**PNG**

Portable network graphics – растровый формат хранения графической информации, использующий сжатие без потерь по алгоритму Deflate.

Днём рождения PNG можно считать 4 января 1995 года, когда Т. Боутелл предложил в ряде конференций Usenet создать свободный формат, который был бы не хуже GIF. И уже через три недели после публикации идеи были разработаны четыре версии нового формата. Вначале он имел название PBF (Portable Bitmap Format), а нынешнее имя получил 23 января 1995 года. Уже в декабре того же года спецификация PNG версии 0.92 была рассмотрена консорциумом W3C, а с выходом 1 октября 1996 года версии 1.0 PNG был рекомендован в качестве полноправного сетевого формата.

Формат PNG спроектирован для замены устаревшего, платного и более простого формата GIF, а также, в некоторой степени, для замены значительно более сложного формата TIFF. Формат PNG позиционируется прежде всего для использования в Интернете и редактирования графики.

Преимущества перед GIF:

1. Цвета. PNG в основном использует полутоновое изображение (с глубиной цвета 16 бит), цветное индексированное изображение (палитра 8 бит для цветов глубиной 24 бит (16 777 216)) и полноцветное изображение (с глубиной цвета 48 бит (281 474 976 710 656)).

GIF использует в лучшем случае 8-битный цвет.

1. Полноценная поддержка прозрачности(до 16 бит).
2. Гамма-коррекции. Позволяет прикрепить к изображению настройки его отображения, для того чтобы на разных мониторах изображение отображалось точно так же, как и у автора. Данные о настройках дисплея, видеоплаты и программного обеспечения (информация о гамме) сохраняются в файле вместе с самим изображением, что и обеспечивает идентичность копии оригиналу при переносе на другой компьютер. По сравнению с камерами мы более чувствительны к малейшим изменениям тёмных оттенков и менее чувствительны к достаточно большим изменениям в ярких тонах. Для такой странности есть свои биологические причины: это позволяет нашему зрению работать в более широком диапазоне освещённости. В противном случае типичный диапазон яркостей, с которым мы сталкиваемся на улице, был бы невыносим.
3. Двумерная схема для многоуровневых изображений (Adam7) в отличие от чересстрочной развёртки в GIF.
4. Более быстрый алгоритм Deflate против LZW.
5. Метаданные (сжатые или несжатые) (APNG (анимация и цветная прозрачность)).
6. Формат, свободный от патентов.

**Области и особенности применения.**

«PNG, в основном, используется для создания 24-битных RGB сформированных изображений, например, картин с рассчитанным освещением с минимальным числом текстур или математических объектов. Они все обладают искусственно сглаженными цветовыми переходами, которые хорошо сжимаются при помощи PNG-фильтров. Некоторые фракталы могут вести себя таким же образом, но у многих из самых лучших примеров имеется достаточно «зашумленных» областей, которые сжимаются весьма слабо». Greg Roelofs.

Для веб-страниц вполне можно использовать PNG8 (8 битный формат), с помощью которого дизайнеры могут заменить существующие GIF-изображения. У PNG также может быть альфа-значение для каждого цвета в палитре, которое, фактически, означает, что используется RGBA-палитра, а не RGB-XOR-маска, как GIF. Это позволяет варьировать прозрачность цвета в больших пределах, сохраняя преимущества 8-битного изображения перед 32-битным.

Многослойный PNG-файл также может быть отображен на экране по загрузке только 25% всего файла, в то время как GIF требует загрузки 50% размера перед распознаванием. За исключением весьма редких случаев замена GIF-изображений на PNG-эквиваленты способна существенно уменьшить их размер.

PNG работает лучше с большими однотонными областями. Чтобы увеличить степень сжатия, стоит уменьшить количество «шума» в ваших картинках для увеличения размеров однотонных областей. Уменьшайте области размывания, хотя для изображений с градиентами размывание позволяет использовать меньшую глубину цвета. Избегайте сглаживания текста для уменьшения числа цветов, которые для него необходимы, в финальном изображении.

По возможности, следует уменьшать число цветов в исходном изображении, если в процессе разработки этот фактор можно контролировать. Стоит также избегать использования многослойных PNG-изображений для уменьшения размера файла, так как семипроходная многослойная схема может добавить от 20% до 35% к размеру PNG- файла. Наконец, можно применять специальное программное обеспечение для оптимизации PNG-изображений, которое специально разрабатывается с этой целью, например, pngout или pngcrush.

При необходимости сжать горизонтальный или вертикальный градиент PNG в обоих случаях выдаст более легковесный результат.

Формат PNG обладает более высокой степенью сжатия для файлов с большим количеством цветов, чем GIF, но разница составляет около 5-25 %, что недостаточно для абсолютного преобладания формата, так как небольшие 2-16-цветные файлы формат GIF сжимает с не меньшей эффективностью.

PNG является хорошим форматом для редактирования изображений, даже для хранения промежуточных стадий редактирования, так как восстановление и пересохранение изображения проходят без потерь в качестве. Также, в отличие, например, от TIFF, спецификация PNG не позволяет авторам реализаций выбирать, какие возможности они собираются реализовать. Поэтому любое сохранённое изображение PNG может быть прочитано в любом другом приложении, поддерживающем PNG.

Различные реализации алгоритма Deflate дают разную степень сжатия, поэтому были созданы программы для пережатия изображений с несколькими вариантами настроек в целях получения наилучшего сжатия — например, форк pngcrush OptiPNG и advpng из комплекта AdvanceCOMP (использует 7-Zip).

Раньше узким местом формата в работе с браузерами были:

* частичная прозрачность (альфа-канал);
* поддержка прозрачности в палитре;
* гамма-коррекция;
* поддержка расширений PNG с анимацией;
* цветовая коррекция (ICC).

В настоящее время таких проблем не наблюдается в связи с широким использованием формата в интернете.

**Алгоритм.**

1. Pass extraction (Interlace or no Interlace). Переплетение. Если необходимо. Разбивает исходное изображение на 7 частей для последующего отдельного сжатия. Используется алгоритм Adam7.

1. Scanline serialization. Линеаризация изображения.
2. Filtering. Фильтрация.

PNG предварительно фильтрует данные по конкретному изображению при помощи предсказательных функций. Одной из них является «Up», которая ищет похожие наборы данных в вертикальных шаблонах для полноцветных PNG. PNG с индексированными цветами (8 битов или меньше) обычно не выигрывает от использования фильтрации, поэтому стоит использовать «None», если есть возможность для выбора. Для полноцветных или серых изображений лучше применять «Adaptive» алгоритм.

1. Compression. Сжатие.
2. Chunking. Чанкинг. Разбиение сжатых данных на кусочки и сборка в файл согласно структуре.

**Adam7.**

Adam7 — алгоритм образования чересстрочного изображения, специфичный для формата PNG. Чересстрочные изображения разбиваются на семь меньших, накладывая на каждый участок 8x8 следующую маску:

1 6 4 6 2 6 4 6

7 7 7 7 7 7 7 7

5 6 5 6 5 6 5 6

7 7 7 7 7 7 7 7

3 6 4 6 3 6 4 6

7 7 7 7 7 7 7 7

5 6 5 6 5 6 5 6

7 7 7 7 7 7 7 7

Числа соответствуют номеру маленького изображения, куда попадёт пиксель.

В этом алгоритме используется семь проходов в двух измерениях, в отличие от подобных алгоритмов, используемых в GIF и использующих только четыре прохода по вертикали. Это даёт более высокую скорость развертки за меньшее число операций, чем в алгоритме бикубической интерполяции.

Adam7 получил своё название в честь Адама Костелло, который предложил этот метод 30 января 1995, расширив существовавший алгоритм с пятью проходами Ли Даниэля Крокера:

1 4 2 4

5 5 5 5

3 4 3 4

5 5 5 5

**Фильтры.**

1. None. Без фильтрации.
2. Sub. Разница между текущим пикселем и предыдущим (левым).

Filt(x) = Orig(x) - Orig(b)

1. Up. То же, но с верхним пикселем.

Filt(x) = Orig(x) - Orig(b)

1. Average. Арифметическое среднее между левым и верхним.

Filt(x) = Orig(x) - floor((Orig(a) + Orig(b)) / 2)

1. Paeth. Предсказательная функция. Похожа на предыдущий фильтр.

Filt(x) = Orig(x) - PaethPredictor(Orig(a), Orig(b), Orig(c))

function PaethPredictor (a, b, c)

begin

; a = left, b = above, c = upper left

p := a + b - c ; initial estimate

pa := abs(p - a) ; distances to a, b, c

pb := abs(p - b)

pc := abs(p - c)

; return nearest of a,b,c, breaking ties in order a,b,c.

if pa <= pb AND pa <= pc then return a

else if pb <= pc then return b

else return c

end

**Сжатие.**

Deflate — это алгоритм сжатия без потерь, который использует комбинацию алгоритма LZ77 и алгоритма Хаффмана. Изначально он был описан Филом Кацом для 2-й версии своей утилиты для создания архивов PKZIP, который впоследствии был определён в RFC 1951.

Deflate считается свободным от всех существующих патентов, и пока патент на LZW (который используется в формате GIF) оставался в силе, это привело к использованию deflate в файлах, сжимаемых gzip, и изображениях в формате PNG вдобавок к формату ZIP, для которого Кац изначально его спроектировал.

Deflate поток содержит серии блоков. Перед каждым блоком находится 3-битовый заголовок:

1 бит: флаг последнего блока.

1: блок последний.

0: не последний блок.

2 бита: метод, с помощью которого были закодированы данные.

00: данные не были закодированы. (В блоке находятся выходные данные)

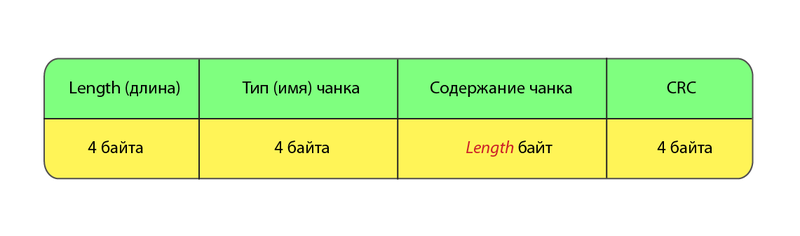
01: данные, закодированные с помощью метода статического Хаффмана.

10: данные, закодированные с помощью метода динамического Хаффмана.

11: зарезервированное значение.(ошибка)

Большая часть блоков кодируется с помощью метода 10(динамический Хаффман), который предоставляет оптимизированное дерево кодов Хаффмана для каждого нового блока. Инструкции для создания дерева кодов Хаффмана следуют непосредственно за заголовком блока.

**Чанкинг.**

****

Критические чанки

IHDR — заголовок файла, содержит основную информацию о изображении. Обязан быть первым чанком.

PLTE — палитра, список цветов.

IDAT — содержит, собственно, изображение. Рисунок можно разбить на несколько IDAT чанков, для потоковой передачи. В каждом файле должен быть хотя бы один IDAT чанк.

IEND — завершающий чанк, обязан быть последним в файле.

Вспомогательные чанки

bKGD — этот чанк задает основной фоновый цвет.

cHRM используется для задания CIE 1931 цветового пространства.

gAMA — определяет гамму.

hIST — в этом чанке может храниться гистограмма или общее содержание каждого цвета в изображении.

iCCP — цветовой профиль ICC

iTXt — содержит текст в UTF-8, возможно сжатый, с необязательной языковой меткой. iTXt чанк с ключевым словом 'XML:com.adobe.xmp' может содержать Extensible Metadata Platform (XMP).

pHYs — содержит предполагаемый размер пикселя и/или отношение сторон изображения.

sBIT (significant bits) — определяет «цветовую точность» (color-accuracy) изображения (черно-белое, полный цвет, черно-белое с прозрачностью и т.д.), для более простого декодирования.

sPLT — предлагает палитру для использования, если полный спектр цветов недоступен.

sRGB — свидетельствует о использовании стандартной sRGB схемы.

sTER — индикатор стереоскопических изображений.

tEXt — может содержать текст в ISO/IEC 8859-1 формате, с одной name=value парой для каждого чанка.

tIME — хранит дату последнего изменения изображения.

tRNS — содержит информацию о прозрачности.

zTXt — сжатый текст, с теми же ограничениям, что и tEXt.

IHDR. Блок данных в IHDR содержит следующие поля:

Ширина, 4 байта

Высота, 4 байта

Битовая глубина (bit depth), количество бит на каждый сэмпл(не пиксель), 1 байт

Тип цвета, состоит из 3 флагов 1 (палитра), 2 (цвет, не монохромное изображение), and 4 (есть альфа-канал), 1 байт

Метод сжатия. Всегда должно быть 0 — сжатие по алгоритму deflate. Если значение отлично от 0, чанк считается нераспознанным, и декодер рапортует об ошибке. 1 байт

Метод фильтрации. Всегда 0. 1 байт

Interlace. Порядок передачи данных. 0 (no interlace) и 1 (Adam7 interlace). 1 байт

IEND. Сигнализирует о конце файла, блок данных этого чанка не содержит ничего.

IDAT. Содержит данные, закодированные, в соответствии с полем метода сжатия в заголовке.

**Ссылки:**

История

<http://www.libpng.org/pub/png/slashpng-1999.html>

Спецификация

<http://www.w3.org/TR/PNG/>

<http://tools.ietf.org/html/rfc2083>

<http://www.libpng.org/pub/png/spec/1.2/PNG-Contents.html>

<http://allo.usaaa.ru/workshop/samples/pngread/pngread.html> ([http://web.archive.org/web/20110829133205/http://allo.usaaa.ru/workshop/samples/pngread/pngread.htm](http://web.archive.org/web/20110829133205/http:/allo.usaaa.ru/workshop/samples/pngread/pngread.htm)архив)

Обзоры и статьи

<http://www.artlebedev.ru/tools/technogrette/img/png-1/>

<http://www.cwpro.ru/seo-optimisation/against-png-gif.html>

<http://habrahabr.ru/post/130472/>

<http://habrahabr.ru/post/113393/>

<http://en.wikipedia.org/wiki/Portable_Network_Graphics>

Алгоритм Deflate

<http://compression.ru/download/articles/lz/mihalchik_deflate_decoding.html>