



# WGET

Section: GNU Wget (1)

## NOM

wget - Un récupérateur réseau non interactif

## SYNOPSIS

wget [*option*]... [*URL*]...

## DESCRIPTION

GNU Wget est un programme non interactif de téléchargement de fichiers depuis le Web. C'est un logiciel libre. Il supporte les protocoles HTTP, HTTPS et FTP ainsi que le téléchargement au travers des proxies HTTP.

Wget est non interactif c'est-à-dire qu'il peut travailler en arrière-plan, sans intervention de l'utilisateur. Ceci vous permet de lancer un téléchargement et de vous déconnecter du système, laissant Wget finir le travail. En revanche, la plupart des navigateurs Web requièrent la présence constante de l'utilisateur, ce qui est particulièrement pénible lorsqu'on transfère beaucoup de données.

Wget peut suivre les liens des pages HTML et XHTML et créer une copie locale de sites web distants, en récréant complètement la structure du site original. Ceci est parfois désigné sous le nom de « téléchargement récursif ». En faisant cela, Wget respecte le standard d'exclusion de robots (*/robots.txt*). Wget peut aussi convertir les liens dans les fichiers HTML téléchargés pour la consultation locale.

Wget a été conçu pour être robuste en dépit des connexions réseaux lentes ou instables : si un téléchargement échoue suite à un problème réseau, il réessayera jusqu'à ce que l'intégralité du fichier soit téléchargée. Si le serveur supporte la reprise, il lui demandera de reprendre là où le téléchargement s'est interrompu.

# OPTIONS

## Options de base pour le démarrage

**-V**

**--version**

Affiche la version de Wget.

**-h**

**--help**

Affiche un message d'aide décrivant toutes les options de ligne de commande de Wget.

**-b**

**--background**

Passer en arrière-plan immédiatement après le démarrage. Si aucun fichier de sortie n'est spécifié via **-o**, la sortie est redirigée vers *wget-log*.

**-e** *commande*

**--execute** *commande*

Exécute *commande* comme si elle faisait partie de *.wgetrc*. La commande sera exécutée *après* celles de *.wgetrc*, et donc prendra le pas sur celles-ci.

## Options de suivi et de fichiers d'entrées

**-o** *logfile*

**--output-file=***logfile*

Enregistre tous les messages dans *logfile*. Normalement les messages sont affichés sur la sortie d'erreur standard.

**-a** *logfile*

**--append-output=***logfile*

Ajouter à *logfile*. C'est la même chose que l'option **-o**, sauf que les messages sont ajoutés à la fin du fichier *logfile* au lieu d'écraser l'ancien fichier. Si *logfile* n'existe pas, il sera créé.

**-d**

**--debug**

Active la sortie de débogage, c'est-à-dire des informations importantes pour les développeurs de Wget si quelque chose ne fonctionne pas correctement. Votre administrateur système peut avoir choisi de compiler Wget sans l'option de débogage, dans ce cas **-d** ne fonctionnera pas. Remarquez que la compilation avec cette option est toujours sûre : Wget compilé avec l'option de débogage **n'affichera aucune information de débogage** à moins qu'on ne le lui demande avec **-d**.

**-q**

**--quiet**

Désactive la sortie de Wget.

**-v**

**--verbose**

Active le mode verbeux, avec toutes les données disponibles. Ce mode est activé par défaut.

**-nv**

**--non-verbose**

Sortie non verbeux : désactive le mode verbeux, sans toutefois être complètement muet (utilisez **-q** pour ça), et donc seuls les messages d'erreurs et les informations de bases seront affichés.

**-i fichier**

**--input-file=fichier**

Lit les URL depuis *fichier*. Dans ce cas, aucune URL n'est requise sur la ligne de commande. S'il y a des URL sur la ligne de commande et dans un fichier d'entrée, celles de la ligne de commande sont téléchargées en premier. *fichier* n'est pas nécessairement un document HTML, une simple liste d'URL convient également.

Cependant, si vous spécifiez **--force-html**, le document sera considéré comme étant du **html**. Dans ce cas vous pouvez avoir des problèmes avec les liens relatifs, problème qui se résout en ajoutant `<base href="url">` aux documents ou en spécifiant **--base=url** sur la ligne de commande.

**-F**

**--force-html**

Lorsque l'entrée est lue depuis un fichier, force Wget à le

considérer comme un fichier HTML. Cela vous permet de rapatrier les liens relatifs depuis un fichier HTML de votre disque local, en ajoutant `<base href="url">` au HTML, ou en utilisant l'option de ligne de commande `--base`.

**-B** *URL*

`--base=URL`

Utilisé en conjonction avec `-F`, préfixe les liens relatifs avec *URL* dans le fichier spécifié par `-i`.

### Options de téléchargement

`--bind-address=ADRESSE`

Lors de la création de la connexion TCP/IP cliente, utilise *ADRESSE* sur la machine locale (`bind()`). *ADRESSE* peut être un nom d'hôte ou une adresse IP. Cette option peut être utile si votre machine possède plusieurs adresses IP.

`-t n`

`--tries=n`

Définit le nombre de tentatives à *n*. Spécifiez 0 ou `inf` pour un nombre illimité de tentatives. Par défaut 20 tentatives sont effectuées, à moins qu'une erreur fatale telle que « connexion refusée » ou « non trouvé » (404) apparaisse.

`-O fichier`

`--output-document=fichier`

Les documents ne seront pas écrits dans les fichiers appropriés, mais tous concaténés les uns aux autres et écrits dans *fichier*. Si *fichier* existe déjà, il sera remplacé. Si *fichier* vaut `-`, les documents seront écrits sur la sortie standard. Cette option positionne automatiquement le nombre de tentatives à 1.

`-nc`

`--no-clobber`

Si un fichier est téléchargé plus d'une fois dans un même répertoire, le comportement de Wget dépend de plusieurs options parmi lesquelles `-nc`. Dans certains cas, le fichier local sera remplacé au fur et à mesure des téléchargements. Dans les autres cas, il sera préservé.

Quand Wget est lancé sans **-N**, **-nc**, ni **-r**, le téléchargement d'un même fichier dans un même répertoire préservera le fichier original *fichier*, les copies suivantes seront renommées *fichier.1*, *fichier.2* et ainsi de suite. Si l'option **-nc** est spécifiée, ce comportement est supprimé, Wget ne téléchargera pas les nouvelles copies de *fichier*. C'est pourquoi **no-clobber** n'est pas un nom totalement adapté pour ce mode : ce n'est pas contre l'expulsion du fichier qu'elle protège (les suffixes numériques empêchent déjà le clobbering), mais elle empêche la sauvegarde des version multiples.

Quand Wget est lancé avec **-r**, mais sans **-N** ni **-nc**, le retéléchargement d'un fichier écrasera l'ancienne version de celui-ci. L'option **-nc** empêche ce comportement : la version originale est préservée et les nouvelles copies du serveur sont ignorées.

Quand Wget est lancé avec **-N**, indépendamment de **-r**, la décision de télécharger la nouvelle version du fichier dépend des dates du fichier local et distant et de la taille du fichier. **-nc** ne devrait pas être spécifié en même temps que **-N**.

Remarquez que si l'option **-nc** est spécifiée, les fichiers avec l'extension **.html** ou (beurk) **.htm** seront chargés depuis le disque local et analysés comme s'ils avaient été rapatriés depuis le Web.

**-c**

**--continue**

Reprend le téléchargement d'un fichier incomplet. C'est utile pour achever le téléchargement commencé par une instance précédente de Wget, ou par un autre programme. Par exemple :

```
wget -c ftp://sunsite.doc.ic.ac.uk/ls-lR.Z
```

S'il y a un fichier nommé *ls-lR.Z* dans le répertoire courant, Wget supposera que c'est la première partie du fichier distant, et demandera au serveur de continuer le téléchargement à l'offset égal à la longueur du fichier local.

Remarquez que vous n'avez pas besoin de spécifier cette option si vous voulez que l'invocation courante de Wget réessaie de retélécharger un fichier parce que la connexion a été perdue. C'est le comportement par défaut. `-c` n'affecte que la reprise des téléchargements démarrés *antérieurement* à cette invocation de Wget, et dont les fichiers locaux sont encore présents.

Sans l'option `-c`, l'exemple précédent aurait juste téléchargé le fichier distant dans *ls-IR.Z.1*, laissant intact le fichier tronqué *ls-IR.Z*.

Depuis Wget 1.7, si vous utilisez `-c` sur un fichier non vide, et que le serveur ne supporte pas la reprise du téléchargement, Wget refusera de télécharger le fichier depuis zéro, ce qui détruirait le contenu existant. Si vous voulez vraiment recommencer le téléchargement depuis zéro, effacez le fichier.

Également depuis Wget 1.7, si vous utilisez `-c` sur un fichier de taille supérieure ou égale à celle du serveur, Wget refusera de télécharger le fichier et affichera un message d'explication car dans ce cas la reprise du téléchargement n'a aucun sens.

En revanche, l'utilisation de `-c`, sur tout fichier qui est plus grand sur le serveur que localement sera considéré comme un téléchargement incomplet ; les octets manquants seront téléchargés et ajoutés la fin du fichier. C'est vraiment pratique dans certains cas : par exemple vous pouvez utiliser, `wget -c` pour ne télécharger que la nouvelle portion de données qui ont été ajoutés à un fichier de log.

Toutefois, si le fichier est plus grand du côté du serveur parce qu'il a été *modifié*, par opposition à un simple *ajout*, vous vous retrouvez avec un fichier corrompu. Wget n'a aucun moyen de vérifier que le fichier local est réellement un préfixe valide du fichier distant. Vous devez faire attention quand vous utilisez simultanément `-c` et `-r`, puisque chaque fichier sera considéré

comme un candidat à la reprise d'un téléchargement interrompu.

Autre exemple : vous récupérez un fichier corrompu si vous utilisez `-c` avec un proxy HTTP boiteux qui insère une chaîne « `transfer interrupted` » dans le fichier local. Dans le futur, une option retour en arrière pourrait être ajoutée pour s'occuper de ce cas.

Remarquez que `-c` ne fonctionne qu'avec les serveurs FTP et HTTP qui supportent l'en-tête *Range*.

#### `--progress=type=type`

Positionne le type d'indicateur de progression. Les indicateurs valides sont *dot* (le point) et *bar* (une barre).

Par défaut c'est la barre qui est utilisée. C'est une longue barre de progression ASCII pour matérialiser le téléchargement. Si la sortie n'est pas un terminal, les points seront utilisés. Si vous voulez forcer l'indicateur barre, utilisez `--progress=bar:force`.

`--progress=dot` permet de sélectionner l'affichage de points. Des points sont imprimés à l'écran pour matérialiser le téléchargement réussi d'une quantité déterminée de données.

Quand vous utilisez les points, vous pouvez aussi positionner le style en spécifiant `dot:style`. Les différents styles permettent de choisir la signification du point. Dans le style par défaut (*default*) chaque point représente 1K, il y a 10 points par bloc et 50 points par ligne. Le style *binary* est orienté ordinateur : 8 K par point, 16 points par bloc et 48 points par ligne (ce qui fait 384 K par ligne). Le style *mega* est plus adapté pour le téléchargement de très gros fichiers : chaque point représente 64 K il y a 8 points par bloc et 48 points sur chaque ligne (ce qui fait 3 M par ligne).

Remarquez qu'il est possible d'indiquer le style par défaut en utilisant la commande "progress" dans `.wgetrc`. Cette option a une priorité inférieure à celle de la ligne de commande.

**--timestamping**

Active l'estampille temporelle.

**-S****--server-response**

Affiche les en-têtes envoyés par les serveurs HTTP et les réponses envoyées par les serveurs FTP.

**--spider**

Lorsqu'il est invoqué avec cette option, Wget se comportera comme une *araignée*, sur la toile, il ne téléchargera pas les pages, il vérifiera simplement leur présence. Vous pouvez l'utiliser pour vérifier vos signets par exemple

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

Cette caractéristique nécessite encore quelques efforts pour se rapprocher des fonctionnalités de véritable araignées WWW.

**-T secondes****--timeout=secondes**

Positionne le décompte de lecture à *secondes* secondes. En cas d'émission d'une lecture réseau, le descripteur de fichier est vérifié pour un décompte, sinon une connexion pendante (lecture non interrompue) pourrait subsister. Le décompte par défaut est 900 secondes (quinze minutes). Positionnez le décompte à 0 pour désactiver la vérification des décomptes.

Veuillez ne pas diminuer la valeur par défaut du décompte sans savoir exactement ce que vous êtes en train de faire.

**--dns-timeout=secondes**

Positionne le décompte de recherche DNS à *secondes* secondes. Une recherche DNS qui dépasse le temps spécifié sera stoppée. Par défaut ce temps est dépendant de l'implantation des bibliothèques systèmes.

**--connect-timeout=secondes**

Positionne le décompte de connexion à *secondes* secondes. Les connexions TCP qui mettent plus de temps que spécifier pour s'établir sont stoppées. Par défaut, ce temps est dépendant de l'implémentation des bibliothèques système.



**--read-timeout=**

Positionne le décompte d'écriture (et de lecture) à *secondes* secondes. Les écritures prenant plus de temps que spécifié sont stoppées. La valeur par défaut est de 900 secondes.

**--limit-rate=valeur**

Limite la débit du téléchargement à *valeur* octets par seconde. Il est possible d'exprimer cette valeur en octets, kilooctets à l'aide du suffixe **k** ou en mégaoctets à l'aide du suffixe **m**. Par exemple, **--limit-rate=20k** limitera le débit à 20 ko/s. Cette option est utile, entre autre, lorsqu'on ne souhaite pas que Wget consomme toute la bande passante disponible.

Remarquez que Wget implante la limitation en dormant la durée appropriée après une réception depuis le réseau qui a pris moins de temps que spécifié par le débit. Au bout du compte, cette stratégie ralentit le transfert TCP approximativement au débit requis. Cependant, il se peut que cela prenne un peu de temps avant de se stabiliser ; ne soyez pas surpris si cette limite n'est pas vraiment respectée lors du transfert de petits fichiers.

**-w secondes****--wait=secondes**

Attendre la durée spécifiée en secondes entre les téléchargements. L'utilisation de cette option, est recommandée, puisqu'elle allège la charge du serveur en réduisant la fréquence des requêtes. À la place des secondes, la durée peut être spécifiée en minutes en utilisant le suffixe *m*, en heures avec le suffixe *h*, ou en jours avec le suffixe *d*.

La spécification d'une durée importante est utile si le réseau ou l'hôte cible est coupé, aussi Wget peut attendre suffisamment longtemps que le réseau soit réparé avant de réessayer.

**--waitretry=secondes**

Si vous ne voulez pas que Wget attende entre *chaque* téléchargement, mais uniquement entre les essais ou téléchargement infructueux, vous pouvez utiliser cette option. Wget attend *n* secondes au *n*-ième échec sur un fichier, jusqu'au maximum spécifié. C'est pourquoi une valeur de 10 fera patienter jusqu'à  $(1 + 2 + 3 + \dots + 10) = 55$  secondes par fichier.

Remarquez que cette option est activée par défaut dans le fichier global *.wgetrc*.

**--random-wait**

Certains sites web peuvent analyser les logs pour identifier les programmes de téléchargement comme Wget en recherchant les similarités statistiques temporelles entre les requêtes. Cette option fait varier la durée inter-requêtes entre 0 et  $2 * \textit{wait}$  secondes, où *wait* a été spécifié par l'option *-w* (ou *--wait*), dans le but de dissimuler la présence de Wget dans de telles analyses.

Un article récent dans une publication consacrée au développement sur une plate-forme cliente populaire fournissait le code pour réaliser cette analyse au vol. Son auteur suggérait de bloquer les adresses de classe C pour s'assurer que les programmes de téléchargement seront bloqués en dépit des changements d'adresses dues au DHCP.

L'option *--random-wait* a été inspirée par ce conseil mal avisé de bloquer l'accès à un site web à de nombreux utilisateurs innocents en raison des actions d'un seul.

**-Y on/off****--proxy=on/off**

Active/Désactive le support proxy. Le proxy est activé par défaut si la variable d'environnement appropriée est définie.

Pour plus d'information sur l'utilisation de proxys, voir la section Proxies de l'entrée GNU Info.

**-Q quota****--quota=quota**

Spécifie le quota de données pour les téléchargements automatiques. La valeur peut être spécifiée en octets (défaut), kilooctets (suffixe *k*), ou mégaoctets (avec le suffixe *m*).

Remarquez que le quota n'affectera jamais le téléchargement d'un unique fichier. Aussi, si vous spécifiez `wget -Q10k ftp://wuarchive.wustl.edu/ls-IR.gz`, l'intégralité du fichier *ls-IR.gz* sera téléchargée. La même chose se produira si vous spécifiez

plusieurs URL sur la ligne de commande. Wget ne tient compte du quota que pour les téléchargements récursifs ou depuis un fichier d'entrée. Aussi vous pouvez taper sans crainte **wget -Q2m -i sites** : le téléchargement s'arrêtera lorsque le quota sera atteint.

Positionner le quota à 0 ou **inf** signifie aucun quota.

**--dns-cache=off**

Désactive le cache lors de recherches DNS. Normalement, Wget garde en mémoire les adresses qu'il a déjà recherchées sur le serveur de noms, ce qui lui évite de recontacter à chaque fois le DNS pour le même (petit) ensemble d'adresses à partir desquelles la récupération est faite. Ce cache existe uniquement en mémoire vive, à chaque exécution Wget contacte de nouveau le DNS.

Toutefois, dans certains cas, il n'est pas désirable de garder en cache les noms d'hôte, même pour une durée courte telle que l'exécution de Wget. Par exemple, les serveurs HTTP qui sont hébergés sur des machines dont l'adresse IP est allouée dynamiquement et change fréquemment. Leur entrées DNS sont mises à jour à chaque changement. Quand le téléchargement depuis de tels hôtes est interrompu par un changement d'adresse IP, Wget réessaye de télécharger, mais (en raison du cache du DNS) il contacte l'ancienne adresse. Avec le cache DNS désactivé, Wget répétera la recherche DNS à chaque connexion, et ainsi obtiendra l'adresse dynamique correcte à chaque fois ; ceci au coût de recherches DNS supplémentaires et majoritairement inutiles.

Si vous ne comprenez pas la description ci-dessus, c'est que vous n'avez probablement pas besoin de cette option. (NDT : ou alors le traducteur s'est couché un peu trop tard !)

**--restrict-file-names=mode**

Modifie les caractères des URL distantes qui peuvent être utilisés dans les noms de fichiers locaux générés à partir de celles-ci. Les caractères qui sont *restreints* par cette option sont remplacés par %HH, où HH est le nombre hexadécimal qui correspond au caractère remplacé.

Par défaut Wget remplace les caractères considérés invalides

dans les noms de fichiers par le système d'exploitation, ainsi que les caractères de contrôle qui, pour la plupart, ne sont pas affichables. Cette option est utile pour modifier ce comportement par défaut, soit parce que vous souhaitez enregistrer les fichiers sur une partition non-native, soit parce que vous souhaitez garder les caractères de contrôle.

Quand le mode est mis à « unix », Wget remplace le caractère / ainsi que les caractères de contrôle faisant partie des ensembles 0-31 et 128-159. C'est le mode par défaut sur les systèmes d'exploitation à la UNIX.

Quand le mode est mis à « windows », Wget remplace les caractères \, |, /, :, ?, ", \*, <, > ainsi que les caractères de contrôle faisant partie des ensembles 0-31 et 128-159. En plus de cela, dans le mode windows, Wget utilise + à la place de : pour séparer l'hôte du port dans les noms de fichiers locaux et utilise @ à la place de ? pour séparer la partie de requête du reste dans le nom de fichier. Ainsi, une URL qui serait sauvegardée sous le nom [www.xemacs.org:4300/search.pl?input=blah](http://www.xemacs.org:4300/search.pl?input=blah) en mode unix aurait le nom [www.xemacs.org+4300/search.pl@input=blah](http://www.xemacs.org+4300/search.pl@input=blah) en mode windows. Ce mode est celui par défaut sous Windows.

Si vous ajoutez `,nocontrol` au mode, comme par exemple `unix,nocontrol`, le remplacement des caractères de contrôle est désactivé. Il est possible d'indiquer `--restrict-file-names=nocontrol` pour désactiver le remplacement des caractères de contrôle sans affecter le choix automatique du mode de restriction de caractère en fonction du système d'exploitation.

### Options de répertoires

**-nd**

**--no-directories**

Ne pas créer la hiérarchie de répertoires lors du téléchargement récursif. Si cette option est activée, tous les fichiers seront sauvegardés dans le répertoire courant, sans écrasement (si un

nom apparaît plus d'une fois, les noms de fichiers auront un suffixe numéroté .n).

**-x**

**--force-directories**

C'est l'opposé de **-nd** : créer une hiérarchie de répertoires, même si aucune n'aurait été créée autrement. Par exemple, **wget -x <http://fly.srk.fer.hr/robots.txt>** sauvegardera le fichier téléchargé dans le *fly.srk.fer.hr/robots.txt*.

**-nH**

**--no-host-directories**

Désactive la génération de la racine des répertoires avec le nom de l'hôte. Par défaut, l'invocation de Wget avec **-r <http://fly.srk.fer.hr/>** créera une hiérarchie de répertoires avec *fly.srk.fer.hr/* comme racine. Cette option désactive ce comportement.

**--cut-dirs=*n***

Ignore les *n* composantes du répertoire. C'est utile pour contrôler finement le répertoire lors du téléchargement récursif.

Par exemple, avec le répertoire <ftp://ftp.xemacs.org/pub/xemacs/>. Si vous le téléchargez avec l'option **-r**, il sera sauvegardé localement sous <ftp.xemacs.org/pub/xemacs/>. Alors que l'option **-nH** peut supprimer la partie <ftp.xemacs.org/>, vous êtes encore ennuyé avec *pub/xemacs*. C'est là que l'option **--cut-dirs** est intéressante ; elle permet de cacher à Wget *n* composantes du répertoire distant. Voici plusieurs exemples pour illustrer la fonctionnement de l'option **--cut-dirs** option.

```
pas d'option      -> ftp.xemacs.org/pub/xemacs/  
-nH               -> pub/xemacs/  
-nH --cut-dirs=1  -> xemacs/  
-nH --cut-dirs=2  -> .
```

```
--cut-dirs=1      -> ftp.xemacs.org/xemacs/  
...
```

Si vous voulez juste vous débarrasser de la structure des répertoires, cette option est similaire à la combinaison **-nd** et **-P**. Cependant, contrairement à **-nd**, **--cut-dirs** ne se perd pas dans les sous-répertoires. Par exemple, avec **-nH --cut-dirs=1**, le sous-répertoire *beta/* se retrouvera placé dans *xemacs/beta*, comme on s'y attend.

**-P** *prefixe*

**--directory-prefix=prefixe**

Positionne le préfixe pour les répertoires à *prefixe*. Le répertoire *prefixe* est le répertoire où tous les autres fichiers seront sauvegardés, c-a-d le sommet de l'arbre de téléchargement. Par défaut c'est **.** (le répertoire courant).

## Options HTTP

**-E**

**--html-extension**

Si un fichier de type **application/xhtml+xml** ou **texte/html** est téléchargé et que l'URL ne se termine par l'expression régulière **\.[Hh][Tt][Mm][Ll]?**, cette option provoque l'ajout du suffixe **.html** au nom de fichier local. C'est pratique, si vous êtes en train de répliquer un site distant qui utilise des pages **.asp**, mais vous voulez que les pages répliquées soient visibles sur votre serveur Apache. Cette option est aussi très utile lorsque vous téléchargez la sortie de CGI. Une URL comme **http://site.com/article.cgi?25** sera sauvegardé sous *article.cgi?25.html*.

Remarquez que les noms de fichiers ainsi modifiés seront retéléchargés chaque fois que vous ferez une réplique du site, car Wget ne peut pas savoir que le fichier local *X.html* correspond à l'URL *X* distante (puisque'il ne sait pas encore que l'URL produira une sortie de type **texte/html** ou **application/xhtml+xml**). Pour prévenir ce retéléchargement, vous devez utiliser **-k** et **-K** aussi la version originale de fichier sera sauvegardée sous le nom *X.orig*.

**--http-user=utilisateur**

**--http-passwd=mot-de-passe**

Spécifie le nom d'utilisateur *flutilisateur* et le mot de passe *flmot\_de\_passe* pour un serveur HTTP. En fonction du type de défi,

Wget le cryptera en utilisant le mécanisme d'authentification *basic* (non-sécurisé) ou *digest*.

Une autre manière de spécifier le nom d'utilisateur et le mot de passe est d'utiliser l'URL elle-même. Les deux méthodes révèlent votre mot de passe à quiconque se donnant la peine de lancer "ps". Pour éviter que le mot de passe soit vu, stockez-le dans *.wgetrc* ou *.netrc* et assurez-vous que vous avez protégé ces fichiers des autres utilisateurs à l'aide de la commande "chmod". Si les mots de passe sont vraiment importants, supprimez-les de ces fichiers dès que Wget a terminé le téléchargement.

Pour plus d'informations en ce qui concerne les problèmes de sécurité, reportez-vous à la section Security Considerations de l'entrée GNU Info.

**-C on/off**

**--cache=on/off**

Lorsque positionné à off, désactive le cache côté serveur. Dans ce cas, Wget enverra au serveur distant la directive appropriée (**Pragma:no-cache**) pour récupérer le fichier depuis le service distant, plutôt que la version cachée. C'est particulièrement utile pour télécharger et purger des documents obsolètes sur des serveurs proxies.

Le cache est autorisé par défaut.

**--cookies=on/off**

Lorsque positionné à off, désactive l'utilisation des cookies. Le cookie est un mécanisme pour préserver l'état du serveur. Le serveur envoie au client un cookie en envoyant l'en-tête Set-Cookie, et le client répondra avec le même cookie aux requêtes ultérieures. Comme les cookies permettent aux propriétaires du serveur de suivre à la trace les visiteurs et pour les sites d'échanger cette information, on peut les considérer comme une brèche dans la sphère privée. Par défaut, les cookies sont utilisés ; cependant le *stockage* des cookies n'est pas activé par défaut.

**--load-cookies fichier**

Charge les cookies depuis *fichier* avant le premier téléchargement HTTP. *fichier* est un fichier texte dans le format originellement

utilisé par Netscape pour le fichier *cookies.txt*

Vous utiliserez cette option lorsque vous répliquerez des sites qui requièrent d'être authentifié pour accéder à leur contenu. Le processus d'authentification repose typiquement sur l'envoi d'un cookie HTTP pour recevoir et vérifier votre accréditation. Le cookie est alors renvoyé par le navigateur lorsque vous accédez à certaines parties du site pour prouver votre identité.

Faire une copie d'un tel site requiert que Wget envoie les mêmes cookies que votre navigateur lorsqu'il communique avec le site. Ceci est fait en indiquant l'emplacement du fichier *cookies.txt* avec **--load-cookies**. Wget enverra les même cookies que votre navigateur dans la même situation. Les navigateurs stockent leurs cookies dans des endroits différents :

Netscape 4.x.

Les cookies sont dans *~/.netscape/cookies.txt*.

Mozilla and Netscape 6.x.

Les cookies de Mozilla s'appellent aussi *cookies.txt*, et sont situés quelque part dans le répertoire *~/.mozilla*. Le chemin complet ressemble souvent à *~/.mozilla/default/une-chaine-bizarre/cookies.txt*.

Internet Explorer.

Vous pouvez fabriquer un fichier de cookies pour Wget en utilisant dans le menu Fichier, Importer et Exporter, Exporter les cookies. Ceci a été testé avec Internet Explorer 5 ; ce n'est pas garanti que ça marche avec les versions antérieures.

Autres navigateurs.

Si vous utilisez un autre navigateur pour créer vos cookies **--load-cookies** ne fonctionnera que si vous pouvez indiquer l'emplacement ou produire un fichier cookie dans le format Netscape comme s'y attend Wget.

Si vous ne pouvez pas utiliser **--load-cookies**, il reste encore une alternative. Si votre navigateur supporte le gestionnaire de cookies, vous pouvez l'utiliser pour voir les cookies utilisés pour l'accès au site que vous copiez.



Recopier le nom et la valeur du cookie et dites manuellement à Wget d'envoyer ces cookies, en court-circuitant le support officiel des cookies :

```
wget --cookies=off --header "Cookie: <nom>=<valeur>"
```

**--save-cookies *fichier***

Sauvegarde les cookies dans *fichier* à la fin de la session. Les cookies sans date d'expiration, ou qui sont déjà périmés, ne sont pas sauvegardés.

**--ignore-length**

Malheureusement, certains serveurs HTTP (des programmes CGI, pour être plus précis) envoient des en-têtes avec un bogue dans « Content-Length », ce qui fait que Wget se comporte bizarrement et ne pense pas que tout le document a été téléchargé. Vous pouvez détecter ce syndrome si Wget essaye de récupérer le même document encore et encore, disant à chaque fois que la connexion a été fermée au même endroit dans le document.

Avec cette option, Wget ignorera l'en-tête « Content-Length », s'il est présent.

**--header=*en-tête-additionnel***

Définit un en-tête additionnel à passer aux serveurs HTTP. L'en-tête doit contenir un « : » précédé par un ou plusieurs caractères non blancs, et ne doit pas contenir de retour à la ligne.

Vous pouvez définir plus d'un en-tête additionnel en utilisant plusieurs fois l'option **--header**.

```
wget --header='Accept-Charset: iso-8859-2' \  
      --header='Accept-Language: hr' \  
      http://fly.srk.fer.hr/
```

La spécification d'une chaîne de caractères vide comme valeur d'en-tête effacera tous les en-têtes prédéfinis par l'utilisateur.

**--proxy-user=**

**--proxy-passwd=*mot-de-passe***

Spécifie le nom d'utilisateur *utilisateur* et le mot de passe *mot-de-passe* pour l'authentification sur un serveur proxy. Wget les encodera en utilisant le mécanisme d'authentification de base.

Les mêmes considérations sur la sécurité sont valables que pour l'option **--http-passwd**.

**--referer=*url***

Inclut « Referer: *url* » dans la requête HTTP. C'est utile pour récupérer des documents quand le traitement du côté serveur suppose qu'ils sont toujours récupérés par des navigateurs web interactifs et qu'ils ne sont envoyés correctement que lorsque « Referer » est positionné à l'une des pages qui pointent vers eux.

**-s**

**--save-headers**

Enregistre les en-têtes envoyés par le serveur HTTP vers le fichier, avant le contenu réel, avec une ligne vide comme séparateur.

**-U *agent-string***

**--user-agent=*agent-string***

S'identifier sous le nom *agent-string* pour le serveur HTTP.

Le protocole HTTP autorise les clients à s'identifier eux-mêmes en utilisant le champ « User-Agent » dans l'en-tête. Ceci permet de distinguer les logiciels WWW, le plus souvent pour des buts statistiques ou pour tracer des violations de protocole. Wget s'identifie normalement sous le nom **Wget/*version***, *version* étant la numéro de la version courante de Wget.

Cependant, certains sites sont connus pour imposer une politique de filtrage sur des clients en fonction de « User-Agent ». Si, conceptuellement, ce n'est pas une trop mauvaise idée, ça se traduit souvent par le refus de servir les clients autres que Mozilla ou Microsoft Internet Explorer. Cette option vous permet de falsifier la valeur de « User-Agent » envoyée par Wget. L'utilisation de cette option n'est pas recommandée, à moins que vous ne sachiez vraiment ce que vous êtes en train de faire.

**--post-data=*chaîne***

**--post-file=fichier**

Utilise POST comme méthode pour toutes les requêtes HTTP et envoie les données spécifiées dans le corps de la requête.

"--post-data" transmet *chaîne* comme donnée tandis que

"--post-file" transmet le contenu de *fichier*. Autrement, les deux commandes sont identiques.

Veillez noter que Wget a besoin de connaître la taille des données POST à l'avance. Par conséquent l'argument à "--post-file" doit être un fichier normal, une FIFO des entrées telles que */dev/stdin* ne fonctionnent pas. Il n'est pas vraiment certain comment cette limitation de HTTP/1.0 peut être contournée. Même si HTTP/1.1 introduit la notion de transferts en *blocs* qui ne requiert pas de connaître la taille à l'avance, un client ne peut utiliser cette fonction tant qu'il n'est pas sûr que le serveur supporte HTTP/1.1. Malheureusement, il n'est pas possible de le savoir avant que de recevoir la première réponse, ce qui requiert que la requête soit finie... le problème de l'oeuf et de la poule !

Remarquez que si Wget est redirigé après l'envoi de la requête POST, il ne transmettra pas les données POST à la nouvelle adresse. Ceci est dû au fait que souvent, les URL qui traitent le POST renvoient une redirection vers une page normale (même si, techniquement, ce n'est pas vraiment autorisé) qui ne désire pas ou ne supporte pas le POST. Il n'est pas encore clair si ce comportement est optimal ; si ça ne marche pas, ça changera.

L'exemple ci-dessous montre comment s'identifier sur un serveur en utilisant POST puis comment télécharger les pages souhaitées, qui ne sont accessible qu'aux utilisateurs autorisés :

```
# Identification sur le serveur. Cela peut être fait une seule fois.  
wget --save-cookies cookies.txt \  
      --post-data 'user=truc&password=bidule' \  
      http://server.com/auth.php
```

```
# Maintenant on peut récupérer tout ce qui nous intéresse.  
wget --load-cookies cookies.txt \  
-p http://server.com/interesting/article.php
```

## Options FTP

**-nr**

**--dont-remove-listing**

Ne pas supprimer les fichiers temporaires *.listing* générés par les requêtes FTP. Normalement, ces fichiers contiennent la liste brute des répertoires reçus depuis les serveurs FTP. Ne pas les supprimer peut être utile pour le débogage ou pour vérifier facilement le contenu des répertoires distants (par exemple pour vérifier que le miroir que vous êtes en train de consulter est complet).

Remarquez que, même si Wget écrit dans un nom de fichier connu, ce n'est pas un trou de sécurité dans le scénario où un utilisateur crée un lien symbolique *.listing* vers */etc/passwd* ou quelque chose d'autre et demande à root de lancer Wget dans son répertoire. En fonction de l'option utilisée, Wget refusera d'écrire dans *.listing*, faisant échouer l'opération d'écrasement, ou il supprimera le lien symbolique et le remplacera par le fichier *.listing*, ou le listing sera écrit dans un fichier *.listing.number*.

Même si cette situation n'est pas un problème, root ne devrait jamais lancer Wget dans le répertoire d'un utilisateur en qui il n'a pas confiance. Un utilisateur peut faire quelque chose d'aussi simple qu'un lien *index.html* vers */etc/passwd* et demander à root de lancer Wget avec **-N** ou **-r** aussi le fichier sera remplacé.

**-g on/off**

**--glob=on/off**

Active/désactive l'englobement FTP. L'englobement signifie que vous pouvez utiliser les caractères génériques comme *\**, *?*, [ et ] pour récupérer plusieurs fichiers d'un même répertoire en une fois. Par exemple :

```
wget ftp://gnjilux.srk.fer.hr/*.msg
```

Par défaut, l'englobement est activé si l'URL contient un caractère générique. Cette option permet d'activer ou de désactiver l'englobement de manière permanente.

Vous pouvez avoir à protéger l'URL avec des guillemets pour empêcher le shell d'interpréter les caractères génériques. L'englobement oblige Wget à regarder un listing de répertoires, qui est dépendant du système. C'est pourquoi, pour le moment ça ne marche qu'avec les serveurs FTP Unix (et ceux qui émulent la sortie du « ls » Unix).

#### **--passive-ftp**

Utilise le mécanisme de téléchargement FTP *passif*, dans lequel le client initialise la connexion de données. C'est parfois requis pour accéder à des FTP qui sont derrière des pare-feux.

#### **--retr-symlinks**

Habituellement, lors du téléchargement récursif de répertoires FTP, si on rencontre un lien symbolique, le fichier lié n'est pas téléchargé. À la place, on crée un lien symbolique identique sur le système de fichiers local. Le fichier pointé ne sera pas téléchargé, à moins bien sûr que le téléchargement récursif ne le rencontre plus tard.

Cependant, quand **--retr-symlinks** est spécifié, les liens symboliques sont suivis et les fichiers désignés sont récupérés. Pour le moment, cette option ne permet pas à Wget de suivre les liens symboliques sur les répertoire pour y poursuivre sa récursion, mais ceci devrait être amélioré dans le futur.

Remarquez que cette option n'a aucun effet pour le téléchargement d'un fichier (et non un répertoire) si celui-ci a été spécifié sur la ligne de commande plutôt que rencontré lors de la récursion. Dans ce cas les liens symboliques sont toujours suivis.

## Options de téléchargement récursif

**-r**

**--recursive**

Active le téléchargement récursif.

**-l** *profondeur*

**--level=profondeur**

Spécifie la profondeur maximale *profondeur* pour la récursion. Par défaut la profondeur maximale est 5.

**--delete-after**

Cette option indique à Wget de détruire tous les fichiers qu'il télécharge, *après* l'avoir fait. C'est utile pour précharger les pages les plus populaires dans un proxy, par exemple :

```
wget -r -nd --delete-after http://whatever.com/~popular/page/
```

L'option **-r** télécharge récursivement et **-nd** ne crée pas de répertoires.

Remarquez que **--delete-after** efface les fichiers sur la machine locale. Ça n'a rien à voir avec la commande **DELE** des serveurs FTP. Remarquez aussi que si **--delete-after** est spécifié, **--convert-links** est ignoré, et donc les fichiers **.orig** ne sont pas créés.

**-k**

**--convert-links**

Une fois que le téléchargement est terminé, convertit les liens du document pour qu'il soit consultable en local. Ceci affecte non seulement les liens hypertextes, mais aussi toute partie du document qui lie un contenu extérieur, comme les images, les liens vers les feuilles de style, les hyperliens vers du contenu non-HTML, etc.

Chaque lien sera modifié de l'une des deux façons suivantes :

Les liens vers les fichiers qui ont été téléchargés par wget seront transformés en liens relatifs.

Exemple : si le fichier téléchargé */toto/doc.html* a un lien vers */titi/image.gif*, aussi téléchargé, alors le lien dans *doc.html* sera modifié pour pointer vers *../titi/image.gif*. Ce type de transformation fonctionne très bien quelle que soit la combinaison de répertoires.

Les liens vers les fichiers qui n'ont pas été téléchargés par Wget seront modifiés pour inclure le nom d'hôte et le chemin absolu vers la destination.

Exemple : si le fichier téléchargé */toto/doc.html* a un lien vers */titi/image.gif* (ou vers *../titi/image.gif*), alors le lien dans *doc.html* sera modifié pour pointer vers <http://nom-d-hote/titi/image.gif>.

Grâce à ceci, la navigation locale fonctionne bien : si un fichier lié a été téléchargé, le lien désigne son nom local ; sinon, le lien désigne son nom Internet complet plutôt qu'un lien brisé. Le fait que les anciens liens soient convertis en liens relatifs vous permet de déplacer ailleurs les hiérarchies téléchargées.

Remarquez que Wget ne peut pas savoir avant la fin du téléchargement quels seront les liens téléchargés. C'est pourquoi le travail accompli par **-k** n'a lieu qu'après la fin de tous les téléchargements.

**-K**

**--backup-converted**

Lors de la conversion d'un fichier, sauvegarde le fichier original avec le suffixe **.orig**. Ceci affecte le comportement de **-N**.

**-m**

**--mirror**

Active toutes les options convenables pour faire un miroir. Cette option active la récursion, l'estampille temporelle, positionne la profondeur de récursion maximale à +infini et conserve le listing des répertoires FTP. Actuellement c'est équivalent à **-r -N -I inf -nr**.

-p

#### --page-requisites

Cette option oblige Wget à télécharger tous les fichiers nécessaires à l'affichage convenable d'une page HTML donnée. Notamment les images, sons et feuilles de styles référencées.

Ordinairement, lors du téléchargement d'une unique page HTML, les documents requis pour l'afficher proprement ne sont pas téléchargés. L'utilisation de `-r` et `-I` peut aider, mais puisque Wget ne fait pas d'ordinaire la distinction entre les documents externes et internes, on se retrouve parfois avec des « documents feuilles » qui n'ont pas leurs prérequis.

Par exemple, si le document *1.html* contient une balise "`<IMG>`" référençant *1.gif* et une balise "`<A>`" pointant vers un document externe *2.html*. Et si *2.html* est similaire mais que son image est *2.gif* et qu'il a un lien vers *3.html*. On peut aller loin comme ça.

Si on exécute la commande

```
wget -r -l 2 http://<site>/1.html
```

alors *1.html*, *1.gif*, *2.html*, *2.gif* et *3.html* seront téléchargés. Comme vous pouvez le constater, *3.html* n'aura pas sa dépendance *3.gif* parce Wget compte simplement le nombre de sauts (ici jusqu'à 2) depuis *1.html* dans le but de déterminer où il doit arrêter la récursion. Cependant, avec cette commande :

```
wget -r -l 2 -p http://<site>/1.html
```

tous les fichiers ci-dessus *et* la dépendance *3.gif* de *3.html* seront téléchargés. Similairement,



```
wget -r -l 1 -p http://<site>/1.html
```

provoquera le téléchargement de *1.html*, *1.gif*, *2.html*, et de *2.gif*.  
On pourrait croire que :

```
wget -r -l 0 -p http://<site>/1.html
```

ne téléchargera que *1.html* et *1.gif*, mais malheureusement ce n'est pas le cas car `-l 0` est équivalent à `-l inf`, la récursion infinie. Pour télécharger une unique page HTML (ou une poignée d'entre elles, toutes spécifiées sur la ligne de commande, ou dans un fichier d'entrée `-i URL`) et leurs dépendances, laissez simplement de côté `-r` et `-l` :

```
wget -p http://<site>/1.html
```

Remarquez que Wget se comportera comme si `-r` avait été spécifié, mais une seule page sera téléchargée, avec ses dépendances. Les liens de cette page vers des documents externes ne seront pas suivis. En réalité, pour télécharger une page unique avec ses dépendances, (même si celles-ci sont sur d'autres sites) et être sûr que tout s'affichera correctement en local, l'auteur utilise conjointement d'autres options :

```
wget -E -H -k -K -p http://<site>/<document>
```

Pour clore ce sujet, ça vaut la peine de connaître l'idée que se fait Wget d'un lien vers un document externe : c'est toute URL spécifiée dans une balise "`<A>`", "`<AREA>`", ou "`<LINK>`" autre que "`<LINK REL='stylesheet'>`".

#### **--strict-comments**

Active l'analyse syntaxique stricte des commentaires HTML. Par défaut la première occurrence de `-->` est considérée comme la fin

du commentaire.

D'après les spécifications, les commentaires HTML sont représentés comme les déclarations SGML. Une déclaration est une balise spéciale qui commence par `<!>` et se termine par `>`, tel que `<!DOCTYPE ...>`. Elles peuvent contenir des commentaires qui sont alors délimités du reste par une paire de `--`. Les commentaires HTML sont des « déclarations vides », des déclarations SGML qui ne contiennent que des commentaires. Par conséquent `<!--trucOmega->` est un commentaire valide, de même que `<!--unOmega- --deuxOmega->` mais pas `<!--1--2-->`.

D'un autre côté, la plupart des auteurs HTML ne perçoivent les commentaires que comme du texte délimité par `<!--` et `-->`, ce qui est légèrement différent. Par exemple, quelque chose comme `<!------->` n'est un commentaire valide que s'il y a un nombre de tirets multiple de quatre ! Sinon, techniquement, le commentaire ne s'arrête qu'au prochain `--`, qui peut être situé à l'autre bout du document. À cause de cela, la plupart des navigateurs Web ignorent complètement la spécification et implantent la définition plus intuitive qu'un commentaire est délimité par `<!--` et `-->`.

Jusqu'à la version 1.9 Wget interprétait les commentaires de manière stricte, ce qui provoquait parfois la disparition de certains liens dans des pages qui s'affichent correctement dans les navigateurs mais qui ont la malchance de contenir des commentaires non valides. Depuis la version 1.9, Wget a rejoint le cercle des clients qui implantent des commentaires « intuitifs », considérant la première occurrence de `-->` comme la fin d'un commentaire.

Si, pour une raison quelconque, vous souhaitez une analyse syntaxique stricte, utilisez cette option.

**-A *accliste* --accept *accliste***

**-R *rejliste* --reject *rejliste***

spécifie la liste (avec la virgule comme séparateur) des suffixes ou modèles de noms de fichiers qui doivent être acceptés ou rejetés.

**-D *liste-domaines***

**--domains=*liste-domaines***

Indique les noms de domaine qui peuvent être suivis.

*liste-domaines* est une liste avec la virgule comme séparateur.

Remarquez que ça *n'active pas -H*.

**--exclude-domains *liste-domaines***

Exclut les domaines contenus dans la liste *liste-domaines* des liens à suivre.

**--follow-ftp**

Suit les liens FTP depuis les documents HTML. Sans cette option, Wget ignorera tous les liens FTP.

**--follow-tags=*liste***

Wget a une table interne de balises HTML qu'il considère lorsqu'il examine les documents liés durant un téléchargement récursif. Si un utilisateur veut se limiter à un sous-ensemble de ces balises, celui-ci peut les spécifier à l'aide de cette option dans *liste*, avec la virgule comme séparateur.

**-G *liste***

**--ignore-tags=*liste***

C'est l'opposé de l'option **--follow-tags** option. Pour laisser de côté certaines balises HTML lors du téléchargement récursif, spécifier les dans une liste avec la virgule comme séparateur.

Dans le passé, l'option **-G** était la meilleure pour télécharger une page avec ses dépendances, en utilisant une ligne de commande comme :

```
wget -Ga,area -H -k -K -r http://<site>/<document>
```

Cependant, l'auteur de cette option a trouvé des pages avec des balises comme `<LINK REL="home" HREF="/">` et il réalisa que l'option **-G** n'était pas suffisante. On ne peut pas non plus dire à Wget d'ignorer `<LINK>`, sinon les feuilles de style ne seront pas

téléchargées. Maintenant, pour télécharger une page unique avec ses dépendances, il y a l'option dédiée **--page-requisites**.

**-H**

**--span-hosts**

Autorise le changement d'hôtes lors d'un téléchargement récursif.

**-L**

**--relative**

Ne suivre que les liens relatifs. C'est utile pour récupérer une page web sans être distrait, même par celles qui sont sur le même hôte.

**-I** *liste*

**--include-directories=***liste*

Spécifie une liste de répertoires séparés par des virgules que vous voulez suivre lors du téléchargement. Les éléments de *liste* peuvent contenir des caractères génériques.

**-X** *liste*

**--exclude-directories=***liste*

Spécifie une liste de répertoires séparés par des virgules que vous voulez exclure du téléchargement. Les éléments de *liste* peuvent contenir des caractères génériques.

**-np**

**--no-parent**

Ne pas remonter dans les répertoires parents. C'est une option utile, puisqu'elle garantit que seuls les fichiers *en dessous* d'une certaine hiérarchie seront téléchargés.

## EXEMPLES

Les exemples sont regroupés en trois sections en fonction de leur complexité.

### Utilisation simple

Si vous voulez juste télécharger une URL. Tapez juste :

wget <http://fly.srk.fer.hr/>

Mais que se passera-t-il si la connexion est lente et le fichier gros ? La connexion sera probablement interrompue avant que l'intégralité du fichier ne soit rapatriée. Dans ce cas, Wget essaiera de re-télécharger le fichier jusqu'à ce qu'il y arrive ou qu'il dépasse le nombre de tentatives par défaut (c'est-à-dire 20). Il est facile de changer le nombre de tentatives par 45, pour s'assurer que l'intégralité du fichier arrivera :

```
wget --tries=45 http://fly.srk.fer.hr/jpg/flyweb.jpg
```

Maintenant laissons Wget travailler en arrière-plan, et écrire sa progression dans le fichier de log. C'est fatigant de taper `--tries`, aussi nous utiliserons `-t`.

```
wget -t 45 -o log http://fly.srk.fer.hr/jpg/flyweb.jpg &
```

L'esperluette à la fin de la ligne envoie Wget à l'arrière plan. Pour avoir un nombre infini de tentatives, utilisez `-t inf`.

L'usage du FTP est aussi simple. Wget tiendra compte du login et du mot de passe.

```
wget ftp://gnjilux.srk.fer.hr/welcome.msg
```

Si vous spécifiez un répertoire, Wget téléchargera l'inventaire du répertoire et le transformera en document HTML. Essayez :

```
wget ftp://prep.ai.mit.edu/pub/gnu/  
links index.html
```

### Utilisation avancée

Vous avez un fichier qui contient les URL que vous voulez télécharger, utiliser l'option `-i` :

```
wget -i <file>
```

Si vous spécifiez « `-` » comme nom de fichier, les URL seront lues depuis l'entrée standard.

Créer une image miroir de profondeur 5 du site web de GNU, avec la même structure de répertoire que l'original, avec un seul essai par document, et en sauvegardant le log des activités dans *gnulog* :

```
wget -r http://www.gnu.org/ -o gnulog
```

Même chose que ci-dessus mais en convertissant les liens dans les fichiers HTML pour pointer vers des fichiers locaux, pour que vous puissiez les consulter hors-ligne :

```
wget --convert-links -r http://www.gnu.org/ -o gnulog
```

Télécharger une seule page HTML, mais en étant sûr que tous les éléments nécessaires à son affichage, tels que les images incluses et les feuilles de styles externes seront aussi téléchargées. Et s'assurer aussi que la page téléchargée référence les liens téléchargés.

```
wget -p --convert-links http://www.server.com/dir/page.html
```

La page HTML sera sauvegardé dans www.server.com/dir/page.html, et les images, feuilles de styles, etc, quelque part dans www.server.com/, en fonction de l'endroit où elles étaient sur le serveur distant.

Même chose que ci-dessus, mais sans créer le répertoire www.server.com/. En fait, je ne veux pas de tous ces noms de répertoires aléatoires du serveur, je veux juste sauvegarder *tous* les fichiers dans le sous-répertoire *download/* du répertoire courant.

```
wget -p --convert-links -nH -nd -Pdownload \  
http://www.server.com/dir/page.html
```

Télécharger le fichier index.html de www.linuxfr.org, en affichant les en-têtes originaux du serveur :

```
wget -S http://www.linuxfr.org/
```

Enregistrer les en-têtes du serveur, peut-être pour un traitement ultérieur.

```
wget -s http://www.linuxfr.org/  
more index.html
```

Télécharger les deux premiers niveaux de wuarchive.wustl.edu, et les mettre dans */tmp*.

```
wget -r -l2 -P/tmp ftp://wuarchive.wustl.edu/
```

Vous voulez télécharger tous les images GIF d'un répertoire depuis un serveur HTTP. Vous avez essayé `wget http://www.server.com/dir/*.gif`, mais ça ne fonctionne pas car le téléchargement HTTP ne gère pas l'expansion de nom. Dans ce cas, utilisez :

```
wget -r -l1 --no-parent -A.gif http://www.server.com/dir/
```

Explication : `-r -l1` signifie téléchargement récursif, jusqu'à la profondeur 1. `--no-parent` signifie que les références au répertoire parent sont ignorées et `-A.gif` signifie de ne télécharger que les fichiers GIF. `-A "*.gif"` marche aussi.

Supposons que vous étiez au milieu d'un téléchargement, et que Wget a été interrompu. Maintenant vous voulez remplacer les fichiers déjà présents Ce sera :

```
wget -nc -r http://www.gnu.org/
```

Si vous voulez encoder votre nom d'utilisateur et votre mot de passe pour HTTP ou FTP, utilisez la syntaxe URL appropriée :

```
wget ftp://mon_nom:mot_de_passe@unix.server.com/.emacs
```

Vous voulez rediriger la sortie des documents sur la sortie standard au lieu des fichiers :



```
wget -O - http://jagor.srce.hr/ http://www.srce.hr/
```

Vous pouvez aussi combiner les deux options et fabriquer des pipelines pour télécharger des documents référencé par un autre site.

```
wget -O - http://cool.list.com/ | wget --force-html -i -
```

### Utilisation très avancée

Si vous voulez que Wget conserve la réplique d'une page (ou de sous-répertoire FTP ), utilisez `--mirror (-m)`, qui est le raccourci pour `-r -l inf -N`. Vous pouvez mettre Wget dans le fichier crontab pour lui demander de vérifier le site chaque dimanche :

```
crontab
0 0 * * 0 wget --mirror http://www.gnu.org/ -o /home/me/weeklog
```

En plus de ci-dessus, vous désirez convertir les liens pour la consultation locale. Mais après avoir lu ce manuel, vous savez que la conversion de lien ne gère pas bien l'estampille temporelle, aussi vous désirez que Wget conserve une copie des fichiers HTML avant conversion. L'invocation de Wget ressemblera à :

```
wget --mirror --convert-links --backup-converted \  
http://www.gnu.org/ -o /home/me/weeklog
```

Mais vous avez aussi remarqué que la consultation locale ne

fonctionne pas du tout quand les fichiers HTML sont sauvegardés avec d'autres extensions que `.html`, peut-être parce qu'ils ont été envoyés sous le nom `index.cgi`. Aussi, vous voudriez que Wget renomme les fichiers avec content-type `text/html` et le nom `nom.html`.

```
wget --mirror --convert-links --backup-converted \  
--html-extension -o /home/me/weeklog \  
http://www.gnu.org/
```

Ou, en abrégé :

```
wget -m -k -K -E http://www.gnu.org/ -o /home/me/weeklog
```

## FICHIERS

`/etc/wgetrc`

Emplacement par défaut du fichier de démarrage *global*.

`.wgetrc`

Fichier de démarrage de l'utilisateur.

## BOGUES

Les rapports de bogues sont les bienvenus. Envoyez-les à [<bug-wget@gnu.org>](mailto:bug-wget@gnu.org).

Avant de soumettre un rapport de bogue, veuillez suivre les conseils simples suivants :

1.

Assurez-vous que le comportement que vous voyez est réellement un bogue. Si Wget crashe, c'est un bogue. Si Wget ne se comporte pas comme la documentation l'indique, c'est un bogue. Si les choses fonctionnent bizarrement, et que vous n'êtes pas sûr de la manière dont elles sont supposées fonctionner, ça peut aussi être un bogue.

2.

Essayer de mettre en évidence les circonstances qui provoquent le bogue. Par exemple si Wget crashe avec `wget -rLI0 -t5 -Y0 http://yoyodyne.com -o /tmp/log`, vous devriez essayer de voir s'il

crashe avec un ensemble plus réduit d'option.

Bien que je sois aussi intéressé par le contenu du fichier `.wgetrc`, il ne faut pas le recopier directement dans le message de debug. Il vaut mieux essayer de reproduire le bogue avec un `.wgetrc` le plus simple possible. Seuls les paramètres de `.wgetrc` qui ont une influence sur le bogue, devrait être envoyés par courrier électronique.

3.

Démarrez s'il vous plaît avec l'option `-d` et envoyez le log (ou uniquement les parties concernées). Si Wget a été compilé sans le support debug, recompilez-le. C'est *vraiment* plus facile de traquer les bogues avec le support du débogage activé.

4.

Si Wget a planté, essayez de le lancer dans un débogueur, par exemple `gdb `which wget` core` et tapez `where` pour obtenir la pile des appels.

## VOIR AUSSI

L'entrée GNU Info de `wget`.

## AUTEUR

Écrit à l'origine par Hrvoje Niksic <[hniksic@arsdigita.com](mailto:hniksic@arsdigita.com)>.

## COPYRIGHT

Copyright (c) 1996, 1997, 1998, 2000, 2001, 2002, 2003 Free Software Foundation, Inc.

Vous avez l'autorisation de créer et de distribuer des copies textuelles de ce manuel, à condition que la notice de copyright et la notice de permission soient préservées dans toutes les copies.

Vous avez l'autorisation de copier, distribuer et/ou modifier ce documents sous les conditions de la GNU Free Documentation License, Version 1.1 ou toute version ultérieure publiée par la Free Software Foundation; avec comme sections invariables « GNU General Public License » et « GNU Free Documentation License », sans texte avant ou après. Une copie de la licence est incluse dans la section intitulée « GNU Free Documentation License ».

## TRADUCTION

Sébastien Blanchet, 2002

Mise à jour par Eric Piel, 2004 <Eric.Piel AT tremplin-utc POINT net>

## **AVERTISSEMENT SUR LA TRADUCTION**

Il est possible que cette traduction soit imparfaite ou périmée. En cas de doute, veuillez vous reporter au document original en langue anglaise fourni avec le programme.

