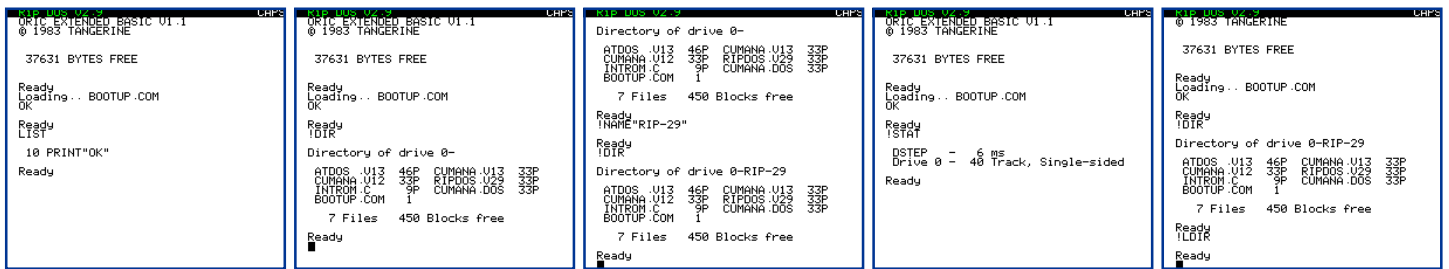


RipDOS V2.9

Manual summary - Manuel de Base

by Jim Polmear et André Chéramy



In the English part of the preceding article (Cumana v1.x), the commands include !STORE and !RECALL at the end. It is indicated in the French part that they do not work, but unfortunately not in the English part. Rip Dos v2.9 is a debugged and extended version of Cumana Dos. Let us review this Dos today. Note that the syntax of RipDOS is identical to that of Cumana Dos, unless otherwise stated. Reminder: Drive, from 0 to 3, filename, 6+3 letters, «fsp» for file specification, «asfp» for ambiguous file specification.

Summary of new or modified RipDOS commands

<!CALL address> This is a new command, added to enable a machine code routine sitting in the shadow ROM area to be called directly from BASIC. The syntax is: <!CALL address> followed by any other information / variables required by the routine called. !CALL works just like CALL, but has the shadow ROM #C000 to #FFFF paged in, instead of the normal Oric ROM. On exit (RTS), the main ROM is paged back in again. The address called can be any from #0000 to #FFFF, not necessarily above #C000.

<LDIR «asfp»> This is a new command. It works exactly the same as <!DIR>, but directs the output to a printer.

```
Directory of drive 0-RIP-29
ATDOS .U13 46P CUMANA.U13 33P
CUMANA.U12 33P RIPDOS.U29 33P
INTRON.C 9P CUMANA.DOS 33P
BOOTUP.COM 1
7 Files 450 Blocks free
```

<!DSTEP val> RipDOS modifications to <!DSTEP>: Now this command changes only the onboard information. Use <!SET> to update the system information on the disk in the default drive.

<!LOAD»fsp»(,D)(,N)(,J)(,A adresse)> RipDOS modifications to <!LOAD>: The DOS is NOT paged out if the auto start (Transfer) address is in the shadow ROM area (#C000 to #FFFF). If the T address is in normal RAM, or there is no T address specified, then the DOS is still paged out.

Jim a courageusement débogué le Cumana Dos v1.x et il en a profité pour restructurer le code, améliorer certaines commandes et en ajouter de nouvelles. Enfin, dans la place restante, il a développé une extension, qui fera l'objet des articles suivants. Pour l'heure, voici une brève revue du RipDOS v2.9, dont la syntaxe est quasi identique à celle du Cumana Dos v1.x (Ceo-Mag de février). Les différences seules seront abordées ici. Rappelons que le lecteur de disquette est désigné par un n° de 0 à 3 et que les noms de fichiers comportent 6+3 caractères.

Résumé des nouvelles commandes de RipDOS

<!CALL adresse> A partir du Basic, cette nouvelle commande permet d'exécuter une routine en langage machine située n'importe où en RAM de #000 à #FFFF. La ROM est donc invalidée au profit de la RAM overlay. Après un <!CALL>, l'adresse indiquée peut éventuellement être suivie des paramètres requis par la routine appelée. Au retour (RTS), la ROM est de nouveau validée. En résumé, <!CALL> fonctionne comme le CALL du Basic, sauf qu'avec ce dernier la routine appelée doit être située soit en RAM de #0000 à #BFFF soit en ROM de #C000 à #FFFF.

<LDIR "nom_fichier_ambigu"> Cette nouvelle commande fonctionne exactement comme <!DIR>, mais la sortie est redirigée vers l'imprimante.

<!DSTEP valeur> Analogue à la commande de Cumana Dos, sauf qu'elle n'affecte que la valeur du système en mémoire. Pour mettre à jour cette valeur sur la disquette, il faut ensuite faire un <!SET> (voir plus bas). Cette modification assainit la situation d'origine, car il fallait protéger la disquette en écriture pour limiter la mise à jour !

<!LOAD"nom_fichier"(,D)(,N)(,J)(,A adresse)> Cette commande fonctionne comme avant, sauf que si l'adresse d'exécution du programme chargé est située en RAM overlay (de #C000 à #FFFF), celle-ci (donc aussi le Dos) n'est pas invalidé au profit de la ROM comme avec Cumana Dos. Cela permet d'exécuter des routines dans la partie de la RAM overlay non utilisée par RipDOS. Dans les autres cas (pas d'exécution automatique ou adresse en RAM normale), la RAM overlay, donc le Dos, est toujours invalidée. Ceci permet d'utili-

<!NAME «**diskname**»> This is a new command, added to name or re-name a disk. «Diskname» can be up to 9 characters, and can include the drive number, as in <!FORMAT>.

<!READ **d,t,s,a**> This is a new command, added to read a defined single sector from a disk. d is the disc drive number. t is the track number (0 to 39 / 79). s is the sector number (1 to 16). a is the address at which to load the block into memory (256 bytes will be loaded). It has not been tested on a double sided disk drive.

<!SET(**drv**),(**trks**),(**sides**)> RipDOS modifications to <!SET>: This command works as previously to tell the operating system how many drive units it has and what type they are, but changes only the onboard information. <!SET> alone copies out the onboard system information to the default drive, including the DSTEP value.

<!STAT> Normally used to find out what has been set with the <!SET> command. RipDOS modifications to <!STAT>: Displays the DSTEP value too. <!STAT 0> added to display the information from the system track on the default drive.

New Feature of RipDOS

A routine has been added so that an additional table of commands and their code can be installed in an unused area of memory from #D000 to #DFFF.

The facility, which allows a new set of commands to be added, works through the normal command interpreter routine in the DOS.

If an unrecognised command is encountered, the routine looks to see if there is another look up table in page #D000. If it finds a match there, it will run the routine. If it does not find a match, it will come up with an error message.

The command word table MUST start at #D000 and each entry must be separated with a null (0). Don't forget that some words may be partly or completely tokenised. For example, FORGET will be represented by #8D (FOR) #BE (GET) #00 (null to separate).

#D000 to #D0BF is available for the new table of words, which MUST end with #FF after the null of the last entry. Memory #D0C0 to D0FF is reserved for the corresponding table of addresses for the new routines. Each address is 2 bytes long with the low byte first. There must be a 2 byte entry for each new command word in the first part of the table. Each address must be 1 less than the starting address for its routine, since 1 is added when the routine is called using the RTS instruction.

Memory #D100 to DFFF is available for the routines themselves. This is in an unused area of DOS memory, and provides huge flexibility for providing your own set of additional utilities without using RAM.

I really do need an example to show how this works!

To be followed...

ser les routines de la ROM.

<!NAME "**nom_disquette**"> Cette nouvelle commande permet de donner un nom ou de modifier le nom de la disquette (9 caractères maximum). Comme avec <!FORMAT>, "nom_disquette" peut inclure l'indication du n° de lecteur.

<!READ **d,p,s,a**> Une nouvelle commande géniale, puisqu'elle permet de lire n'importe quel secteur de la disquette et de le copier en RAM. "d" n° de lecteur de disquette, "p" n° de piste (de 0 à 39 ou 79), "s" n° de secteur (de 1 à 16) et "a" adresse de chargement des 256 octets en RAM. Pas testé avec des lecteurs double face.

<!SET **d,p,f**> Comme pour <!DSTEP>, cette commande a été modifiée pour n'affecter que les paramètres système en mémoire ("d" n° de lecteur de disquette, "p" nombre de piste (40 ou 80), "f" nombre de faces, S ou D). Il faut ensuite faire un <!SET> sans paramètre pour recopier les paramètres (ainsi que la valeur de <!DSTEP>) sur la disquette. La mise à jour se fait donc en deux temps.

<!STAT> Cette commande a été étendue. Elle permet d'afficher les paramètres initialisés avec <!SET>, mais aussi avec <!DSTEP> (valeurs lues en mémoire). Enfin <!STAT 0> permet d'afficher les mêmes informations présentes sur la disquette système.

Extension de la table des mots clés

Une routine a été installée, qui exploite une table de commandes additionnelles située dans la partie inutilisée de la RAM overlay (de #D000 à #DFFF). Cet utilitaire permet d'ajouter de nouvelles commandes en utilisant l'interpréteur normal du Dos.

Si une commande inconnue est rencontrée par le Dos, la routine regarde dans la liste de la table en #D000. Si la commande est trouvée, elle est exécutée, sinon un message d'erreur est généré.

La liste des noms de commandes supplémentaire doit commencer en #D000 et chaque entrée doit être suivie d'un zéro. Certains noms de commandes peuvent être partiellement tokenisés. Par exemple FORGET sera représenté par #8D (FOR), #BE (GET) et #00 (zéro final).

Un #FF doit être placé après le dernier #00 du dernier nom de commande. Seule la zone #D000 à #D 0BF est utilisable pour cette première table.

En effet une seconde table, de #D0C0 à #D0FF, est réservée pour les adresses d'exécution correspondantes. Chaque adresse d'exécution (en fait adresse -1 à cause du RTS utilisé pour appeler la routine) comporte deux octets (faible puis fort, comme d'habitude). A chaque nom de commande de la 1e table doit correspondre une adresse dans la 2e.

La mémoire de #D100 à #DFFF est utilisable pour implanter les routines elles-mêmes. La mise en oeuvre de ce système de mots clés supplémentaires est donc très simple. Exemples à suivre...