WGet - программа для загрузки файлов и скачивания сайта целиком.

Скачать WGet для Windows можно здесь

Пришедшая из мира Linux, свободно распространяемая утилита Wget позволяет скачивать как отдельные файлы из интернета, так и сайты целиком, следуя по ссылкам на веб-страницах.

Чтобы получить подсказку по параметрам WGet, наберите команду man wget в Linux или wget.exe --help в Windows.

Допустим, мы хотим создать полную копию сайта www.site.com на своем диске. Для этого открываем командную строку (Wget - утилита консольная) и пишем такую команду:

```
wget.exe -r -l10 -k -p -E -nc http://www.site.com
```

WGET рекурсивно (параметр -r) обойдет каталоги и подкаталоги на удалённом сервере включая css-стили(-k) с максимальной глубиной рекурсии равной десяти (-l), а затем заменить в загруженных HTML-документах абсолютные ссылки на относительные (-k) и расширения на html(-E) для последующего локального просмотра скачанного сайта. При повторном скачивании не будут лица и перезаписываться повторы(-nc). К сожалению, внутренние стили и картинки указанные в стилях не скачиваются

Если предполагается загрузка с сайта какого-либо одного каталога (со всеми вложенными в него папками), то логичнее будет включить в командную строку параметр -np. Он не позволит утилите при поиске файлов подниматься по иерархии каталогов выше указанной директории:

```
wget.exe -r -l10 -k http://www.site.com -np
```

Если загрузка данных была случайно прервана, то для возобновления закачки с места останова, необходимо в команду добавить ключ -c:

```
wget.exe -r -l10 -k http://www.site.com -c
```

По умолчанию, всё скаченное сохраняется в рабочей директории утилиты. Определить другое месторасположение копируемых файлов поможет параметр -P:

```
wget.exe -r -l10 -k http://www.site.com -P c:\internet\files
```

Наконец, если сетевые настройки вашей сети предполагают использование прокси-сервера, то его настройки необходимо сообщить программе. См. Конфигурирование WGET

```
wget -m -k -nv -np -p --user-agent="Mozilla/5.0 (compatible; Konqueror/3.0.0/10; Linu
```

Загрузка всех URL, указанных в файле FILE:

```
wget -i FILE
```

Скачивание файла в указанный каталог (-P):

```
wget -P /path/for/save ftp://ftp.example.org/some_file.iso
```

Использование имени пользователя и пароля на FTP/HTTP (вариант 1):

```
wget ftp://login:password@ftp.example.org/some_file.iso
```

Использование имени пользователя и пароля на FTP/HTTP (вариант 2):

```
wget --user=login --password=password ftp://ftp.example.org/some_file.iso
```

Скачивание в фоновом режиме (-b):

```
wget -b ftp://ftp.example.org/some_file.iso
```

Продолжить (-c continue) загрузку ранее не полностью загруженного файла:

```
wget -c http://example.org/file.iso
```

Скачать страницу с глубиной следования 10, записывая протокол в файл log:

```
wget -r -l 10 http://example.org/ -o log
```

Скачать содержимое каталога http://example.org/~luzer/my-archive/ и всех его подкаталогов, при этом не поднимаясь по иерархии каталогов выше:

```
wget -r --no-parent http://example.org/~luzer/my-archive/
```

Для того, чтобы во всех скачанных страницах ссылки преобразовывались в относительные для локального просмотра, необходимо использовать ключ -k:

```
wget -r -l 10 -k http://example.org/
```

Также поддерживается идентификация на сервере:

```
wget --save-cookies cookies.txt --post-data 'user=foo&password=bar' http://example.or
```

Скопировать весь сайт целиком:

```
wget -r -l0 -k http://example.org/
```

Например, не загружать zip-архивы:

```
wget -r -R «*.zip» http://freeware.ru
```

Залогиниться и скачать файлик ключа

```
@echo off
wget --save-cookies cookies.txt --post-data "login=ТВОЙЛОГИН&password=ТВОЙПАРОЛЬ" htt
wget --load-cookies cookies.txt "http://beta.drweb.com/files/?p=win%%2Fdrweb32-betate
```

Внимание! Регистр параметров WGet различен!

Базовые ключи запуска

-V
--version
Отображает версию Wget.

-h
--help
Выводит помощь с описанием всех ключей командной строки Wget.

-b
--background
Переход в фоновый режим сразу после запуска. Если выходной файл не задан -o, выход перенаправляется в wget-log.

-e command
--execute command
Выполнить command, как если бы она была частью файла .wgetrc. Команда, запущенная таким образом, будет выполнена после команд в .wgetrc, получая приоритет над ними. Для задания более чем одной команды wgetrc используйте несколько ключей -e.

Протоколирование и ключи входного файла

-o logfile
--output-file=logfile
Протоколировать все сообщения в logfile. Обычно сообщения выводятся в standard error.

-a logfile
--append-output=logfile
Дописывать в logfile. То же, что -o, только logfile не перезаписывается, а дописывается. Если logfile не существует, будет создан новый файл.

-d
--debug
Включает вывод отладочной информации, т.е. различной информации, полезной для разработчиков Wget при некорректной работе. Системный администратор мог выбрать сборку Wget без поддержки отладки, в этом случае -d работать не будет. Помните, что сборка с поддержкой отладки всегда безопасна - Wget не будет выводить отладочной информации, пока она явно не затребована через -d.

-q
--quiet
Выключает вывод Wget.

-v
--verbose
Включает подробный вывод со всей возможной информацией. Задано по умолчанию.

-nv
--non-verbose
Неподробный вывод - отключает подробности, но не замолкает совсем (используйте -q для этого), отображаются сообщения об ошибках и основная информация.

-i file
--input-file=file
Читать URL из входного файла file, в этом случае URL не обязательно указывать в командной строке.

Если адреса URL указаны в командной строке и во входном файле, первыми будут запрошены адреса из командной строки. Файл не должен (но может) быть документом HTML - достаточно последовательного списка адресов URL. Однако, при указании --force-html входной файл будет считаться html. В этом случае могут возникнуть проблемы с относительными ссылками, которые можно решить указанием <base href="url"> внутри входного файла или --base=url в командной строке.

-F

--force-html

При чтении списка адресов из файла устанавливает формат файла как HTML. Это позволяет организовать закачку по относительным ссылкам в локальном HTML-файле при указании <base href="url"> внутри входного файла или --base=url в командной строке.

-B URL

--base=URL

Используется совместно с -F для добавления URL к началу относительных ссылок во входном файле, заданном через -i.

Ключи скачивания

--bind-address=ADDRESS

При открытии клиентских TCP/IP соединений bind() на ADDRESS локальной машины. ADDRESS может указываться в виде имени хоста или IP-адреса. Этот ключ может быть полезен, если машине выделено несколько адресов IP.

-t number

--tries=number

Устанавливает количество попыток в number. Задание 0 или inf соответствует бесконечному числу попыток. По умолчанию равно 20, за исключением критических ошибок типа "в соединении отказано" или "файл не найден" (404), при которых попытки не возобновляются.

-O file

--output-document=file

Документы сохраняются не в соответствующие файлы, а конкатенируются в файл с именем file. Если file уже существует, то он будет перезаписан. Если в качестве file задано -, документы будут выведены в стандартный вывод (отменяя -k). Помните, что комбинация с -k нормально определена только для скачивания одного документа.

-nc

--no-clobber

Если файл скачивается более одного раза в один и тот же каталог, то поведение Wget определяется несколькими ключами, включая -nc. В некоторых случаях локальный файл будет затёрт или перезаписан при повторном скачивании, в других - сохранён.

При запуске Wget без -N, -nc или -r скачивание того же файла в тот же каталог приводит к тому, что исходная копия файла сохраняется, а новая копия записывается с именем file.1. Если файл скачивается вновь, то третья копия будет названа file.2 и т.д. Если указан ключ -nc, такое поведение подавляется, Wget откажется скачивать новые копии файла. Таким образом, "no-clobber" неверное употребление термина в данном режиме - предотвращается не затирание файлов (цифровые суффиксы уже предотвращали затирание), а создание множественных копий.

При запуске Wget с ключом -r, но без -N или -nc, перезакачка файла приводит к перезаписыванию на место старого. Добавление -nc предотвращает такое поведение, сохраняя исходные версии файлов и игнорируя любые новые версии на сервере.

При запуске Wget с ключом -N, с или без -r, решение о скачивании новой версии файла зависит от локальной и удалённой временных отметок и размера файла. -nc не может быть указан вместе с -N.

При указании -nc файлы с расширениями .html и .htm будут загружаться с локального диска и обрабатываться так, как если бы они были скачаны из сети.

-c

--continue

Продолжение закачки частично скачанного файла. Это полезно при необходимости завершить закачку, начатую другим процессом Wget или другой программой. Например:

```
wget -c ftp://htmlweb.ru/ls-lR.Z
```

Если в текущем каталоге имеется файл ls-lR.Z, то Wget будет считать его первой частью удалённого файла и запросит сервер о продолжении закачки с отступом от начала, равному длине локального файла.

Нет необходимости указывать этот ключ, чтобы текущий процесс Wget продолжил закачку при пи потере связи на полпути. Это изначальное поведение. -c влияет только на закачки, начатые до текущего процесса Wget, если локальные файлы уже существуют.

Без -c предыдущий пример сохранит удалённый файл в ls-lR.Z.1, оставив ls-lR.Z без изменения.

Начиная с версии Wget 1.7, при использовании -c с непустым файлом, Wget откажется начинать закачку сначала, если сервер не поддерживает закачку, т.к. это привело бы к потере скачанных данных. Удалите файл, если вы хотите начать закачку заново.

Также начиная с версии Wget 1.7, при использовании -c для файла равной длины файлу на сервере Wget откажется скачивать и выведет поясняющее сообщение. То же происходит, если удалённый файл меньше локального (возможно, он был изменён на сервере с момента предыдущей попытки) - т.к. "продолжение" в данном случае бессмысленно, скачивание не производится.

С другой стороны, при использовании -c локальный файл будет считаться недокачанным, если длина удалённого файла больше длины локального. В этом случае (длина(удалённая) - длина(локальная)) байт будет скачано и приклеено в конец локального файла. Это ожидаемое поведение в некоторых случаях: например, можно использовать -c для скачивания новой порции собранных данных или лог-файла.

Однако, если файл на сервере был изменён, а не просто дописан, то вы получите испорченный файл. Wget не обладает механизмами проверки, является ли локальный файл начальной частью удалённого файла. Следует быть особенно внимательным при использовании -c совместно с -r, т.к. каждый файл будет считаться недокачанным.

Испорченный файл также можно получить при использовании -c с кривым HTTP прокси, который добавляет строку тима "закачка прервана". В будущих версиях возможно добавление ключа "откат" для исправления таких случаев.

Ключ -c можно использовать только с FTP и HTTP серверами, которые поддерживают заголовок Range.

--progress=type

Выбор типа индикатора хода закачки. Возможные значения: "dot" и "bar".

Индикатор типа "bar" используется по умолчанию. Он отображает ASCII полосу хода загрузки (т.н. "термометр"). Если вывод не в TTY, то по умолчанию используется индикатор типа "dot".

Для переключения в режим "dot" укажите --progress=dot. Ход закачки отслеживается и выводится на экран в виде точек, где каждая точка представляет фиксированный размер скачанных данных.

При точечной закачке можно изменить стиль вывода, указав dot:style. Различные стили определяют различное значение для одной точки. По умолчанию одна точка представляет 1K, 10 точек образуют кластер, 50 точек в строке. Стиль binary является более "компьютер"-ориентированным - 8K на точку, 16 точек на кластер и 48 точек на строку (384K в строке). Стиль mega наиболее подходит для

скачивания очень больших файлов - каждой точке соответствует 64K, 8 точек на кластер и 48 точек в строке (строка соответствует 3M).
Стиль по умолчанию можно задать через .wgetrc. Эта установка может быть переопределена в командной строке. Исключением является приоритет "dot" над "bar", если вывод не в TTY. Для непременного использования bar укажите --progress=bar:force.

-N
--timestamping
Включает использование временных отметок.

-S
--server-response
Вывод заголовков HTTP серверов и ответов FTP серверов.

--spider
При запуске с этим ключом Wget ведёт себя как сетевой паук, он не скачивает страницы, а лишь проверяет их наличие. Например, с помощью Wget можно проверить закладки:

```
wget --spider --force-html -i bookmarks.html
```

Эта функция требует большой доработки, чтобы Wget достиг функциональности реальных сетевых пауков.

-T seconds
--timeout=seconds
Устанавливает сетевое время ожидания в seconds секунд. Эквивалентно одновременному указанию --dns-timeout, --connect-timeout и --read-timeout.
Когда Wget соединяется или читает с удалённого хоста, он проверяет время ожидания и прерывает операцию при его истечении. Это предотвращает возникновение аномалий, таких как повисшее чтение или бесконечные попытки соединения. Единственное время ожидания, установленное по умолчанию, - это время ожидания чтения в 900 секунд. Установка времени ожидания в 0 отменяет проверки.
Если вы не знаете точно, что вы делаете, лучше не устанавливать никаких значений для ключей времени ожидания.

--dns-timeout=seconds
Устанавливает время ожидания для запросов DNS в seconds секунд. Незавершённые в указанное время запросы DNS будут неуспешны. По умолчанию никакое время ожидания для запросов DNS не устанавливается, кроме значений, определённых системными библиотеками.

--connect-timeout=seconds
Устанавливает время ожидания соединения в seconds секунд. TCP соединения, требующие большего времени на установку, будут отменены. По умолчанию никакое время ожидания соединения не устанавливается, кроме значений, определённых системными библиотеками.

--read-timeout=seconds
Устанавливает время ожидания чтения (и записи) в seconds секунд. Чтение, требующее большего времени, будет неуспешным. Значение по умолчанию равно 900 секунд.

--limit-rate=amount
Устанавливает ограничение скорости скачивания в amount байт в секунду. Значение может быть выражено в байтах, килобайтах с суффиксом k или мегабайтах с суффиксом m. Например, --limit-rate=20k установит ограничение скорости скачивания в 20KB/s. Такое ограничение полезно, если по какой-либо причине вы не хотите, чтобы Wget не утилизировал всю доступную полосу пропускания. Wget реализует ограничение через sleep на необходимое время после сетевого чтения, которое заняло меньше времени, чем указанное в ограничении. В итоге такая стратегия приводит к замедлению

скорости TCP передачи приблизительно до указанного ограничения. Однако, для установления баланса требуется определённое время, поэтому не удивляйтесь, если ограничение будет плохо работать для небольших файлов.

-w seconds

--wait=seconds

Ждать указанное количество seconds секунд между закачками. Использование этой функции рекомендуется для снижения нагрузки на сервер уменьшением частоты запросов. Вместо секунд время может быть указано в минутах с суффиксом m, в часах с суффиксом h или днях с суффиксом d.
Указание большого значения полезно, если сеть или хост назначения недоступны, так чтобы Wget ждал достаточное время для исправления неполадок сети до следующей попытки.

--waitretry=seconds

Если вы не хотите, чтобы Wget ждал между различными закачками, а только между попытками для сорванных закачек, можно использовать этот ключ. Wget будет линейно наращивать паузу, ожидая 1 секунду после первого сбоя для данного файла, 2 секунды после второго сбоя и так далее до максимального значения seconds. Таким образом, значение 10 заставит Wget ждать до (1 + 2 + ... + 10) = 55 секунд на файл. Этот ключ включён по умолчанию в глобальном файле wgetrc.

--random-wait

Некоторые веб-сайты могут анализировать логи для идентификации качалок, таких как Wget, изучая статистические похожести в паузах между запросами. Данный ключ устанавливает случайные паузы в диапазоне от 0 до 2 * wait секунд, где значение wait указывается ключом --wait. Это позволяет исключить Wget из такого анализа. В недавней статье на тему разработки популярных пользовательских платформ был представлен код, позволяющий проводить такой анализ на лету. Автор предлагал блокирование подсетей класса C для блокирования программ автоматического скачивания, несмотря на возможную смену адреса, назначенного DHCP. На создание ключа --random-wait подвигла эта больная рекомендация блокировать множество невиновных пользователей по вине одного.

-Y on/off

--proxy=on/off

Включает или выключает поддержку прокси. Если соответствующая переменная окружения установлена, то поддержка прокси включена по умолчанию.

-Q quota

--quota=quota

Устанавливает квоту для автоматических скачиваний. Значение указывается в байтах (по умолчанию), килобайтах (с суффиксом k) или мегабайтах (с суффиксом m). Квота не влияет на скачивание одного файла. Так если указать wget -Q10k ftp://htmlweb.ru/ls-lR.gz, файл ls-lR.gz будет скачан целиком. То же происходит при указании нескольких URL в командной строке. Квота имеет значение при рекурсивном скачивании или при указании адресов во входном файле. Т.о. можно спокойно указать wget -Q2m -i sites - закачка будет прервана при достижении квоты. Установка значений 0 или inf отменяет ограничения.

--dns-cache=off

Отключает кеширование запросов DNS. Обычно Wget запоминает адреса, запрошенные в DNS, так что не приходится постоянно запрашивать DNS сервер об одном и том же (обычно небольшом) наборе адресов. Этот кэш существует только в памяти. Новый процесс Wget будет запрашивать DNS снова. Однако, в некоторых случаях кеширование адресов не желательно даже на короткий период запуска такого приложения как Wget. Например, секоторые серверы HTTP имеют динамически выделяемые адреса IP, которые изменяются время от времени. Их записи DNS обновляются при каждом изменении. Если закачка Wget с такого хоста прерывается из-за смены адреса IP, Wget повторяет попытку скачивания, но (из-

за кеширования DNS) пытается соединиться по старому адресу. При отключенном кешировании DNS Wget будет производить DNS-запросы при каждом соединении и, таким образом, получать всякий раз правильный динамический адрес. Если вам не понятно приведённое выше описание, данный ключ вам, скорее всего, не понадобится.

--restrict-file-names=mode

Устанавливает, какие наборы символов могут использоваться при создании локального имени файла из адреса удалённого URL. Символы, запрещённые с помощью этого ключа, экранируются, т.е. заменяются на %HH, где HH - шестнадцатиричный код соответствующего символа. По умолчанию Wget экранирует символы, которые не богут быть частью имени файла в вашей операционной системе, а также управляющие символы, как правило непечатные. Этот ключ полезен для смены умолчания, если вы сохраняете файл на неродном разделе или хотите отменить экранирование управляющих символов. Когда mode установлен в "unix", Wget экранирует символ / и управляющие символы в диапазонах 0-31 и 128-159. Это умолчание для Ос типа Unix. Когда mode установлен в "windows", Wget экранирует символы \, |, /, :, ?, ", *, <, > и управляющие символы в диапазонах 0-31 и 128-159. Дополнительно Wget в Windows режиме использует + вместо : для разделения хоста и порта в локальных именах файлов и @ вместо ? для отделения запросной части имени файла от остального. Таким образом, адрес URL, сохраняемый в Unix режиме как www.htmlweb.ru:4300/search.pl?input=blah, в режиме Windows будет сохранён как www.htmlweb.ru+4300/search.pl@input=blah. Этот режим используется по умолчанию в Windows. Если к mode добавить, nocontrol, например, unix,nocontrol, экранирование управляющих символов отключается. Можно использовать --restrict-file-names=nocontrol для отключения экранирования управляющих символов без влияния на выбор ОС-зависимого режима экранирования служебных символов.

Ключи каталогов

-nd

--no-directories

Не создавать структуру каталогов при рекурсивном скачивании. С этим ключом все файлы сохраняются в текущий каталог без затирания (если имя встречается больше одного раза, имена получат суффикс .n).

-x

--force-directories

Обратное -nd - создаёт структуру каталогов, даже если она не создавалась бы в противном случае. Например, wget -x http://htmlweb.ru/robots.txt сохранит файл в htmlweb.ru/robots.txt.

-nH

--no-host-directories

Отключает создание хост-каталога. По умолчания запуск Wget -r http://htmlweb.ru/ создаст структуру каталогов, начиная с htmlweb.ru/. Данный ключ отменяет такое поведение.

--protocol-directories

Использовать название протокола как компонент каталога для локальный файлов. Например, с этим ключом wget -r http://host сохранит в http/host/... вместо host/....

--cut-dirs=number

Игнорировать number уровней вложенности каталогов. Это полезный ключ для чёткого управления каталогом для сохранения рекурсивно скачанного содержимого. Например, требуется скачать каталог ftp://htmlweb.ru/pub/xxx/. При скачивании с -r локальная копия будет сохранена в ftp.htmlweb.ru/pub/xxx/. Если ключ -nH может убрать ftp.htmlweb.ru/ часть, остаётся ненужная pub/xemacs. Здесь на помощь приходит --cut-dirs; он заставляет Wget закрывать глаза на number удалённых подкаталогов. Ниже приведены несколько рабочих примеров --cut-dirs.

```
No options        -> ftp.htmlweb.ru/pub/xxx/
-nH               -> pub/xxx/
-nH --cut-dirs=1  -> xxx/
-nH --cut-dirs=2  -> .
--cut-dirs=1      -> ftp.htmlweb.ru/xxx/
```

Если вам нужно лишь избавиться от структуры каталогов, то этот ключ может быть заменён комбинацией -nd и -P. Однако, в отличии от -nd, --cut-dirs не теряет подкаталоги - например, с -nH --cut-dirs=1, подкаталог beta/ будет сохранён как xxx/beta, как и ожидается.

-P prefix

--directory-prefix=prefix

Устанавливает корневой каталог в prefix. Корневой каталог - это каталог, куда будут сохранены все файлы и подкаталоги, т.е. вершина скачиваемого дерева. По умолчанию . (текущий каталог).

Ключи HTTP

-E

--html-extension

Данный ключ добавляет к имени локального файла расширение .html, если скачиваемый URL имеет тип application/xhtml+xml или text/html, а его окончание не соответствует регулярному выражению \.[Hh][Tt][Mm][Ll]?. Это полезно, например, при зеркалировании сайтов, использующих .asp страницы, когда вы хотите, чтобы зеркало работало на обычном сервере Apache. Также полезно при скачивании динамически-генерируемого содержимого. URL типа http://site.com/article.cgi?25 будет сохранён как article.cgi?25.html. Сохраняемые таким образом страницы будут скачиваться и перезаписываться при каждом последующем зеркалировании, т.к. Wget не может сопоставить локальный файл X.html удалённому адресу URL X (он ещё не знает, что URL возвращает ответ типа text/html или application/xhtml+xml). Для предотвращения перезакачивания используйте ключи -k и -K, так чтобы оригинальная версия сохранялась как X.orig.

--http-user=user

--http-passwd=password

Указывает имя пользователя user и пароль password для доступа к HTTP серверу. В зависимости от типа запроса Wget закодирует их, используя обычную (незащищённую) или дайджест схему авторизации. Другой способ указания имени пользователя и пароля - в самом URL. Любой из способов раскрывает ваш пароль каждому, кто запустит ps. Во избежание раскрытия паролей, храните их в файлах .wgetrc или .netrc и убедитесь в недоступности этих файлов для чтения другими пользователями с помощью chmod. Особо важные пароли не рекомендуется хранить даже в этих файлах. Вписывайте пароли в файлы, а затем удаляйте сразу после запуска Wget.

--no-cache

Отключает кеширование на стороне сервера. В этой ситуации Wget посылает удалённому серверу соответствующую директиву (Pragma: no-cache) для получения обновлённой, а не кешированной версии файла. Это особенно полезно для стирания устаревших документов на прокси серверах. Кеширование разрешено по умолчанию.

--no-cookies

Отключает использование cookies. Cookies являются механизмом поддержки состояния сервера. Сервер посылает клиенту cookie с помощью заголовка Set-Cookie, клиент включает эту cookie во все последующие запросы. Т.к. cookies позволяют владельцам серверов отслеживать посетителей и обмениваться этой информацией между сайтами, некоторые считают их нарушением конфиденциальности. По умолчанию cookies используются; однако сохранение cookies по умолчанию не производится.

--load-cookies file

Загрузка cookies из файла file до первого запроса HTTP. file - текстовый файл в формате, изначально использовавшемся для файла cookies.txt Netscape. Обычно эта опция требуется для зеркалирования сайтов, требующих авторизации для части или всего содержания. Авторизация обычно производится с выдачей сервером HTTP cookie после получения и проверки регистрационной информации. В дальнейшем cookie посылается обозревателем при просмотре этой части сайта и обеспечивает идентификацию. Зеркалирование такого сайта требует от Wget подачи таких же cookies, что и обозреватель. Это достигается через --load-cookies - просто укажите Wget расположение вашего cookies.txt, и он отправит идентичные обозревателю cookies. Разные обозреватели хранят файлы cookie в разных местах: Netscape 4.x. ~/.netscape/cookies.txt. Mozilla and Netscape 6.x. Файл cookie в Mozilla тоже называется cookies.txt, располагается где-то внутри ~/.mozilla в директории вашего профиля. Полный путь обычно выглядит как ~/.mozilla/default/some-weird-string/cookies.txt. Internet Explorer. Файл cookie для Wget может быть получен через меню File, Import and Export, Export Cookies. Протестировано на Internet Explorer 5; работа с более ранними версиями не гарантируется. Other browsers. Если вы используете другой обозреватель, --load-cookies будет работать только в том случае, если формат файла будет соответствовать формату Netscape, т.е. то, что ожидает Wget. Если вы не можете использовать --load-cookies, может быть другая альтернатива. Если обозреватель имеет "cookie manager", то вы можете просмотреть cookies, необходимые для зеркалирования. Запишите имя и значение cookie, и вручную укажите их Wget в обход "официальной" поддержки:

```
wget --cookies=off --header "Cookie: name=value"
```

--save-cookies file

Сохранение cookies в file перед выходом. Эта опция не сохраняет истекшие cookies и cookies без определённого времени истечения (так называемые "сессионные cookies"). См. также --keep-session-cookies.

--keep-session-cookies

При указании --save-cookies сохраняет сессионные cookies. Обычно сессионные cookies не сохраняются, т.к подразумевается, что они будут забыты после закрытия обозревателя. Их сохранение полезно для сайтов, требующих авторизации для доступа к страницам. При использовании этой опции разные процессы Wget для сайта будут выглядеть как один обозреватель. Т.к. обычно формат файла cookie file не содержит сессионных cookies, Wget отмечает их временной отметкой истечения 0. --load-cookies воспринимает их как сессионные cookies, но это может вызвать проблемы у других обозревателей Загруженные таким образом cookies интерпретируются как сессионные cookies, то есть для их сохранения с --save-cookies необходимо снова указывать --keep-session-cookies.

--ignore-length

К сожалению, некоторые серверы HTTP (CGI программы, если точнее) посылают некорректный заголовок Content-Length, что сводит Wget с ума, т.к. он думает, что документ был скачан не полностью. Этот синдром можно заметить, если Wget снова и снова пытается скачать один и тот же документ, каждый раз указывая обрыв связи на том же байте. С этим ключом Wget игнорирует заголовок Content-Length, как будто его никогда не было.

--header=additional-header

Укажите дополнительный заголовок additional-header для передачи HTTP серверу. Заголовки должны содержать ":" после одного или более непустых символов и недолжны содержать перевода строки. Вы можете указать несколько дополнительных заголовков, используя ключ --header многократно.

```
wget --header='Accept-Charset: iso-8859-2' --header='Accept-Language: hr' http://aaa.
```

Указание в качестве заголовка пустой строки очищает все ранее указанные пользовательские заголовки.

--proxy-user=user

--proxy-passwd=password

Указывает имя пользователя user и пароль password для авторизации на прокси сервере. Wget кодирует их, использую базовую схему авторизации. Здесь действуют те же соображения безопасности, что и для ключа --http-passwd.

--referer=url

Включает в запрос заголовок `Referer: url'. Полезен, если при выдаче документа сервер считает, что общается с интерактивным обозревателем, и проверяет, чтобы поле Referer содержало страницу, указывающую на запрашиваемый документ.

--save-headers

Сохраняет заголовки ответа HTTP в файл непосредственно перед содержанием, в качестве разделителя используется пустая строка.

-U agent-string

--user-agent=agent-string

Идентифицируется как обозреватель agent-string для сервера HTTP. HTTP протокол допускает идентификацию клиентов, используя поле заголовка User-Agent. Это позволяет различать программное обеспечение, обычно для статистики или отслеживания нарушений протокола. Wget обычно идентифицируется как Wget/version, где version - текущая версия Wget. Однако, некоторые сайты проводят политику адаптации вывода для обозревателя на основании поля User-Agent. В принципе это не плохая идея, но некоторые серверы отказывают в доступе клиентам кроме Mozilla и Microsoft Internet Explorer. Этот ключ позволяет изменить значение User-Agent, выдаваемое Wget. Использование этого ключа не рекомендуется, если вы не уверены в том, что вы делаете.

--post-data=string

--post-file=file

Использует метод POST для всех запросов HTTP и отправляет указанные данные в запросе. --post-data отправляет в качестве данных строку string, а --post-file - содержимое файла file. В остальном они работают одинаково. Пожалуйста, имейте в виду, что Wget должен изначально знать длину запроса POST. Аргументом ключа --post-file должен быть обычный файл; указание FIFO в виде /dev/stdin работать не будет. Не совсем понятно, как можно обойти это ограничение в HTTP/1.0. Хотя HTTP/1.1 вводит порционную передачу, для которой не требуется изначальное знание длины, клиент не может её использовать, если не уверен, что общается с HTTP/1.1 сервером. А он не может этого знать, пока не получит ответ, который, в свою очередь, приходит на полноценный запрос. Проблема яйца и курицы. Note: если Wget получает перенаправление в ответ на запрос POST, он не отправит данные POST на URL перенаправления. Часто URL адреса, обрабатывающие POST, выдают перенаправление на обычную страницу (хотя технически это запрещено), которая не хочет принимать POST. Пока не ясно, является ли такое поведение оптимальным; если это не будет работать, то будет изменено. Пример ниже демонстрирует, как авторизоваться на сервере, используя POST, и затем скачать желаемые страницы, доступные только для авторизованных пользователей:

```
wget --save-cookies cookies.txt --post-data 'user=foo&password=bar' http://htmlweb.ru
```

```
wget --load-cookies cookies.txt -p http://server.com/interesting/article.php
```

Конфигурирование WGET

Основные настроки, которые необходимо писать каждый раз, можно указать в конфигурационном файле программы. Для этого зайдите в рабочую директорию Wget, найдите там файл sample.wgetrc, переименуйте его в **.wgetrc** и редакторе пропишите необходимые конфигурационные параметры.

```
user-agent = "Mozilla/5.0"
tries = 5 количество попыток скачать
wait = 0 не делать паузы
continue = on нужно докачивать
dir_prefix = ~/Downloads/ куда складывать скачаное
use_proxy=on - использовать прокси
http_proxy - характеристики вашего прокси-сервера.
```

Как под Windows заставить WGET читать настройки из wgetrc файла:

- Задать переменную окружения WGETRC, указав в ней полный путь к файлу.
- Задать переменную HOME, в которой указать путь к домашней папке пользователя (c:\Documents and settings\jonh). Тогда wget будет искать файл "wgetrc" в этой папке.
- Кроме этого можно создать файл wget.ini в той же папке, где находится wget.exe, и задать там дополнительные параметры командной строки wget.

Полезную информацию по WGET можно почерпнуть здесь:

- http://mydebianblog.blogspot.com/2007/09/wget.html
- http://forum.ru-board.com/topic.cgi?forum=5&topic=10066
- PhantomJS - Используйте, если вам нужно скачать сайт, часть данных на котором загружается с помощью JavaScript