

SMAKY NEWS

Le 3 juillet 1987

No 45

CLE révision 4.1, par Michael Walz

Le CLE a subi dernièrement un certain nombre de transformations concernant surtout les fichiers macros. Rappelons qu'un fichier macro (avec l'extension .CLE) peut contenir plusieurs ordres du CLE, et lorsqu'on tape le nom de ce fichier depuis le CLE, ces ordres seront exécutés l'un après l'autre.

Prenons comme exemple le fichier *CALCULS.CLE* dont voici le contenu:

```
dir #mm0:stat:
basic calculs.cbases %1
```

Cette macro met le dossier STAT: comme dossier par défaut, et exécute le BASIC avec le programme CALCULS.CBAS. Si l'utilisateur tape alors "calculs test1", le CLE met STAT: comme dossier par défaut et exécute la ligne de commande "basic calculs.cbases test1". Le symbole %1 a en effet été remplacé par "test1". On peut mettre plusieurs paramètres sur dans un fichier macro .CLE. Ils seront désignés par %1, %2, %3 etc.

Tout ceci était déjà possible; les nouveaux ordres permettent, en revanche, de tester si un paramètre est présent dans la ligne de commande, de tester le numéro du SMAKY, etc. En plus on peut maintenant exécuter une macro depuis une autre macro. On peut également stopper une macro par une pression sur la touche **END**.

NB: La stratégie de recherche des fichiers dans le CLE est la suivante: Si on tape "TOTO" le CLE cherche d'abord TOTO.CODE dans le dossier système. S'il ne le trouve pas, il cherche TOTO.CODE dans le ou les dossiers définis par l'ordre **DIR**. C'est seulement lorsqu'il n'a pas trouvé TOTO.CODE qu'il commence à chercher TOTO.CLE selon la même méthode.

On peut dorénavant exécuter une macro .CLE dans une autre fenêtre. Ceci est extrêmement simple: au lieu de taper par exemple "calculs" depuis le CLE, il suffit de taper "cle calculs", et un nouveau CLE sera mis dans une fenêtre dans laquelle l'exécution de la macro CALCULS.CLE se déroulera. Si on tape "CLE" tout court, un deuxième CLE est exécuté dans une fenêtre et on peut y taper des ordres comme dans le CLE normal. L'ordre **EXIT** permet de sortir du CLE dans la fenêtre. Un CLE dans une fenêtre est appelé *CLE secondaire* tandis que le CLE du haut de l'écran est appelé *CLE principale*.

Il existe maintenant des paramètres systèmes du CLE qui dépendent de l'environnement de travail, contrairement aux paramètres utilisateurs (%1, %2 etc.), qui eux dépendent de ce que l'utilisateur tape.

1. Paramètres système

- `%P` Imprimante actuelle définie avec START, (OKI192, LQ800, LQ1500 etc.)
- `%Q` Nom abrégé de l'imprimante (LQ au lieu de LQ800, OKI au lieu de OKI192 etc.)
- `%N` Numéro de série du SMAKY
- `%M` Genre de SMAKY (100 pour SMAKY 100, SYS pour SMAKY 8)
- `%S` Numéro de la plaque SWAN. Si elle est absente, `%S` est une chaîne de longueur nulle.
- `%D` Chaîne d'accès première aux fichiers.
- `%O` Chaîne d'accès complète aux fichiers, telle qu'elle a été au démarrage de la macro. La même chaîne sera affichée par l'ordre `GDIR`.
- `%J` Date lors du démarrage de la macro.
- `%T` Heure lors du démarrage de la macro.
- `%E` Message d'erreur correspondant à la dernière erreur survenue dans la macro. L'ordre `NOERROR` permet d'annuler ce message d'erreur. Si il n'y a pas eu d'erreur depuis le dernier `NOERROR`, ou depuis le début de la macro, `%E` est une chaîne nulle.

2. Commandes uniquement valables dans les macros

`.PARDEF chaîne1 chaîne2 chaîne3 ...`

Remplace %1 par chaîne1 si le paramètre 1 n'a pas été spécifié, idem avec le paramètre 2, etc.

Exemple: Soit la ligne de commande: "TEST T1 T2 T3" et la macro:

```
.PARDEF X1 X2 X3 X4 X5  
.MESSAGE/S %1 %2 %3 %4 %5
```

Le CLE affichera: "T1 T2 T3 X4 X5"

`.MESSAGE message quelconque`

Affiche le texte qui suit. Le texte restera jusqu'à la fin de l'exécution de la macro ou jusqu'au prochain ordre `.MESSAGE`.

`.MESSAGE/S` Attend la pression sur une touche avant de poursuivre l'exécution de la macro.

`.MESSAGE/I` Affiche le texte en inverse vidéo.

`.ELSE`

Alternative dans une exécution conditionnelle. Voir l'exemple plus bas.

`.IFEQ chaîne1 chaîne2`

Exécute jusqu'au prochain `.ENDIF` ou `.ELSE` si la chaîne1 est équivalente à la chaîne2. Chaîne2 peut contenir des * ou des ?.

Exemple:

```
.IFEQ %1 N*  
    basic  
.ELSE  
    edit  
.ENDIF
```

Si cette macro s'appelle TEST.CLE et que on tape "TEST NON" depuis le CLE, le BASIC sera exécuté, si l'on tape autre chose que "TEST N...", c'est EDIT qui sera exécuté.

- Le mécanisme de fonctionnement pour les autres ordres `.IF...` est le même que pour `.IFEQ`.
- Plusieurs structures `.IF` peuvent être emboîtées.

`.IFF xyz.bas`

Teste si le fichier xyz.bas existe

Exemple

```
.IFF %2
... (traitement si le fichier existe)
.ELSE
.MESSAGE/S Le fichier %2 n'existe pas !
.ENDIF
```

`.IFP P`

Teste si le paramètre P est présent.

```
.IFP %E   teste si une erreur a eu lieu
.IFP %S   teste si la plaque SWAN est présente
.IFP %I   teste si l'utilisateur a donné un paramètre
```

`.IF question quelconque`

Pose une question. Si on répond "Oui", l'exécution sera poursuivie jusqu'au prochain

`.ENDIF` ou `.ELSE`.

`.IF/I` Idem, mais la question est affichée en inverse vidéo.

`.IFU unité`

Teste si l'unité spécifiée est installée.

```
.IFU #printer
.IFU $FLO_1
```

`.LABEL X` Permet de placer une étiquette dans la macro pour l'ordre `.GOTO`.
X peut être un chiffre de 0 à 9.

`.GOTO X` Sauter à la ligne qui commence par ".LABEL X".

`.END` Stoppe la macro, même si elle est en exécution; une macro s'arrête automatiquement à la fin de son fichier.

`.EXIT` Stoppe la macro en cours d'exécution. Si la macro était exécutée dans un CLE secondaire, stoppe aussi celle-là. L'ordre `.EXIT` peut aussi être exécuté à la main depuis un CLE secondaire.

`.PRIORITY nombre`

Change la priorité du CLE. Attention, cet ordre est à utiliser avec précaution. La priorité normale du CLE est de 6. On ne peut pas changer la priorité du CLE *principal*.

NB: Les exemples ne servent qu'à illustrer l'explication.

3. Commandes concernant les fichiers

RENAME fichier1 fichier2

Renomme le fichier1 en fichier2 (comme dans le FILER).

DELETE fichier

Détruit un fichier en posant la question si l'on est sûr, comme dans le FILER.

DELETE/A Ne pose pas la question.

MAKE fichier

Crée un fichier vide.

SDIR rep:

Spécifie un nouveau dossier système.

GSDIR

Affiche le dossier système actuel.

4. Exemple

REP.CLE: Cette macro met un dossier par défaut. On peut mettre les ":" à la fin du dossier ou non. Cette macro peut avoir une vraie utilité. Si on désire travailler dans le dossier BASIC:, il suffit de taper depuis le CLE "REP BASIC"

<u>macro</u>	<u>explication</u>
abort no	ne stoppe pas en cas d'erreur
.ifp %1	paramètre 1 existe ?
rel *	rel total
.ifeq %1 *:	%1 termine par ":"
dir #mm0:%1,SYSTEM:,#mm0:	oui
.else	
dir #mm0:%1:,SYSTEM:,#mm0:	non
.endif	
.else	
Message/s argument manquant	parametre 1 n'existe pas
.endif	
.IFP %E	
.MESSAGE/S/I %E	Affiche l'erreur
rel *	
dir %0	Remet l'ancien dossier
.ENDIF	

5. Quelques conseils

- Les macros devraient être écrites à l'aide du programme EDIT.
- Le déverminage de macros plus compliquées se fait aisément dans un CLE secondaire, en ayant provisoirement placé l'ordre **DISPLAY YES** au début de la macro.
- Avant de vous lancer dans des essais, vérifiez à l'aide de la commande **HEADER/V** du FILER que votre 100_CLE.CODE a bien la revision 4.1