

Wget – A Noob's guide

By Tim | Published: November 2, 2010 @ Regravity.com

Wget is a great tool, and has been for years, it was designed to connect to and download files directly from a Web Server live on the Internet.

Since the mid 90's, when we were all on dial-up, Unix users have had the pleasure of using Wget in some form or another.

Fast-forward to 2010 and Wget is still here, albeit much upgraded over the last 14 years.

What is Wget?

Wget is a command line application for retrieving content from web servers.

It supports HTTP, HTTPS and FTP protocols.

Suffice to say, Wget is a method to download files from a network resource (read: THE INTERNET) from the command line, and it's mighty powerful.

Why use Wget?

Valid question, why would you want to use a command line application when there are so many other tools to download files?

One answer: Recursive Downloads

Wget's power lies in its ability to recursively download by traversing links in a HTML file or Web Directory.

Sure other graphical tools can also do this, but if you are looking for a method that can be scripted or incorporated into another program then Wget is for you.

So how do I use Wget?

Woah, nice enthusiasm kiddo but lets install the tool first!

Linux Users: Nothing to do here, most distros have this included by default.

Windows Users: [Download Here](#) - To install just drop the Wget.exe into your Windows System32 Directory (c:\windows\system32\)

Mac Users: This is a little trickier, check out this guide: [Mac Tricks and Tips](#)

Ok, its installed, now what?

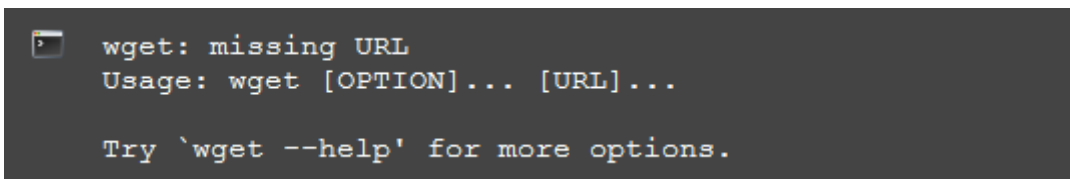
Great! You've installed Wget! Let's get down to business.

Fire up your Command Window / Console / Shell of choice and type in the following:



```
wget
```

You should have received something like:



```
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try 'wget --help' for more options.
```

If you did, congratulations, you've successfully installed Wget.

If you'd like to read the help file, type:



```
wget --help
```

Be prepared for a wall of text though, its a long help file.

Wget Command-Fu...

Lets get into some downloading, try this out:

```
wget http://www.google.com
```

You'll see an output like this:

```
--2010-11-01 00:16:28-- http://www.google.com/  
Resolving www.google.com... 66.102.11.104  
Connecting to www.google.com|66.102.11.104|:80...  
connected.  
HTTP request sent, awaiting response... 302 Found  
Location: http://www.google.com.au/ [following]  
--2010-11-01 00:16:28-- http://www.google.com.au/  
Resolving www.google.com.au... 66.102.11.104  
Reusing existing connection to www.google.com:80.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [text/html]  
Saving to: `index.html'  
  
[ < => ] 8,699 --.-K/s in 0s  
  
2010-11-01 00:16:29 (69.5 MB/s) - `index.html' saved  
[8699]
```

What you have just downloaded index.html from Google itself. Not a very useful file in the grand scheme of things but a nice test.

If you are wondering where the file is downloaded to, in this case it will be in a folder called **www.google.com** in the directory you originally run the command from.

This the simplest form of the Wget application, lets get a little more complex with the **--mirror** and **--recursive** switches.

Both of these switches, as most Wget switches, can be shortened to **-m** and **-r**. The use of these switches will both mirror the source directory and recursively dive into any directory that it finds.

```
wget -m -r http://some-website.com/files/
```

Ok so while that will do for starters, lets take a look at a few more useful switches. Specifically **-e robots=off** and **-nc** and **-np**.

```
wget -e robots=off -r -nc -np http://some-website.com/files/
```

The “robots” file on a web server is designed to keep automated search engine spiders and other directory structure tools from discovering directories and files. Essentially this hides tells a spider or script to ignore all files listed in the “robots” file. Wget also navigates directories in the same way a spider does, meaning you can’t download anything blocked by the robots file.

Thankfully, Wget has the capability to ignore this file using **-e robots=off**

The **-nc** or **--no-clobber** is to skip downloads that would download to existing files. Using this switch we have Wget look at already downloaded files and ignore them, making a second pass or retry to download possible without downloading files all over again.

The **-np** or **--no-parent** is to stop Wget from ascending into a parent directory. While this doesn’t generally happen, there are some cases where Wget will ascend into a parent directory and attempt to download more files than you have requested.

So now we have a fairly complex Wget command that will allow you to download files from a web server recursively, but what if you are looking to only download certain file-types or only download to a depth of 2 directories?

This is where we’d use the **-accept** and **-level=** switches

```
wget -e robots=off -r --level=0 -nc -np --accept jpg,gif,bmp http://some-website.com/files/
```

The command above using these new switches is much more targeted to both the types of files and the depth of directories.

--accept jpg,gif,bmp as you may have guessed is a filter for file-types. In the above example it will attempt to only download files with the *.jpg or *.gif or *.bmp file extension. Note that the list needs to be in a comma separated format.

Similarly you can use the **--reject** command to ignore specific file-types, handy for removing the pesky ‘index.htm’ and ‘.dstore’ files from your downloaded directories.

--level=0 dictates the depth of the directories you'd like to download, in this case its set to 0, meaning that there is no pre-determined depth to download (aka it will recursively download everything). You can also use *-level=inf* to achieve this same goal.

A higher number such as **--level=2** makes it stop at the desired depth, this example would dive into 2 directories below the parent to download along with the parent directory specified in the original command.

Where this becomes handy is if you have a content directory with a second level directory inside with supporting files you don't need (eg images, text files...etc...)

Scripting with Wget

Wget is powerful by itself, but what if you'd like to script a batch file to download with different variables? Yep you can do that to.

Lets take a look at a simple **FOR IN DO** loop that we can construct to do what we want.

Say you have a directory structure that has a parent directory and inside are the directories A-Z containing a bunch files, but you only want the *.txt files from each. Instead of having to run the Wget command in each of the directories we can use a **FOR IN DO** loop to grab the specific content.

On Windows it would look like:

```
FOR %g IN (A B C D E F G H I J K L M N O P Q R S T U  
V W X Y Z) DO wget -e robots=off -r --level=0 -nc -np  
--accept txt http://some-website.com/files/%g
```

On Linux and Mac it looks like:

```
FOR g IN A B C D E F G H I J K L M N O P Q R S T U V  
W X Y Z; DO wget -e robots=off -r --level=0 -nc -np -  
-accept txt http://some-website.com/files/$g; done
```

So to break it down, **FOR** the variable %g (or g) substitute **IN** A-Z and **DO** the **Wget** command with the correct address.

Now the **IN** part doesn't just have to be A-Z or 1-whatever number, you can put in here the specific names of each directory you want to download from. (ie **FOR %g IN** (Apple Banana Orange Pear Peach))

So What Now?

Go forth and prosper!

What I have shown here should get you off to a cracking start to using Wget for downloading from Web Servers.

There is a lot more to Wget than what I've explained above so I'd recommend reading the [MANUAL](#) if you want to get into the grittier side of using Wget.

*This article is published All Rights Reserved. 2010 by Regravity.com. It may not be reproduced or republished in part or in whole via digital content management system, digital transmission or in traditional print media without written consent from the article creator, please email info@regravity.com for further information about usage of this article.