

The REBOL Documentation Project

-- FR - Documentation REBOL - Manuels --

Manuels

Manuel de l'utilisateur - Chapitre 2 - Les Opérations

Philippe Le Goff

Première publication : 5 juin 2005, et mis en
ligne le dimanche 5 juin 2005

Résumé :

Ce document est la traduction française du Chapitre 2 du User Guide de REBOL/Core, qui concerne les Opérations.

Ce document est la traduction française du Chapitre 2 du User Guide de REBOL/Core, qui concerne les Opérations.

Traducteur : Philippe Le Goff

Historique de la Traduction

Date	Version	Commentaires	Auteur	Email
3 mai 2005 17:46	1.0.0	Traduction initiale	Philippe Le Goff	lp—legoff—free—fr

Installation de REBOL

L'installation de REBOL prend seulement quelques secondes et est très facile, non intrusive, et non perturbante !

Pour REBOL/Core, la seule procédure d'installation consiste à décompresser les fichiers de la distribution, et à les placer dans n'importe quel répertoire sur votre ordinateur.

Pour d'autres produits REBOL, l'installation peut nécessiter de fournir des informations complémentaires, comme l'endroit où mettre les fichiers.

Voir les notes spécifiques à la version choisie.

Les fichiers de la distribution

REBOL/Core comprend dans sa distribution de base les fichiers :

rebol (.exe) : *un programme exécutable qui démarre la console REBOL.*

rebol.r : *un fichier exécuté au lancement de REBOL (mais non nécessaire à son fonctionnement)*

setup.html : *des informations concernant l'installation et la mise en place.*

changes.html : *les modifications relatives aux versions récentes.*

license.txt : *la license REBOL.*

D'autres fichiers peuvent être fournis, selon le type de produit et la version de REBOL.

Variable d'environnement : HOME

Bien que cela ne soit pas nécessaire, si votre système d'exploitation utilise la variable d'environnement HOME, REBOL l'utilisera pour localiser ses fichiers de démarrage.

Pour beaucoup de systèmes d'exploitation comme UNIX, ou Linux, la variable HOME est positionnée par défaut (de sorte que vous n'avez pas à vous en préoccuper).

Paramétrage du Réseau

La première fois que vous utilisez REBOL, des informations vous sont demandées concernant le réseau. Ces informations sont optionnelles. Certains protocoles, comme le SMTP (email) ou le FTP, nécessitent une adresse email ou un nom de serveur SMTP. De plus, si vous êtes derrière un pare-feu (firewall) ou si vous utilisez un serveur de proxy, vous aurez à donner des informations spécifiques pour accéder à Internet.

Pour paramétrer le réseau :

- Saisir votre adresse email. Par exemple, nom@exemple.com.
- Saisir le nom de votre serveur SMTP. Par exemple : mail.exemple.com. Utilisez le nom usuel de votre serveur SMTP. Si vous n'êtes pas sûr du nom du serveur, contactez votre administrateur réseau ou votre fournisseur d'accès à Internet pour le nom de votre serveur de mail.
- Indiquez si vous utilisez un serveur de proxy. Si vous êtes directement connecté à Internet avec un modem ou Ethernet, saisissez N (non). Si vous passez par un proxy ou un pare-feu (firewall), fournissez les informations demandées dans les paramètres de Proxy et du Pare-feu, comme ci-dessous.

Une fois que vous aurez répondu aux questions posées, REBOL créera un fichier **user.r** et écrira dedans les paramètres Réseau. Vous pouvez changer ces paramètres à n'importe quel moment en éditant le fichier **user.r**.

Paramétrages du Proxy et d'un Pare-feu

Des serveurs de proxy ou des firewalls sont couramment utilisés par les entreprises ou les organisations pour sécuriser les accès depuis et vers Internet. Afin que REBOL puisse accéder à Internet via ces systèmes, vous devrez fournir quelques informations :

Quand REBOL demande si vous utilisez un serveur de proxy, répondez Y (yes).

Saisissez le nom de votre serveur de proxy (host). Il s'agit du serveur ou du pare-feu sur votre réseau, qui est utilisé pour cela. Saisissez le numéro du port utilisé par le serveur de proxy pour ses requêtes. Typiquement, c'est le port 1080, mais cela peut varier.

Si vous ne connaissez pas le numéro du port, regardez les paramètres de votre navigateur Web, ou demandez-le à votre administrateur réseau.

REBOL est configuré par défaut pour utiliser le protocole de proxy SOCKS. Vous pouvez indiquer un autre type en éditant le fichier **user.r** ou en fournissant à la fonction **set-net** les informations exactes sur le type de proxy utilisé. Les types suivants sont supportés :

```
socks    - utilise la dernière version de SOCKS (5)
socks5   - utilise un proxy socks5
socks4   - utilise un proxy socks4
generic  - le proxy générique CERN
none     - pas de proxy
```

Ces paramètres sont fournies en tant que sixième argument de la fonction **set-net**, qui est appelée pour le fichier **user.r**.

Pour plus d'information sur la modification des caractéristiques du proxy dans le fichier **user.r**, voir le chapitre sur les protocoles Réseau.

Contrat de license

Le contrat de license REBOL pour un utilisateur, que vous acceptez en téléchargeant ou en installant REBOL, peut être consulté à n'importe quel moment à partir de la console REBOL, en y saisissant le mot **license**.

Le démarrage de REBOL

REBOL fonctionne sur une large variété de plate-formes. REBOL se démarre comme d'autres applications de votre système. Selon les spécificités de votre système d'exploitation, REBOL peut être lancé selon l'une ou n'importe laquelle des possibilités suivantes : une icône, une ligne de commande , une autre application.

A partir d'une icône

REBOL peut être lancé en double-cliquant sur l'icône du programme REBOL, ou un fichier ayant l'extension **".r"** ou une icône de raccourci REBOL.

Si vous double-cliquez sur l'icône du programme, REBOL s'initialise, affiche la console et attend une saisie en ligne de commande.

Si vous voulez exécuter REBOL via un script, vous pouvez également procéder ainsi :

- glisser le script sur l'icône du programme
- ou associer le fichier avec l'exécutable REBOL
- ou créer un raccourci ou un alias avec les informations concernant l'exécutable le script.

Voir le manuel de votre système d'exploitation pour plus d'information.

A partir de la ligne de commande

Depuis la ligne de commande (le shell), allez dans le répertoire qui contient le fichier **rebol.exe** (ou **rebol** pour les systèmes autres que Windows), et saisissez **rebol** ou **./rebol**.

Sur certains systèmes d'exploitation, tels qu'UNIX, vous pouvez créer des alias de commandes qui sont susceptibles d'exécuter REBOL avec un jeu d'arguments et de fichiers. De plus, UNIX vous permet de créer des scripts shell qui incluent un chemin comme :

```
#!/path/to/rebol
```

au tout début du fichier. Quand vous tapez le nom d'un fichier à la ligne de commande, UNIX lancera

REBOL pour exécuter le script.

Depuis une autre application

Pour écrire et déboguer des scripts REBOL, il est possible de configurer votre éditeur de texte préféré pour qu'il exécute REBOL avec votre script courant passé en argument. Chaque éditeur de texte procède différemment.

Par exemple, dans l'éditeur CodeWright de Premia, vous pouvez utiliser les options de l'interpréteur du langage pour configurer l'usage de REBOL. De sorte que vous n'aurez plus qu'à presser une simple combinaison de touches pour que votre script en cours soit sauvegardé, puis évalué par l'interpréteur REBOL.

Problèmes de sécurité

Par défaut, le niveau de sécurité est configuré pour éviter que des scripts modifient des répertoires ou des fichiers.

Sécurité des ports

La fonction **secure** autorise une grande flexibilité dans la gestion et le contrôle des aspects de REBOL relatifs à la sécurité. La configuration courante de la sécurité est retournée comme résultat de la fonction **secure**.

Les paramètres de sécurité utilisent un dialecte REBOL, qui est, un langage à l'intérieur du langage.

Le dialecte normal consiste en un bloc de paires de valeurs. La première valeur de la paire spécifie ce qui doit être sécurisé :

file : *concerne la sécurité des fichiers*

net : *concerne la sécurité du réseau*

NdT :

A noter que suivant les versions de REBOL, sont aussi possibles les mots : **library** (pour l'accès aux bibliothèques dynamiques), ou **shell** (pour l'accès au shell du système d'exploitation).

Fournir un nom de fichier ou de répertoire vous permet de déterminer des niveaux de sécurité plus fins, pour ce fichier spécifique ou ce répertoire.

La seconde valeur de la paire caractérise le niveau de sécurité. Ce peut être soit un mot ou un bloc de mots pour un niveau de sécurité.

Les mots possibles sont :

allow : *un accès sans aucune restriction*

ask : demander la permission en cas de tentative d'accès.

throw : renvoie une erreur en cas de tentative d'accès.

quit : fermeture de la session REBOL en cours, en cas de tentative d'accès.

Par exemple, pour autoriser tous les accès réseau, mais fermer la session REBOL en cas de tentative d'accès aux fichiers :

```
secure [  
  net allow ; les accès réseaux sont autorisés  
  file quit ; cloture de la session REBOL si accès sur des fichiers  
]
```

Si un bloc est utilisé plutôt qu'un mot pour la sécurité, il doit contenir des paires `type de niveau de sécurité - type d'accès`.

Ceci vous permet de détailler finement la sécurité que vous désirez. Les types possibles d'accès sont :

read : contrôle l'accès en lecture

write : contrôle de l'accès en écriture, et les modifications (effacement, changement de nom)

all : contrôle complet

Les paires sont analysées dans l'ordre où elles apparaissent, de sorte que les paires définies à la fin du bloc peuvent avoir pour effet de modifier la sécurité de celles définies au début. Il est possible ainsi de définir un type d'accès sans explicitement définir tous les autres.

Par exemple :

```
secure [  
  net allow  
  file [  
    ask all  
    allow read  
  ]  
]
```

Les définitions précédentes du niveau de sécurité invite à demander l'autorisation de l'utilisateur pour toutes les opérations sur les fichiers, sauf pour la lecture, qui est autorisée. Cette façon de faire peut aussi être utilisée pour des fichiers ou des répertoires particuliers. Par exemple :

```
secure [  
  net allow  
  file quit  
  %source/ [ask read]  
]
```

Ici, l'utilisateur est alerté si une tentative d'accès est faite sur le répertoire *%source/*. Sinon, et par défaut, c'est la fermeture de la session REBOL (**quit**).

Il existe un cas particulier pour lequel la fonction **secure** prend un seul mot en argument, ce mot étant l'une des actions possibles de sécurité. Dans ce cas, le niveau de sécurité pour *la partie réseau et l'accès aux fichiers* est globalement affecté par cette action :

```
secure quit
```

La fonction **secure** accepte aussi l'argument **none**, ce qui définit un accès sans aucune restriction (identique aux possibilités offertes par **allow**).

```
secure none
```

Le niveau de sécurité est maintenant :

```
secure [  
  net allow  
  file [  
    ask all  
    allow read  
  ]  
]
```

Si aucune restriction de sécurité n'est spécifiée, que ce soit pour le réseau ou pour l'accès aux fichiers, le fonctionnement par défaut est : **ask**, c'est à dire interroger l'utilisateur pour savoir que faire.

Le paramétrage courant ne pourra pas être modifié si une erreur se produit durant l'analyse de l'argument, le bloc définissant la sécurité.

Précédents paramètres de sécurité

La fonction **secure** renvoie les paramètres précédents de sécurité, avant qu'un nouveau paramétrage ne soit défini. Il s'agit d'un bloc de paramètres pour le réseau et le système de fichiers, suivi éventuellement de ceux propres à un fichier ou un répertoire. Le mot **query** peut être utilisé pour obtenir les paramètres de sécurité sans les modifier

```
current-security: secure query
```

Vous pouvez modifier le niveau courant de sécurité en récupérant les paramètres courants, puis en les modifiant, et enfin en fournissant les nouvelles valeurs à la fonction **secure**.

Abaisser le niveau de sécurité déclenche une demande envers l'utilisateur, pour validation. Une exception cependant existe quand la session REBOL est en mode *quiet*, avec dans ce cas (baisse de la sécurité), la fermeture de la session.

Aucune demande de confirmation n'est faite à l'utilisateur dans le cas où le niveau de sécurité s'accroît. Notez que les requêtes concernant la sécurité incluent maintenant une option pour permettre un accès total pour le reste du traitement des scripts.

Quand REBOL est exécuté à partir de la ligne de commande (shell), l'argument **-s** est équivalent à :

```
secure allow
```

et l'argument **+s** est équivalent à :

```
secure quit
```

Vous pouvez utiliser aussi l'argument **—secure** avec l'un des types de niveaux de sécurité pour les accès réseau et fichiers :

```
rebol --secure throw
```

Arguments du programme

Il existe plusieurs arguments pouvant être spécifiés en ligne de commande, ou dans un script batch, ou dans les propriétés d'un raccourci.

Pour voir les arguments et les options possibles pour n'importe quelle version du langage REBOL, saisissez la commande **usage** à l'invite de la console, pour obtenir :

```
The command line usage is:
```

```
REBOL
```

```
All fields are optional. Supported options are:
```

```
--cgi (-c)          Check for CGI input
--do expr           Evaluate expression
--help (-?)        Display this usage information
--nowindow (-w)     Do not open a window
--noinstall (-i)    Do not install (View)
--quiet (-q)        Don't print banners
--reinstall (+i)    Force an install (View)
--script file       Explicitly specify script
--secure level      Set security level:
                    (allow ask throw quit)
--trace (-t)        Enable trace mode
--uninstall (-u)    Uninstall REBOL (View)
```

```
Other command line options:
```

```
+q                Force not quiet (View)
-s                No security
+s                Full security
-- args           Provide args without script
```

```
Examples:
```

```
REBOL script.r
```



```
REBOL script.r 10:30 test@domain.dom
REBOL script.r --do "verbose: true"
REBOL --cgi -s
REBOL --cgi --secure throw --script cgi.r "debug: true"
REBOL --secure none
```

Le format de la ligne de commande est :

```
REBOL options script arguments
```

avec :

options : une ou plusieurs options du programme. Voir "Spécifier des options" ci-dessous pour plus de détails.

script : le nom du fichier du script que vous voulez exécutez. Si le nom contient des espaces, il doit être entouré de guillemets.

arguments : les arguments passés au script sous forme de chaîne de caractère. Ces arguments peuvent être récupérés depuis le script.

Tous les arguments ci-dessus sont optionnels, et n'importe quelle combinaison est autorisée.

Icônes / raccourcis :

Pour certains systèmes d'exploitation, comme Windows, ou AmigaOS, vous pouvez utiliser des raccourcis avec des icônes, et définir des propriétés aux raccourcis qui incluent les options précédentes. Vous pouvez ainsi définir des raccourcis qui exécuteront directement vos scripts REBOL avec les bonnes options.

Fichier script

Typiquement, vous exécutez REBOL en fournissant le nom du fichier correspondant au script que vous voulez évaluez. Un seul nom de fichier est autorisé. Par exemple :

```
REBOL script.r
```

Si le nom du fichier contient des espaces, ce nom doit être entouré de guillemets :

```
REBOL "mon chemin avec des espaces.r"
```

Spécifier des options

Les options possibles sont identifiées avec un signe plus (+) ou moins (-), avant un caractère ou par un double tiret (—) avant un mot.

C'est l'usage standard pour définir des options à un programme pour la plupart des systèmes d'exploitation.

Voici plusieurs exemples d'utilisation.

Pour évaluer un script avec une option, comme l'option -s, qui permet d'exécuter le script sans aucune sécurité, saisissez :

```
REBOL -s script.r
```

Pour obtenir des informations sur l'usage de REBOL, tapez :

```
REBOL -?  
REBOL --help
```

Pour exécuter REBOL sans ouvrir une nouvelle fenêtre (ce qui se produit si vous avez besoin de rediriger la sortie vers un fichier ou un serveur), saisissez :

```
REBOL -w  
REBOL --nowindow
```

Pour prévenir l'affichage des informations de démarrage, ce qui peut être nécessaire en cas de redirection vers un fichier ou un serveur, tapez :

```
REBOL -q  
REBOL --quiet
```

Pour évaluer une expression REBOL depuis la ligne de commande :

```
REBOL --do "print 1 + 2"  
REBOL --do "verbose: true" script.r
```

Ceci vous permet d'évaluer un script distant :

```
REBOL --do "do http://www.rebol.com/speed.r"
```

Pour changer le niveau de sécurité de REBOL, tapez :

```
REBOL -s script.r  
REBOL --secure none script.r
```

Pour utiliser des scripts REBOL en mode CGI (voir la partie sur CGI - Common Gateway Interface, dans le chapitre consacré aux protocoles réseaux, pour plus d'information) :

```
REBOL -c cgi-script.r  
REBOL --cgi
```

Des options multiples sont aussi possibles. Les options mono-caractère doivent être spécifiées ensemble. Les options utilisant des mots doivent être séparées par des espaces :

```
REBOL -cs cgi-script.r  
REBOL --cgi --secure none cgi-script.r
```

L'exemple précédent évalue le script `cgi-script.r` en mode CGI, sans sécurité. La méthode compacte est préférable pour divers serveurs Web qui limitent le nombre d'arguments permis sur la ligne de commande. (comme le serveur Web Apache sur Linux).

Redirection vers un fichier

Sur la plupart des systèmes d'exploitation, il est possible de rediriger l'entrée et la sortie standard depuis et vers un fichier. L'exemple :

```
rebol -w script.r > output-file
```

redirige la sortie vers un fichier . Pareillement,

```
rebol -w script.r
```

redirige l'entrée depuis un fichier.

Lors de la redirection des entrées/sorties vers un fichier :

Utilisez l'option `-w` pour éviter d'ouvrir une console REBOL, qui interférerait avec la redirection standard des entrées-sorties .

Arguments des scripts

Tout ce qui est fourni sur la ligne de commande, après le nom d'un fichier script, est passé à ce script en tant qu'argument. Ceci permet d'écrire des scripts qui acceptent directement des arguments depuis la ligne de commande.

Par exemple, si vous démarrez REBOL avec la ligne :

```
REBOL script.r 10:30 test@domain.dom
```

10:30 et test@domain.com seront pris comme des arguments.

Il y a deux façons d'obtenir les arguments de la ligne de commande. La première méthode renvoie les arguments comme un **bloc** de valeurs REBOL :

```
probe system/options/args  
["10:30" "test@domain.dom"]
```

La seconde méthode renvoie les arguments sous la forme d'une **chaîne** :

```
probe system/script/args  
"10:30 test@domain.dom"
```

Note relative aux versions :

Les versions précédentes retournaient un bloc de valeurs à partir de **script/args** (tout comme **options/args**). Il est conseillé de vérifier que votre script manipule le bon type de données **args** selon ce qui est indiqué ci-dessus.

Fichiers de démarrage

Lorsque REBOL démarre, il tente de charger les fichiers **rebol.r** et **user.r** .

Ces fichiers sont optionnels, mais ils peuvent être utilisés pour définir le paramétrage réseau, des fonctions usuelles, et pour initialiser des données utilisées par vos scripts.

Le fichier **rebol.r** manipule des fonctions spéciales ou des extensions pour REBOL, fournies à part de la distribution standard. Il est suggéré de ne pas éditer ce fichier ; en effet, il est écrasé à chaque nouvelle version de REBOL.

Le fichier **user.r** permet de gérer des préférences utilisateur. Vous pouvez éditer ce fichier et y ajouter n'importe quelle définition ou donnée nécessaire à vos besoins.

Sur des systèmes multi-utilisateur, il peut y avoir un fichier **user.r** par utilisateur. Bien que ce fichier **user.r** ne fasse pas partie de la distribution standard, il est automatiquement généré s'il n'existe pas.

Lorsque REBOL démarre, l'interpréteur vérifie d'abord l'existence des fichiers **rebol.r** et **user.r** dans le répertoire "home" et, s'ils ne sont pas trouvés là, dans le répertoire courant.

Pour définir un répertoire HOME, vous devez définir une variable d'environnement au niveau du système, pour le contexte utilisateur ou script approprié. Notez que certains systèmes, comme UNIX ou Linux, font parfois déjà cela, de sorte que cette définition n'est pas nécessaire.

Sur Windows XP, pour définir la variable HOME, vous ajouterez, par exemple :

```
set HOME=C:\REBOL
```

à votre environnement, via les étapes suivantes :

- Choisir Panneau de configuration, dans le menu Démarrer, puis Paramètres,
- Double-cliquez sur l'icône Système puis dans l'onglet Avancé, cliquez sur le bouton "variables d'environnement"
- Cliquez sur le bouton Nouveau
- Saisissez HOME dans le champ "nom de la variable" et "C :\REBOL" (ou le chemin où vous avez placé l'exécutable REBOL) dans le champ "valeur de la variable".

Sur les systèmes UNIX, vous pouvez définir le chemin vers REBOL en ajoutant une ligne, comme celle qui suit à votre fichier .profile ou .bashrc par exemple :

```
set HOME=/usr/bin/rebol
```

Pour certaines versions de REBOL, le chemin est stocké dans un fichier .rebol situé dans votre répertoire personnel (home directory).

Quitter REBOL

A n'importe quel moment, pour quitter REBOL, vous pouvez sélectionner **"Quit"** dans le menu **"File"** de la console, ou bien en tapant **quit** ou **q** en ligne de commande.

Vous pouvez également quitter le programme depuis un script :

```
if now/time > 12:00 [quit]
```

La console REBOL peut aussi provoquer la fermeture de la session si une erreur se produit durant le démarrage.

Exit n'est pas **Quit** :

N'utilisez pas le mot **exit** pour sortir de REBOL. Ce mot est utilisé pour forcer la sortie de fonctions, et renverra une erreur s'il est utilisé en ligne de commande depuis la console.

Usage de la console

Lorsque vous démarrez REBOL/Core, la console s'ouvre pour la saisie et l'affichage. Si vous fournissez en argument au programme un script, celui-ci est évalué et vous voyez l'affichage de la sortie, pour ce script. Le prompt en input ressemble à :

```
>>
```

Si vous tapez une expression sur la ligne de commande, en validant par la touche "Entrée", elle est évaluée et le résultat est retourné avec l'indicateur suivant :

```
==
```

Par exemple :

```
>> 100 + 20
== 120
>> now - 7-Dec-1944
== 20341
```

Modification du prompt :

Les indicateurs du prompt et de réponse peuvent être modifiés. Voir l'Annexe sur Console pour plus

d'information.

La console s'active également lorsqu'une erreur est rencontrée dans un script ou que la fonction **halt** est rencontrée dans le script.

Saisie sur plusieurs lignes

Si vous commencez un bloc sur la ligne de commande et que vous ne le terminez pas, le bloc est "étendu" sur la ligne suivante. Ceci est indiqué par une ligne qui commence avec un crochet, suivie d'une indentation. La ligne sera indentée avec quatre espaces pour chaque bloc ouvert. Par exemple :

```
loop 10 [  
  [ print "example"  
  [ if odd? random 10 [  
    [ print "here"  
    ]  
  ]  
]
```

Ceci est aussi valable pour les chaînes de caractères s'étalant sur plusieurs lignes et comprises entre accolades.

```
Print {This is a long  
{ string that has more  
{ than one line.}
```

Les crochets et les accolades qui apparaissent au sein de chaînes entre quotes sont ignorés. Vous pouvez quitter une saisie en cours à n'importe quel moment en pressant la touche Echap (ESC).

Interruption d'un script

Un script peut être interrompu en pressant la touche Echap (ou ESC), qui retourne immédiatement à la ligne de commande. Pour certains types d'opérations système ou d'activité réseau, il peut y avoir un délai entre l'appui sur Echap et le retour du prompt.

Rappel d'historique

Chaque ligne saisie dans REBOL est mise en mémoire pour être rappelée ultérieurement si nécessaire. Les touches "Flèche haut" et "Flèche bas" sont utilisées pour faire défiler la liste des lignes précédemment tapées. Par exemple, presser la touche "Flèche haut" une fois rappelle la précédente ligne tapée à la console.

Les lignes de commande de l'historique peuvent être écrites dans un fichier, en sauvant le bloc d'historique. Voir l'Annexe sur Console pour plus d'information.

Compléter automatiquement un mot

Pour améliorer la saisie des mots et noms de fichiers longs, il est possible avec la console REBOL que ces mots soient complétés automatiquement. Après la saisie de quelques lettres du mot, tapez sur la touche "Tabulation". Si les lettres saisies permettent d'identifier de manière unique le mot, le

restant de ce mot est affiché.

Par exemple, si vous saisissez :

```
>> sq
```

puis que vous appuyez sur la touche "Tabulation",

```
>> square-root
```

Si les lettres saisies ne permettent pas d'identifier sans ambiguïté le mot, vous pouvez appuyer encore sur la touche "Tabulation" pour obtenir la liste des choix possibles. Par exemple, saisissez :

```
>> so
```

Puis pressez deux fois de suite sur la touche "Tabulation" pour avoir :

```
>> sort source
```

```
so
```

et vous pourrez saisir le reste du mot, ou suffisamment de lettres pour le caractériser de manière unique. L'aide à la saisie fonctionne pour tous les mots, notamment les mots définis par l'utilisateur.

Elle fonctionne également avec les noms de fichiers, quand ceux-ci sont précédés du signe "%" (pourcentage).

```
>> print read %r
```

L'appui sur la touche "Tabulation" produira l'effet suivant :

```
>> print read %rebol.r
```

selon le contenu de votre répertoire courant.

Indicateur d'activité

Lorsque REBOL attend la fin d'une opération réseau, un indicateur d'activité apparaît pour indiquer que quelque chose est en cours. Vous pouvez changer cet indicateur avec votre propre motif (mode caractère). Voir l'Annexe sur Console pour plus d'information.

Connections Réseau

Lorsqu'une connection réseau s'effectue, un message apparaît dans la console. Par exemple, si vous saisissez :

```
>> read http://www.rebol.com  
connecting to: www.rebol.com
```

Si besoin, vous pouvez désactiver cet affichage en utilisant l'option **quiet**. Voir l'Annexe sur Console

pour plus d'information.

Terminal virtuel

La console fournit des possibilités propres à un terminal virtuel, comme le mouvement du curseur, son adressage, l'édition de ligne, l'effacement de l'écran, des raccourcis clavier, et la gestion de la position du curseur. Le terminal virtuel utilise le code de caractères ANSI. Ceci vous permet d'écrire des programmes pour terminaux indépendants des systèmes d'exploitation, comme des éditeurs de texte, des clients email, ou des émulateurs telnet. Plus d'information dans l'Annexe de la Console.

Obtenir de l'aide

Il existe plusieurs sources d'information : l'aide en ligne de REBOL, la fonction **source**, la documentation sur le site Web de REBOL, la bibliothèque de scripts REBOL (library), la mailing liste, et les envois de feedback ou bugs (Rambo, par exemple).

L'aide en ligne

La fonction **help** d'aide en ligne constitue un moyen rapide d'obtenir une information synthétique sur les mots REBOL. Il y a plusieurs manières d'utiliser **help**.

Saisissez **help** ou juste **?** en ligne de commande pour voir un résumé de cette fonction :

```
The help function provides a simple way to get
information about words and values.  To use it
supply a word or value as its argument:
```

```
help insert
help find
```

```
To view all words that match a pattern:
```

```
help "path"
help to-
```

```
To view all words of a specified datatype:
```

```
help native!
help datatype!
```

```
There is also word completion from the command
line.  Type a few chars and press TAB to complete
the word.  If nothing happens, there is more than
one word that matches.  Enough chars are needed
to uniquely identify the word.
```

```
Other useful functions:
```



```
about - for general info
usage - for the command line arguments
license - for the terms of user license
source func - print source for given function
upgrade - updates your copy of REBOL
```

For more information, see the user guides.

Si vous donnez à **help** un mot comme argument, **help** affichera toute l'information relative à ce mot. Par exemple, si vous tapez :

```
help insert
```

vous verrez :

```
USAGE:
    INSERT series value /part range /only /dup count

DESCRIPTION:
    Inserts a value into a series and returns the series after the insert.
    INSERT is an action value.

ARGUMENTS:
    series -- Series at point to insert (Type: series port bitset)
    value -- The value to insert (Type: any-type)

REFINEMENTS:
    /part -- Limits to a given length or position.
    range -- (Type: number series port)
    /only -- Inserts a series as a series.
    /dup -- Duplicates the insert a specified number of times.
    count -- (Type: number)
```

La fonction **help** permet aussi de trouver des mots qui contiennent une chaîne de caractères spécifiques. Par exemple, pour trouvez tous les mots qui incluent la chaîne *path*, saisissez dans la console :

```
? "path"
```

et le résultat devrait être :

```
Found these words:
clean-path      (function)
lit-path!       (datatype)
lit-path?       (action)
path            (file)
path!           (datatype)
path-thru       (function)
path?           (action)
set-path!       (datatype)
```

set-path?	(action)
split-path	(function)
to-lit-path	(function)
to-path	(function)
to-set-path	(function)

Vous pouvez également effectuer des recherches par datatypes, par types de données. Par exemple, pour voir la liste de tous les mots qui sont du type **function !**, vous pouvez taper la commande :

```
? function!
```

et le résultat s'affichera :

```
Found these words:
?           (function)
??          (function)
about       (function)
alert       (function)
alter       (function)
append      (function)
array       (function)
ask         (function)
...
```

Pour obtenir une liste de tous les types de données REBOL :

```
? datatype!
```

```
Found these words:
action!     (datatype)
any-block!  (datatype)
any-function! (datatype)
any-string! (datatype)
any-type!   (datatype)
any-word!   (datatype)
binary!     (datatype)
bitset!     (datatype)
block!      (datatype)
char!       (datatype)
datatype!   (datatype)
date!       (datatype)
decimal!    (datatype)
email!      (datatype)
...
```

En ce qui concerne les Objets :

La fonction `help` ne fournit pas d'aide utile concernant les objets REBOL, par exemple :

```
help system/options/home
system/options/home is a path.
```

N'essayez pas de faire :

```
help system
```

car cette opération prend plusieurs minutes, et produit plus d'un mégaoctet de sortie texte à l'écran.

Consultation du code source

Les utilisateurs expérimentés peuvent apprendre beaucoup sur certaines fonctions REBOL spécifiques, en consultant leur code source. La fonction **source** affiche le code pour n'importe quelle fonction REBOL définie. Si vous tapez ceci :

```
source join
```

alors le code **source** de la fonction **join** sera affiché :

```
join: func [
  "Concatenates values."
  value "Base value"
  rest "Value or block of values"
][
  value: either series? value [copy value] [form value]
  repend value rest
]
```

Les fonctions pré-définies REBOL comprennent les fonctions mezzanine (fonctions internes implémentées en REBOL), et des fonctions définies par l'utilisateur. Les fonctions natives sont des fonctions internes implémentées en code machine, et leurs codes source ne peuvent être affichées.

Téléchargement de documentation

Vous disposez sur le [site Web de REBOL](http://www.rebol.com/) (<http://www.rebol.com/>) d'un ensemble de documentations actualisées.

En supplément de ce manuel, il existe un Dictionnaire REBOL qui couvre tous les mots pré-définis existants en REBOL. Si l'aide en ligne ou ce guide ne donnent pas assez d'information sur un mot REBOL, regardez dans le dictionnaire pour une description détaillée. Le dictionnaire est mis à jour à chaque version de REBOL et se trouve disponible sur le site Web :

<http://www.REBOL.com/docs/dictionary.html>

Bibliothèque de scripts

Le site Web REBOL contient une bibliothèque de scripts avec de nombreux exemples couvrant une large variété de sujets. La bibliothèque est divisée en catégories pour rendre plus facile la recherche

d'un script spécifique à une fonction donnée. Vous pouvez également chercher dans la bibliothèque des scripts qui contiennent un mot particulier. La bibliothèque de scripts peut être trouvée sur le site : <http://www.REBOL.com/library/library.html>

La Mailing liste

Il est possible d'obtenir aussi de l'aide de la communauté des utilisateurs REBOL en rejoignant la liste de discussion. Pour vous inscrire, envoyer un email à rebol-request@rebol.com avec comme sujet juste le mot "**subscribe**". Par exemple :

```
send rebol-request@rebol.com "subscribe"
```

Vérifiez au préalable que votre adresse email est bien correctement paramétrée avec la fonction **set-net**.

Nous contacter

Nous souhaitons savoir ce que vous pensez ; merci de nous contacter pour :

- Les rapports concernant les problèmes, bugs, crashes.
- Nous dire comment vous utilisez REBOL
- Faire des suggestions
- Obtenir plus d'informations concernant nos produits

Vous pouvez nous contacter via la page de [Feedback](#) sur notre site Web.

Erreurs

Messages d'erreur

Il existe plusieurs types de messages d'erreur au sein de REBOL. Lorsqu'une erreur se produit, un message est affiché qui explique de quelle erreur il s'agit, et approximativement, où elle s'est produite. Par exemple, si vous saisissez :

```
abc
** Script Error: abc has no value.
** Where: abc
```

Le type d'erreur est indiqué par les premiers mots du message. Dans l'exemple ci-dessus, l'erreur est du type "Script Error". Les erreurs de script sont les plus communes et se produisent quand vous utilisez improprement une fonction du langage ou avec des arguments incorrects. D'autres types d'erreurs sont décrits dans le tableau ci-dessous sur **Types d'Erreur**.

Type d'erreur	Description
Syntax errors (Erreur de syntaxe)	Se produit lorsqu'un script contient une valeur invalide, ou un en-tête manquant, ou des crochets, accolades, parenthèses, ou guillemets non appariés.
Math errors (Erreur Mathématique)	Survient quand un nombre est divisé par zéro ou qu'un dépassement de capacité se produit.
Access errors (Erreur d'accès)	Se produit lorsqu'un accès à un fichier, un répertoire ou une opération ne peuvent être réalisés, ou que des restrictions d'accès existent.
Throw errors	Survient quand un break, exit, ou un throw est utilisé de façon incorrecte.
User errors (Erreurs Utilisateur)	Sont définies par l'utilisateur dans un script.

La plupart des types d'erreur peuvent être capturées et traitées dans votre script. Voir le chapitre sur les Expressions, Section "Test de Bloc" pour une description de la fonction `try`.

L'Annexe consacrée aux Erreurs inclut aussi des informations utiles concernant les erreurs.

Redirection des erreurs

Quand des erreurs sont rencontrées dans des sessions non interactives, comme celles en mode CGI (-c ou —cgi), ou en mode sans fenêtre (-w ou —nowindow), la session se termine automatiquement.

Si un script se termine durant son exécution en mode non interactif, il vous est possible de rediriger les erreurs rencontrées vers un fichier.

Quand des erreurs sont rencontrées en mode non interactif, comme avec :

```
REBOL -cs my_script.r >> my_script.log
```

Comme pour la plupart des systèmes d'exploitation, ici, la sortie standard est redirigée vers le fichier `my_script.log`.

Mise à jour

Au démarrage, une bannière est affichée qui identifie la version du programme. Les numéros de version ont le format :

```
version.revision.update.platform.variation
```

Par exemple, le numéro de version :

```
2.3.0.3.1
```

indique que vous avez une version 2, revision 3, update 0, pour Windows 95/98/NT (plate-forme REBOL numéro 3.1). Une liste de tous les identifiants des plate-formes est disponible sur <http://www.rebol.com>.

Vous pouvez également obtenir le numéro de votre version en tapant en ligne de commande :

```
print system/version
```

Seule la dernière version est supportée par REBOL Technologies. Vous pouvez vérifier que vous avez la dernière version et automatiquement effectuer une mise à jour si nécessaire. Pour cela, vérifiez que vous pouvez vous connecter à Internet, et depuis la console, saisissez :

```
upgrade
```

REBOL vous retournera l'un ou l'autre des messages suivants (par exemple) concernant votre version :

```
This copy of Windows 95/98/NT ix86 REBOL/core 2.3.0.3.1  
is currently up to date.
```

ou :

```
This copy of Windows 95/98/NT ix86 REBOL/core 2.1.2.3.1  
is not up to date. Current version is: 2.3.0.3.1.  
Download current version?
```

```
To upgrade to the latest version, type Y (yes). Otherwise, type  
N (no).
```

Updated 7-Apr-2005 - Copyright REBOL Technologies - Formatted with MakeDoc2
Translated by Philippe Le Goff