Photo*[24], рядом с обычными заголовками появится картинка. Примерно вот так:

[4] Безусловно, если это поле уже существует, то «X-Face, Face» нужно добавить через запятую к существующему списку значений



То есть, можно сделать предположение (несколько необоснованно оптимистичное, впрочем), что основные почтовые клиенты вполне сносно справляются с задачей развлекать пользователя и загружать канал никому не нужной дополнительной информацией в виде нестандартных заголовочных полей. Но лично я должен признать, что это на самом деле интересная возможность.

Ссылки

1 Википедия, «Электронная почта, Заголовки письма»:
`http://ru.wikipedia.org/wiki/Электронная_почта`

2 RFC 5322, «Internet Message Format»:
`http://tools.ietf.org/html/rfc5322`

3 Claws Mail - the email client that bites!:
`http://www.claws-mail.org/`

4 James Ashton:
`http://users.cecs.anu.edu.au/~jaa/`

5 Compface:
`http://freecode.com/projects/compface`

6 Online X-Face Converter:
`http://www.dairiki.org/xface/`

7 Lars Magne Ingebrigtsen:
`http://quimby.gnus.org/lmi/lmi.html`

8 Gnus, the Emacs newsreader:
`http://www.gnus.org/`

9 The Face Header:
`http://quimby.gnus.org/circus/face/`

10 Netpbm home page:
`http://netpbm.sourceforge.net/`

11 ImageMagick: Convert, Edit, Or Compose Bitmap Images:
`http://www.imagemagick.org/`

12 ImageMagick v6 Examples — Resampling Filters:
`http://www.imagemagick.org/Usage/filter/`

13 ImageMagick v6 Examples — Resize or Scaling (General Techniques):
`http://www.imagemagick.org/Usage/resize/`

14 ImageMagick v6 Examples — Cutting and Bordering:
`http://www.imagemagick.org/Usage/crop/`

15 Unsharped Resizing (USM) — Photoshop Resize Technique:
`http://www.imagemagick.org/Usage/resize/#resize_unsharp`

16 Normalize (auto-level stretching):
`http://www.imagemagick.org/Usage/color_mods/#normalize`

17 ImageMagick v6 Examples — Color Quantization and Dithering:
`http://www.imagemagick.org/Usage/quantize/`

18 Color Quantization and ColorSpace:
`http://www.imagemagick.org/Usage/quantize/#quantize`

19 Error Correction Dithering:
`http://www.imagemagick.org/Usage/quantize/#dither_error`

20 Color Quantization Operator:
`http://www.imagemagick.org/Usage/quantize/#colors`

21 Posterize, Recolor using a Uniform Color Map:
`http://www.imagemagick.org/Usage/quantize/#posterize`

22 Thunderbird:
`http://www.mozilla.org/thunderbird/`

23 Mnenhy:
`https://addons.mozilla.org/thunderbird/addon/mnenhy/`

24 Display Contact Photo:
`https://addons.mozilla.org/thunderbird/addon/display-contact-photo/`

25 KMail – Mail Client:
`http://www.kde.org/applications/internet/kmail/`

26 Add Support for displaying Face: header:
`https://bugs.kde.org/show_bug.cgi?id=109836`

27 Balsa – An e-mail client for GNOME:
`http://balsa.gnome.org/`

28 The Bat! Email Client:
`http://www.ritlabs.com/en/products/thebat/`

29 RITLabs Forum «The Bat! - Configuring the E-mail Client» Topic «Custom header (X-Face and Face)»:
`http://www.ritlabs.com/de/forum.backUP/read.php?FID=4&TID=8671`

30 Mozillazine: Custom headers:
`http://kb.mozillazine.org/Custom_headers`

Спасибо за внимание!