The REBOL Documentation Project

-- FR - Documentation REBOL - Manuels --

Manuels

Manuel de l'utilisateur - Annexe 3 - La console

Philippe Le Goff

Première publication : 8 novembre 2005, et
mis en ligne le mardi 8 novembre 2005

Résumé :

Ce document est la traduction française de l'Annexe 3 du User Guide de REBOL/Core, qui concerne la console.

Ce document est la traduction française de l'Annexe 3 du User Guide de REBOL/Core, qui concerne la console.

1 Historique de la traduction

Date	Version	Commentaires	Auteur	Email
16 Septembre 2005 07:03	1.0.0	Traduction initiale	Philippe Le Goff	lp—legoff—free—fr

2 Le prompt de la ligne de commande

NdT : On appelle "prompt" l'indicateur alphanumérique qui indique que la console attend une saisie. En console Dos, ou Unix, cette notion est bien connue. REBOL possède aussi un prompt en mode console.

Le prompt par défaut de la ligne de commande est ">>". Il est possible de modifier cela avec par exemple le code suivant :

```
system/console/prompt: "Input: "
```

Le prompt devient alors :

```
Input:
```

Le prompt peut être un bloc qui sera évalué à chaque fois. La ligne suivante affiche l'heure courante :

```
system/console/prompt: [reform [now/time " >> "]]
```

Le résultat devrait ressembler à un prompt comme ceci :

```
10:30 >>
```

3 Indicateur de résultat

Par défaut, l'indicateur de résultat est "==", il peut être aussi modifié avec une ligne telle que :

```
system/console/result: "Result: "
```

Ces exemples modifiant les paramètres par défaut peuvent être placés dans votre fichier **user.r** pour les rendre permanent, à chaque démarrage de REBOL.

4 Rappel de l'historique

Chaque ligne saisie dans la console REBOL est conservée dans un bloc, en historique, et peut être rappelée plus tard en utilisant les touches "flèche haut" et "flèche bas" (up et down).

Par exemple, si vous appuyez une seule fois sur la touche "flèche haut", vous rappellerez la dernière ligne de commande saisie.

Le bloc d'historique contenant toutes les lignes entrées à la console peut être consulté en appelant l'attribut **history** de l'objet **system/console**.

```
probe system/console/history
```

Vous pouvez sauvegarder ce bloc dans un fichier :

```
save %history.r system/console/history
```

et il peut être rechargé plus tard :

```
system/console/history: load %history.r
```

Ces lignes peuvent être placées dans le fichier user.r pour sauvegarder et recharger votre historique entre deux sessions.

5 Indicateur d'activité

Lorsque REBOL attend qu'une opération (par exemple, une requête réseau) se termine, un indicateur d'activité apparaît à l'écran pour indiquer que quelque chose est en cours.

Vous pouvez changer l'indicateur d'activité avec un ligne comme celle-ci par exemple :

```
system/console/busy: "123456789-"
```

Quand REBOL travaille en mode "silencieux" (quiet mode), l'indicateur d'activité n'est pas affiché.

6 Opérations spécifiques avec la console

La console se comporte comme un "terminal virtuel" qui vous permet des opérations comme le mouvement et l'adressage du curseur, l'édition de la ligne, l'effacement de l'écran, l'usage de touches de contrôle, et de requêtes sur la position du curseur. Les séquences de contrôle suivent le standard ANSI. Cela vous permet de définir votre propre terminal (indépendamment de la plate-forme sur laquelle vous êtes) pour des programmes comme des éditeurs de textes, des client emails, ou des émulateurs telnet.

Les fonctionnalités de la console s'appliquent aux entrées et aux sorties. Pour les entrées, les touches de fonctions seront converties en séquences d'échappement de plusieurs caractères. Pour les sorties, les séquences d'échappement peuvent être utilisées pour contrôler l'affichage du texte dans la fenêtre de la console. Aussi bien les entrées que les sorties commencent avec le caractère d'échappement ANSI : 27 en décimal (1B en hexa). Le caractère suivant dans la séquence indique les contrôles en entrée ou l'opération en sortie pour le contrôle du terminal. Les caractères ANSI sont sensibles à la casse et nécessitent en principe d'être en majuscules.

6.1 Séquences entrées au clavier

Les séquences saisies avec les touches de fonctions sont listées dans la table ci-dessous (pour les systèmes et les shells qui les supportent, comme Linux, BSD, etc.). Pour recevoir ces séquences en un flux de caractères non évalués, vous devez bloquer le mode "lignes" pour le port "input" de REBOL :

```
set-modes system/ports/input [lines: false]
```

A présent, vous pouvez récupérer l'entrée à partir du port (avec les fonctions **copy** ou **read-io**) ou utiliser une fonction comme input pour récupérer chaque caractère :

```
while [
    code: input
    code  13  ; ENTER
][
    probe code
]
```

Voici quelques unes des fonctions ANSI les plus courantes :

Touches de fonction	Comme séquences d'échappement	Comme bloc REBOL
F1	ESC O P	[27 79 80]
F2	ESC O Q	[27 79 81]
F3	ESC O R	[27 79 82]
F4	ESC O S	[27 79 83]
F5	ESC [1 5	[27 91 49 53 126]
F6	ESC [1 7	[27 91 49 55 126]
F7	ESC [1 8	[27 91 49 56 126]

F8	ESC [1 9	[27 91 49 57 126]
F9	ESC [2 0	[27 91 50 48 126]
F10	ESC [2 1	[27 91 50 49 126]
F11	ESC [2 2	[27 91 50 50 126]
F12	ESC [2 3	[27 91 50 51 126]
Home	ESC [1	[27 91 49 126]
End ou Fin	ESC [4	[27 91 52 126]
Page-up	ESC [5	[27 91 53 126]
Page-down	ESC [6	[27 91 54 126]
Insert	ESC [2	[27 91 50 126]
Up	ESC [A	[27 91 65]
Down	ESC [B	[27 91 66]
Left	ESC [D	[27 91 68]
Right	ESC [C	[27 91 67]

6.2 Séquences en sortie pour le terminal

Il y a plusieurs variantes dans les séquences de caractère de contrôle. Certaines commandes sont précédées par un chiffre (en ASCII), indiquant que l'opération doit être réalisée le nombre de fois indiquées par le chiffre.

Par exemple aussi, la commande permettant le déplacement du curseur peut être précédée par deux nombres séparés d'un point virgule, pour indiquer que la position du curseur doit être modifiée selon la ligne et la colonne courante. Les commandes de contrôle du curseur (les majuscules sont requises) sont indiquées dans la table suivante :

Séquence en sortie	**Description**
(1B)	Séquence d'échappement à utiliser avant les codes ci-dessous
D	Déplace le curseur d'un espace vers la gauche
C	Déplace le curseur d'un espace vers la droite
A	Déplace le curseur d'un espace vers le haut
B	Déplace le curseur d'un espace vers le bas
n D	Déplace le curseur de n espaces vers la gauche
n C	Déplace le curseur de n espaces vers la droite
n A	Déplace le curseur de n espaces vers le haut
n B	Déplace le curseur de n espaces vers le bas
r ; c H	Déplace le curseur vers la ligne r, colonne c*
H	Déplace le curseur vers le coin gauche supérieur (home)*
P	Efface un caractère à la droite de la position courante
n P	Efface deux caractères à la droite de la position courante
@	Insère un espace blanc à la position courante
n @	Insère n espaces blancs à la position courante
J	Efface l'écran et déplace le curseur dans le coin supérieur gauche (home)*
K	Efface tous les caractères entre la position courante et la fin de la ligne
6n	Place la position courante du curseur dans un buffer (input buffer)
7n	Place les dimensions de l'écran dans un buffer (input buffer)

Le coin supérieur gauche est défini à la colonne 1, ligne 1.

L'exemple suivant déplace le curseur de huit espaces vers la droite :

```
print "^(1B)[10CHi!"
Hi
```

Cet exemple déplace le curseur de sept espaces vers la gauche et efface le restant de la ligne :

```
cursor: func [parm [string!]][join "^(1B)[" parm]
print ["How are you" cursor "7D" cursor "K"]
How a
```

Pour trouver la taille courante de la fenêtre de la console, vous pouvez utiliser cet exemple :

```
cons: open/binary [scheme: 'console]

print cursor "7n"
screen-dimensions: next next to-string copy cons
33;105R
close cons
```

L'exemple précédent ouvre la console, envoie un caractère de contrôle vers le buffer de l'input, et copie la valeur retournée. Elle lit la valeur (dimension de l'écran) qui est renvoyée après le caractère de contrôle et ferme la console. La valeur renvoyée est la hauteur et la largeur, séparées par un point-virgule (;), et suivies par un "R". Dans cet exemple, l'écran fait 33 lignes et 105 colonnes.

*** : Autoscrolling**

L'affichage d'un caractère dans le coin inférieur droit de certains terminaux force la création d'une nouvelle ligne, et un réajustement de l'écran. D'autres terminaux n'ont pas le même comportement. Ces différences de fonctionnement doivent être prises en compte pour écrire des scripts REBOL multi-platformes.