Pom's

TimeOut UltraMacros

de Randy Brandt

La puissance d'un macro-langage au service d'AppleWorks™

Adaptation française pour AppleWorks 1.4

Daniel Lurot

Traduction du manuel

Dimitri Geystor

Copyright et diffusion exclusive de la version française

Éditions MEV – Revue Pom's
12, rue d'Anjou 78000 Versailles
© (1) 39 51 24 43 — Serveur Minitel : (1) 39 53 04 40

Publié par

Beagle Bros, Inc.
6215 Feris Square, Suite 100 — San Diego, CA 92121

Chapitre 1
UltraMacros de TimeOut : présentations

UltraMacros est le plus puissant outil de la nouvelle famille des applications TimeOut pour AppleWorks. À AppleWorks, et à toutes les applications TimeOut, il adjoint l'usage des macro-commandes et de la souris. Ceux qui ont déjà pratiqué Super MacroWorks découvriront un air de ressemblance : c'est normal, car UltraMacros est du même auteur, et Randy Brandt s'est efforcé de maintenir un maximum de compatibilité.

Mais UltraMacros n'est pas "un autre SMW" : en fait, c'est un langage de programmation complet, qui contient SMW ! Ceux qui, avec SMW, se sentaient "gênés aux entournures" vont pouvoir s'en donner à cœur joie.

Les tous nouveaux venus, ceux qui n'ont jamais manié une macro-commande, y trouveront aussi leur bonheur : le maniement est simple, et l'apprentissage progressif.

Note : Les versions françaises TimeOut fonctionnent avec une version spéciale d'AppleWorks 1.4. Celle-ci est créée automatiquement lors de l'installation des applications TimeOut francisées. De plus, chaque version française des applications TimeOut a été elle-même modifiée (par rapport à son homologue américain) pour pouvoir fonctionner avec l'AppleWorks 1.4. Les versions américaines de TimeOut ne fonctionneront donc pas sur l'AppleWorks 1.4 (même modifié).

Mais, au fait, qu'est-ce qu'une macro-commande ?

Une "macro" vous permet, en un seul geste, de remplacer plusieurs (ou beaucoup) d'autres commandes. On utilise la touche ⌘ (ou "option" sur le GS, appelée 'sa-' dans les pages qui suivent) en association avec une autre touche. Gain de temps, gain de précision aussi...

Par exemple, où que vous soyez dans AppleWorks, tapez sa-£ : la macro ouvre un nouveau fichier traitement de texte, fait la mise en page, inscrit la date et met le curseur au début de l'adresse du destinataire, qui s'imprimera (quand vous l'aurez écrite) en caractères gras à l'endroit prévu pour une enveloppe à fenêtre.

La touche Pomme-Fermée (ou Option) étant résevée aux macros, la touche Pomme-Ouverte (que nous appelerons 'oa-') fonctionne comme d'habitude, toutes les commandes habituelles d'AppleWorks restent présentes.
UltraMacros est un logiciel très souple, qui convient au débutants comme aux utilisateurs avertis. Un ensemble de macros "prêtes à l'emploi" est installé d'office avec UltraMacros. Le fichier MACROS.ULTRA.VF (sur la disquette) contient les macros, ainsi qu'un bref commentaire. Les familiers de Super MacroWorks ne seront pas dépaysés — mais attention, tout n'est pas pareil, tant s'en faut.
La disquette UltraMacros contient d'autres fichiers d'exemples, dont chacun pourra s'inspirer.

Que vous propose UltraMacros ?

Un simple coup d'œil à la table des matières du chapitre 4 vous donnera une idée de ce que permet UltraMacros.

De nouvelles commandes AppleWorks

UltraMacros ajoute en permanence à AppleWorks de nouvelles commandes sa- et de nouvelles commandes oa-, qui complètent utilement celles dont AppleWorks est déjà doté. Par exemple, si votre Apple est équipé d'une horloge, sa-= imprimera l'heure. De même, sa-' imprimera la date, etc. Une liste complète et commentée de ces nouvelles commandes est donnée à partir de la page 14.

La souris partout dans AppleWorks

UltraMacros installe la souris dans votre AppleWorks. L'application TimeOut "Macro Options" (fournie avec votre disquette UltraMacros) vous en donne un contrôle complet, sans sortir d'AppleWorks et sans avoir à relancer aucun autre logiciel :

— Activer/Désactiver la souris
— Régler la vitesse de réaction du clic de la souris
— Régler la vitesse de déplacement horizontale
— Régler la vitesse de déplacement verticale

Un véritable langage de programmation

C'est sans doute le principal mérite d'UltraMacros, et ce qui en fait un logiciel absolument sans limites : il met en œuvre un véritable langage de programmation, qui non seulement peut manipuler les commandes d'AppleWorks, mais qui possède son propre jeu d'instructions, avec paramètres, acceptant des variables numériques et des variables chaîne (string), ses instrucions conditionnelles, etc. Pour les programmeurs chevronnés, les instructions <call>, <peek> et <poke> ouvrent la porte aux incursions — et excursions — en langage machine.

En prime : l'élimination d'un bug

Un bug bien connu d'AppleWorks l'empêchait d'utiliser le code Control-à pour les instructions à l'imprimante. Une option du menu d'installation d'UltraMacros vous en débarrasse sans douleur. À propos de commandes d'imprimante, essayez la macro MC.IMPRIMANTE, elle vous donnera une petite idée des capacités de programmation d'UltraMacros.

Chapitre 2
Installation

Ce chapitre vous explique comment installer TimeOut, et ajouter les comandes d'UltraMacros sur votre disque AppleWorks (face AMORCAGE si votre AppleWorks est sur disquette 5,25'). Pour disposer des macros et des nouvelles commandes, il vous faut donc faire deux modifications successives à AppleWorks :

1. Il faut installer TimeOut, qui est indispensable pour utiliser les applications TimeOut associées à UltraMacros.
2. Il faut aussi, évidemment, installer UltraMacros.

Ces opérations sont automatiques, il vous suffit de lancer la disquette UltraMacros et suivre les indications des menus. TimeOut UltraMacros est livré sur disquettes 3,5' et 5,25'. Avant toute chose, prenez la précaution de faire des copies de sauvegarde, avec lesquelles vous travaillerez ; rangez les originaux en lieu sûr.

Les disques TimeOut UltraMacros ne sont pas protégés, leur copie est possible avec n'importe quel utilitaire du commerce.

Les logiciels TimeOut, bien que non protégés, sont sous copyright. En plus du travail créatif des Beagle Brothers, l'adaptation française des applications TimeOut est l'aboutissement d'un travail considérable : programmation et francisation (Daniel Lurot), traduction de la documentation (Dimitri Geystor), mise en page, impression et diffusion (Pom's et Éditions MEV). Tout a été fait pour assurer un maximum de convivialité à ces logiciels : aidez nous en les faisant connaître, mais sans donner de copies illégales à vos amis ! Merci d'avance.

Compatibilité de TimeOut, version française

TimeOut est la pièce maîtresse autour de laquelle s'articulent les diverses applications. C'est le programme qui, en modifiant AppleWorks, permet d'accéder ensuite aux diverses applications de l'intérieur même d'AppleWorks. De la sorte, les diverses applications TimeOut sont totalement intégrées à l'"environnement" AppleWorks. Cette innovation révolutionnaire leur apporte puissance, vitesse et facilité d'emploi. La version française de TimeOut fonctionne avec AppleWorks 1.4, qu'elle modifie automatiquement lors de son installation. Cette modification ne change rien au fonctionnement d'AppleWorks tel que vous le connaissez qui, pour tous les usages habituels que vous en faites, restera

le même. Les applications TimeOut francisées ont elles aussi été modifiées, par rapport à leurs homologues américaines, pour être compatibles avec AppleWorks 1.4.
Les Éditions MEV (Revue Pom's) sont le seul distributeur légal des versions françaises de TimeOut et des applications TimeOut des Beagle Bros, Inc.
Si vous associez à TimeOut d'autres patches ou améliorations d'AppleWorks, il est recommandé d'installer TimeOut en dernier.

Une vue d'ensemble de TimeOut et d'UltraMacros

Avant d'aborder l'installation de TimeOut, voici quelques indications sur son fonctionnement.

"TimeOut" est le nom générique de toute une série d'applications nouvelles. Une "application TimeOut" est un programme que vous pouvez lancer sans quitter AppleWorks, juste en appuyant simultanément sur les touches Pomme-Ouverte-Escape ; Escape vous ramène à AppleWorks là où vous l'avez quitté. Chaque application principale TimeOut (comme SuperFonts, Graph, etc) est vendue séparément, mais toutes ont cette faculté de s'intégrer à AppleWorks, grâce au "noyau" TimeOut.

Pour que les applications TimeOut fonctionnent, il faut :

1. "Améliorer" AppleWorks par la "greffe" du noyau TimeOut. Cette opération se fait une fois pour toutes, et elle concerne (si vous avez des lecteurs 5,25') la face AMORCAGE d'AppleWorks.
2. Une fois qu'AppleWorks a été ainsi "amélioré", il faut installer les applications TimeOut sur le disque des applications. Elles seront "pilotées" par le noyau TimeOut, directement à partir d'AppleWorks.

Mais le fait qu'une application soit lancée à partir d'AppleWorks ne signifie pas du tout que le fichier-programme correspondant doit forcément résider sur le disque d'AppleWorks. En fait, pour des raisons de place (surtout si vous avez des lecteurs 5,25'), ces applications seront souvent installées sur un — ou plusieurs — disques différents.

Le noyau TimeOut saura retrouver automatiquement ces applications , là où elles se trouveront. Il vous suffit d'indiquer, en installant TimeOut, le "chemin d'accès" du disque des applications. Le disque des applications peut donc être : un disque 5,25', un disque 3,5', un disque dur, un disque RAM, ou simplement un sous-volume ("dossier", ou "sous-catalogue") dans un de ces disques.

Un mot enfin du cas particulier d'UltraMacros : contrairement aux applications TimeOut décrites ci-dessus, UltraMacros lui-même n'est pas une application TimeOut comme les autres. Voilà pourquoi il faut l'installer séparément. UltraMacros aura cependant besoin, pour fonctionner de façon complète, de deux applications TimeOut : le compilateur TO.COMPILER, et les options TO.MACROS.OPT.
Une fois installé, UltraMacros est compatible avec toutes les applications TimeOut, qu'il devient possible d'automatiser grâce à ses macros.

Installation de TimeOut

Vous avez fait vos copies de sauvegarde, et vous êtes prêt à intaller TimeOut UltraMacros sur votre AppleWorks 1.4.

Note : Si vous êtes déjà un familier de Super MacroWorks, et que celui-ci est installé sur votre AppleWorks, vous pouvez quand-même y installer TimeOut et UltraMacros (dans cet ordre), il n'y a pas d'incompatibilité. Lisez la section "SuperMacroWorks, qu'en reste-t-il ?", page 32. Si vous craignez de vous embrouiller, partez d'une copie d'AppleWorks vierge.

Lancez votre copie de la disquette UltraMacros. Dans le premier menu, tapez T pour "Installer TimeOut VF". Vous arrivez au menu d'installation de TimeOut, qui vous présente les options suivantes :

```
1. Améliorer AppleWorks (Installer TimeOut)
2. Copier les applications TimeOut sur leur disque
3. Démarrer AppleWorks
4. Quitter  (retour au Basic Applesoft)
```

Améliorer AppleWorks

Note : Cette option est destinée à installer le noyau TimeOut sur votre AppleWorks 1.4. Si cette opération a déjà été faite, vous n'avez pas besoin de la répéter. Mais il vous faut copier sur le disque d'applications TimeOut les fichiers TO.' propres à UltraMacros ; dans ce cas, allez directment à l'option 2.

En cours d'installation de TimeOut, diverses options vous sont proposées. Elles sont commentées ci-dessous.

Désirez-vous que le menu TimeOut soit trié ?

Si vous répondez <OUI>, les noms des applications (quand vous les appellerez à partir d'AppleWorks) seront triés par ordre alphabétique. Sinon, leur ordre d'affichage sera celui du Catalogue du disque des applications.

Utiliserez-vous plusieurs disques d'applications TimeOut ?

Si vous avez des lecteurs 5,25', et plusieurs applications TimeOut, la réponse à donner est certainement <OUI>. Ainsi, TimeOut pourra afficher dans son menu une liste d'applications provenant de plusieurs disques différents.
Si la place vous manque pour ranger toutes vos applications TimeOut sur un disque unique, vous pourrez aussi gérer vos diverses applications au moyen de l'option "Ajouter des applications" du programme "Utilitaires" de TimeOut (voir annexe, page 35). Avec cette option des Utilitaires, il est possible d'attribuer un menu différent à chaque disque d'applications.

Note : Ne confondez pas ces deux possibilités : l'option "Utiliserez-vous plusieurs disques d'applications TimeOut" vous permet de décider d'avance, dès l'installation de TimeOut, que vous mettrez dans un même menu les applications de plusieurs disques différents. L'option "Ajouter des applications" des Utilitaires vous permet de construire un menu différent pour chaque disque d'applications, et d'aller chercher ces applications en cours de travail, même après avoir démarré AppleWorks.

Emplacement de vos applications TimeOut

Nous avons déjà vu que les programmes d'applications TimeOut peuvent fort bien résider sur un disque différent de celui d'AppleWorks. C'est même le cas le plus courant. Il faut donc indiquer à TimeOut comment accéder à ces applications. Par "disque d'applications" il faut donc entendre tout "disque" ProDOS (5,25', 3,5', disque RAM, disque dur, ou un de leurs "sous-volumes") qui contient les fichiers des applications TimeOut (le nom de ces fichiers commence toujours par TO.).

Note : Dès que vous démarrez votre AppleWorks amélioré, TimeOut "ausculte" rapidement le ou les disques d'applications et repère tous les fichiers qui commencent par TO.' ; quand, au cours de votre séance de travail AppleWorks, vous appelez TimeOut en tapant Pomme-Ouverte-Escape, TimeOut affiche instantanément, sous forme de menu, la liste des applications disponibles. Vous pouvez sélectionner celle qui vous convient, et TimeOut sait où aller la chercher.

Si vous avez des lecteurs 5,25', il est commode d'indiquer "plusieurs disques" à l'option précédente, et de désigner l'emplacement du disque d'applications par son port et lecteur. Si vous avez un lecteur 3,5', il peut être commode d'installer les applications sur le même disque qu'AppleWorks. Si vous avez un disque dur, indiquez l'emplacment par son nom d'accès ProDOS.

Emplacement du disque AMORCAGE AppleWorks

Le noyau TimeOut sera "greffé" sur la partie AMORCAGE d'AppleWorks. Il faut donc indiquer où se trouvera votre AppleWorks au moment de l'installation de TimeOut. Indiquez soit le port et lecteur, soit le nom d'accès ProDOS, insérez au besoin votre disquette AppleWorks (face AMORCAGE si c'est un disque 5,25'), faites RETURN. La procédure d'installation est enclenchée.

Note : Cette procédure se fait, en principe, une fois pour toutes. Même si vous ajoutez d'autres applications TimeOut par la suite, vous n'aurez plus besoin de passer par la procédure "améliorer AppleWorks", sauf dans le cas envisagé ci-dessous.

Réinstallation de TimeOut

Si, après avoir installé TimeOut, vous souhaitez spécifier un autre emplacement pour le disque des applications TimeOut, ou changer leur affichage (trié/non trié), vous pourvez réinstaller TimeOut par-dessus l'ancien. Cette opération se fera sans problème, à moins que vous n'ayez "patché" AppleWorks entre-temps avec d'autres "améliorations", auquel cas la réinstallation risque de ne pas réussir ; il vous faudra alors recommencer de zéro.

Note : L'option "Ajouter des applications" des Utilitaires TimeOut permet d'ajouter très facilement des applications sans avoir à réinstaller TimeOut (voir annexe, page 35).

Copier des fichiers TO. sur le disque d'applications

Si UltraMacros est (pour le moment) votre seul disque TimeOut, il peut vous servir de disque d'applications, et vous n'aurez pas besoin de cette option. Cette option sert simplement à recopier sélectivement tous les fichiers TO. sur votre disque d'applications. Indiquez le disque de destination (Nom ProDOS ou port/lecteur), insérez-le dans le lecteur approprié, et appuyez sur la barre ESPACE. Vous pourrez suivre visuellement la recopie des fichiers TimeOut.

Quand la copie est terminée, appuyez à nouveau sur la barre ESPACE pour revenir au menu principal.

Note : UltraMacros a besoin des fichiers TimeOut suivants : TO.COMPILER, TO.MACRO.OPT, TO.UTILITAIRES, TO.CLIPBOARD

Installation d'UltraMacros

1. Le noyau TimeOut a bien été installé sur AppleWorks (face AMORCAGE pour les disques 5,25') ? Vérifiez qu'AppleWorks fonctionne correctement. Si vous utilisez une carte d'extension de mémoire, faites-en l'installation avant d'installer UltraMacros.
2. Démarrez le disque UltraMacros, et à partir du menu principal, choisissez U pour "Installer UltraMacrs.VF". Le macro-programme (ULTRA.SYSTEM) sera chargé en mémoire.
3. Retirez le disque UltraMacros et insérez le disque AppleWorks (face AMORCAGE pour les disques 5,25'). Appuyez sur Return.
4. Le programme d'installation va modifier le fichier APLWORKS.SYSTEM (ou SUPER.SYSTEM) de démarrage AppleWorks. S'il ne le trouve pas, un message à l'écran vous demandera d'indiquer l'emplacement de ce fichier. S'il se trouve dans un sous-catalogue ProDOS, spécifiez son chemin d'accès (sous la forme "/NOM.DISQUE/SOUS-CATALOGUE").
5. Si les indications sont correctes, l'installation se poursuit. Les nouvelles commandes et macros sont mises en place, et un message à l'écran vous avertit quand tout est terminé.

Démarrer AppleWorks amélioré

— Vous avez installé TimeOut ?
— Vous avez constitué un "disque des applications TimeOut" ?
— Vous avez installé UltraMacros ?
Alors, vous êtes prêt à démarrer votre "nouveau" AppleWorks. Allez-y, bootez le disque...
Avant d'arriver au menu principal d'AppleWorks, vous devez voir brièvement apparaître l'écran d'identification de TimeOut.

Note : Dès que vous démarrez votre AppleWorks amélioré, TimeOut "ausculte" rapidement le ou les disques d'applications et repère tous les fichiers qui commencent par TO. ; ensuite, quand vous appelez TimeOut en tapant Pomme-Ouverte-Escape, TimeOut affiche instantanément, sous forme de menu, la liste des applications disponibles. Vous pouvez sélectionner celle qui vous convient, et TimeOut sait où aller la chercher.

— Si TimeOut vous affiche un message "Erreurs en lisant ...", c'est qu'il n'arrive pas à localiser les applications. Il vous faut insérer le disque d'applications (si vous ne l'avez pas fait) ou indiquer un autre emplacement. Si vous avez bien inséré le disque d'applications et que vous avez quand-même des messages d'erreur, vérifiez s'il y a bien des applications sur le disque, ou s'il n'a pas été endommagé. Le cas échéant, créez un nouveau disque d'applications.
— Si l'écran d'identification TimeOut ne s'affiche pas au démarrage, c'est que l'installation de TimeOut ne s'est pas

faite. Revenez au point de départ, relisez attentivement le manuel, et recommencez l'installation.

À mesure que TimeOut identifie chaque application, le nom de celle-ci apparaît à l'écran.

Chargement des applications en mémoire : au moyen des Utilitaires, vous pouvez demander à TimeOut de charger des applications en mémoire vive dès le démarrage d'AppleWorks ; leur accès sera ensuite quasi-immédiat. La façon de charger les applications en mémoire est expliquée dans l'Annexe A "Utilitaires". Dans le menu, le nom des applications chargées en mémoire sera marqué d'un astérisque. Pour empêcher ce chargement en mémoire, maintenez enfoncée la touche ESCAPE : l'application sera disponible quand-même, mais résidera sur disque.

Si en installant TimeOut vous aviez choisi l'option "Plusieurs disques d'applications", lors du démarrage d'AppleWorks TimeOut fera une pause et vous demandera s'il y a un autre disque d'applications à lire : insérez le disque suivant et répndez "Oui", sinon répondez "Non".

Appel des applications TimeOut

À tout moment, quand vous travaillez avec AppleWorks, vous pouvez appeler une application TimeOut en appuyant sur Pomme-Ouverte-Escape : une fenêtre-menu s'affiche à l'écran, avec la liste des applications disponibles.
Sélectionnez avec les flèches (ou la souris si vous avez installé SuperMacroWorks ou UltraMacroWorks), validez avec RETURN (Escape vous ramène à AppleWorks).
Si vos applications ne résident pas en mémoire, il faut que votre disque d'applications soit en ligne au moment de valider.

Occupation mémoire

Vous constaterez que TimeOut réduit légèrement la mémoire dont dispose AppleWorks. Les applications qui "résident en mémoire" en consomment elles aussi. Si vous vous trouvez à court de place sur votre Bureau AppleWorks, employez les Utilitaires TimeOut pour "purger" ces applications de la mémoire (elles resteront disponibles sur disque, mais leur accès sera moins rapide).

Une bonne solution est d'installer vos applications sur disque RAM : elles n'envahiront pas le Bureau d'AppleWorks, et l'accès restera très rapide (c'est là aussi qu'il est conseillé d'installer les polices et images de TimeOut SuperFonts).

Patch Control-Reset

Quand TimeOut est installé, il modifie AppleWorks pour que Control-Reset renvoie au menu principal d'AppleWorks au lieu d'envoyer dans le moniteur.

C'est "une bouée de sauvetage" en cas de plantage, et non un moyen recommandé pour aller au menu principal !

Chapitre 3
UltraMacros : prise en main

Ce chapitre est une entrée en matière à l'immense univers que vous ouvre UltraMacros. Il vous familiarisera avec certaines des nouvelles commandes qu'il ajoute à AppleWorks. Nous supposerons que vous avez démarré votre AppleWorks et que vous avez ouvert un fichier de Traitement de texte (UltraMacroWorks fonctionne partout dans AppleWorks, mais un traitement de texte est un endroit commode pour "faire ses premiers pas").

Les nouvelles commandes qu'apporte UltraMacros

UltraMacroWorks ajoute, aux commandes habituelles d'AppleWorks, plusieurs commandes nouvelles, du type - (sa- ou touche "option" sur le IIGS) et Pomme-Ouverte- (oa-). Faites les quelques essais suivants :

1. Tapez SA-' pour imprimer la date (celle fixée au démarrage d'AppleWorks), en toutes lettres. Si rien ne s'imprime et qu'AppleWorks vous envoie un "bip", il est probable que vous n'avez pas installé UltraMacroWorks. Revenez au chapitre précédent.
2. Tapez votre prénom, placez le curseur sur la première lettre, et tapez OA-/ ju squ'à la fin du prénom. Tapez SA-0 : votre prénom s'écrit : il a été lu lettre par lettre avec OA-/, enregistré dans la "memoire zéro", et restitué avec SA-0.
3. Placez le curseur au milieu d'un mot, tapez SA-Del ou OA-Del : le caractère sous le curseur est "avalé". Ces commandes permettent de rattraper une faute de frappe en restant en mode insertion.

Toutes ces nouvelles commandes sont installées d'office avec UltraMacros. Elle ne sont pas modifiables, et vous pouvez les utiliser partout dans AppleWorks où l'écriture est possible et que le curseur est visible. Par exemple, vous pouvez employer OA-Del après avoir tapé OA-N pour changer le nom d'un fichier, ou taper OA-' quand vous imprimez une base de données et que vous êtes priés de donner la date.

Une liste complète des nouvelles commandes est donnée aux pages page 14 et suivantes. Vous pouvez vous y reporter, et continuer à les esayer, ou passer à la section suivante qui vous explique comment saisir vos macros directement au clavier.

Saisie de vos macros directement au clavier

Comme SuperMacroWorks, UltraMacros permet de saisir directement des macros au clavier. Voici quelques exemples à l'intention de ceux qui abordent ce genre d'exercice pour la première fois... Le but de l'opération est de "condenser" en une commande unique tout une chaîne de commandes AppleWorks.

1. Placez-vous dans un nouveau fichier de traitement de texte. Appuyez sur OA-W. Le titre en haut de l'écran doit afficher SAISIE MACRO. S'il ne le fait pas, UltraMacros n'est sans doute pas installé. Relisez le chapitre précédent.
2. En bas à gauche de l'écran, un message dit "Quelle touche ?". Appuyez sur "E" (ce sera la macro "Essai", que vous activerez ensuite en tapant SA-E). Un autre message, en bas à droite, annonce "Saisie de E", et le curseur s'arrête de clignoter.
3. Vous êtes maintenant en "mode saisie", et tout ce que vous allez taper au clavier, y compris les déplacements (avec les flèches et la souris) va s'enregistrer dans la nouvelle macro E. Par exemple, tapez votre nom, RETURN, votre adresse (avec un RETURN après chaque ligne).
4. Maintenant tapez "CONTROL-à" pour arrêter la saisie. Le curseur clignote à nouveau, et en bas à droite un message dit "Enregistré E".

Vous venez de terminer votre première saisie. Allez à la ligne, tapez SA-E : votre nom et votre adresse s'écrivent en un clin d'œil. Poursuivons l'expérience.

1. Tapez à nouveau OA-W suivi de "E". Un nouveau message s'affiche en bas à gauche :"Remplacer macro globale E? Non Oui" : en effet, comme la macro E est définie en ce moment, UltraMacros vous met en garde. Vous pouvez l'écraser (Oui) ou la préserver (Non).
2. Tapez "O" pour "Oui". Vous pouvez maintenant saisir une nouvelle macro E. Tapez "OA-1" (le curseur saute au début du fichier). Tapez "CONTROL-à". La macro est enregistrée.
3. Tapez OA-9 pour aller en fin de fichier, et testez OA-E : le curseur revient au début. Il est possible de "saisir" dans une macro n'importe quelle commande d'AppleWorks, à partir de n'importe quelle partie d'AppleWorks, et même à partir d'applications TimeOut.

   Seules restricions : vous ne pouvez pas taper OA-W pendant que le menu des applications TimeOut est affiché, ni au milieu d'une commande OA-0 (getstring). Mais vous pouvez parfaitement commencer avec OA-W dans AppleWorks, et taper ensuite OA-Escape ou OA-0 en cours de saisie.

La plupart des touches peuvent servir à nommer une macro. Certaines pourtant son interdites (elles sont "réservées"). Par exemple, tapez OA-W et RETURN. En bas à droite vous aurez le message :"Réservé ^M". En effet, RETURN est équivalent à CONTROL-M, et c'est une touche réservée.

Vous pouvez aussi définir des macros "DOUBLE-POMME" (notées ⌘ ou BA-), qui seront activées en enfonçant simultanément trois touches : les deux touches ⌘, et la touche de définition. Pour enregistrer une macro "Double-Pomme", il faut enfoncer Pomme-Ouverte au moment de définir la touche. Exemple :

1. Tapez OA-W, E, o, votre nom, Ctrl-à.
2. Tapez OA-W, OA-E, le nom d'un ami, Ctrl-à.
3. Essayez les macros SA-E et BA-E. Elles impriment bien deux noms différents.

Capacité : Une macro saisie au clavier peut enregistrer jusqu'à un maximum de 2000 à 4000 "frappes" (celà dépend du nombre de macros déjà actives par ailleurs. En cas de saturation, le message "enregistré" apparaîtra dès le début de la saisie, sauf si vous écrasez une macro précédente (vous disposerez du nombre de "frappes" libérées par cette macro.
La seule exception est la macro 0 (zéro), pour laquelle vous disposez toujours d'un maximum de 80 frappes, et qui s'arrête automatiquement à ce maximum (lisez la section consacrée au cas spécial de la macro 0, page 31).

Toutes les macros enregistrées avec OA-W sont perdues si vous quittez AppleWorks. Mais il y a deux façons de les rendre permanentes :

1. En recourant à l'option "3. Sauver la table des macros", des Macro Options (c'est une application TimeOut, fournie avec UltraMacros (voir page 28). Cette option sauve toutes les macros actives, en bloc, qui se retrouveront telles quelles au redémarrage d'AppleWorks.
2. En créant un nouveau fichier de macros, fait sur mesure : c'est la procédure décrite ci-dessous.

Création de nouvelles macros

Vous pouvez vous constituer un nouveau fichier de macros "sur-mesure", en éditant les macros fournies avec UltraMacros, en les complétant par des macros saisies au clavier, en compilant le tout et en sauvant ces macros sur disque.

Adaptation des macros existantes

Les "macros existantes" sont celles qui sont fournies avec UltraMacros, et installées d'office au démarrage d'AppleWorks. Ces macros (exception faite de quelques macros réservées) peuvent être modifiées ou remplacées à votre gré. Eles n'ont rien de définitif : c'est un jeu d'exemples, à partir desquel vous pouvez broder.
Il est quand même chaudement recommandé au débutants de bien les étudier d'abord, et d'en garder une copie intacte !

Comment adapter :

1. Avec AppleWorks, chargez le fichier MACROS.ULTRA.VF. C'est le jeu de "macros existantes" dont vous disposez quand vous avez modifié AppleWorks avec UltraMacros. Imprimez-le pour pouvoir l'examiner et vous y reporter au besoin.
2. En vous aidant du manuel, essayez-vous à adapter une macro. Pour la tester, tapez Pomme-Ouverte-Escape, et sélectionnez "Macro Compiler" dans le menu TimeOut. Choisissez "Compiler nouvelles macros", et faites RETURN : vous "verrez" la compilation se faire sous vos yeux. En cas d'erreur, un message vous avertira (pour plus de détails, voir le chapitre "Références", page 10).

Note : Si vos ambitions vous entrainent, au-delà d'une simple adaptation, vers l'écriture de macros entièrement nouvelles, UltraMacros a tout ce qu'il faut pour vous combler. Mais une étude approfondie du Chapitre 4 s'impose au préalable.

Création d'un fichier de macros

Un fichier de macros n'est rien d'autre qu'un fichier de Traitement de texte AppleWorks contenant des définitions de macros construites selon une certaine syntaxe. Il peut donc être sauvé sur disque comme n'importe quel autre fichier AppleWorks.

Pour activer les macros qu'il contient, il suffit de le charger dans le Bureau, de l'appeler à l'écran et d'utiliser le

compilateur (l'application Macro Compiler mentionnée ci-dessus).

Création d'un macro permanente

Supposons que vous ayez apporté quelques modifications et compléments réussis au fichier MACROS.ULTRA.VF, et que vous souhaitiez en faire votre jeu de macros permanent (celui qui sera activé au démarrage d'AppleWorks). Si vous débutez dans le monde des macros, nous vous suggérons d'essayer une modification toute simple des macros sa-B et sa-N.

1. Dans sa-N, remplacez "Pom's - Editiions MEV" par votre prénom et nom.
2. Dans sa-B, remplacez : "12, rue d'Anjou" et "78000 VERSAILLES" par les éléments de votre propre adresse (ajoutez ou retranchez au besoin des lignes d'adresse en suivant le modèle).

Faites ensuite Pomme-Ouverte-Escape pour appeler le menu TimeOut, Choisissez "Macro Options", puis l'option "3. Sauver la table des macros". À la question "Activer la macro de démarrage" répondez "Non".
Le disque tourne, et le fichier ULTRA.SYSTEM de démarrage d'AppleWorks est récrit avec ce nouveau jeu de macros. C'est tout ; au prochain démarrage d'AppleWorks, ces macros seront automatiquement activées.

Note : Nous arrivons ainsi à la fin du chapitre "Prise en main". Il s'agissait de vous familiariser avec le maniement d'UltraMacros, sans plus. Le chapitre suivant fournit les "Références" nécessaires à l'utilisation et à l'écriture des macros. Le reste est affaire d'exploration et d'imagination personnelles. La disquette UltraMacros contient un certain nombre de fichiers d'exemples, que vous êtes également invités à étudier.

Chapitre 4
UltraMacros : références

Ce chapitre ne vous épargnera aucun détail de toutes les ressources d'UltraMacros. Tous les ingrédients sont là : servez-vous, et bon appétit.

Anatomie d'une macro-commande

Avant d'écrire une macro, la première chose à faire est de se familiariser avec sa structure. Il y a, pour écrire les macros, des règles de syntaxe simples mais nécessaires. Il faut les connaître et s'y conformer, c'est votre meilleur gage de réussite. UltraMacros vous signalera les erreurs les plus grossières au moment de la compilation, mais il ne les "verra" pas forcément toutes.

Les instructions ("tokens")

Examinez les macros du fichier MACROS.ULTRA.VF. Elles sont contenues entre les mots START (obligatoire) et END (facultatif). Chaque macro est composée d'un mélange de texte et d'instructions spéciales.
Une instruction ("token" en anglais), est un mot codé, inclus entre des crochets < >, qu'UltraMacros sait interpréter comme une commande. Il peut s'agir de l'équivalent d'une commande au clavier, comme par exemple <rtn>, <esc>, <left> ou <oa-1>, ou d'une instruction spécifique à UltraMacros comme <if>, <else> ou <awp>.

1. Chaque macro commence obligatoirement par indiquer :
   -> le caractère ou la commande qui seront utilisés pour déclencher cette macro, immédiatement suivi par ...
   -> deux points ":", immédiatement suivi par ...
   -> le code du domaine d'application :
      `<all>` global (tous domaines)
      `<awp>` traitement de texte seulement

`<adb>` base de données seulement
`<asp>` tableur seulement
`<ato>` application TimeOut seulement

2. Vient ensuite le contenu de la macro proprement dit, et ...
3. La macro se termine obligatoirement par "!". Tout texte qui vient après le point d'exclamation est purement explicatif, et n'a aucune incidence sur la macro (il ne consomme pas de mémoire non plus).

Voici comment écrire les instructions correspondant à des commandes du clavier :

```
<del>      touche DELETE
<esc>      touche ESCAPE
<rtn>      touche RETURN
<tab>      touche TAB
<left>     touche FLECHE-A-GAUCHE
<right>    touche FLECHE-A-DROITE
<up>       touche FLECHE-EN-HAUT
<down>     touche FLECHE-EN-BAS
<spc>      touche ESPACE
```

Les codes pour Pomme-Ouverte, Pomme-Fermée, Double-Pomme (c'est-à-dire Pomme-Ouverte ET Pomme-Fermée) et Control utilsent les abréviations oa, sa, ba et ctrl suivies d'un trait-d'union et de l'indication de la touche associée. Par exemple :

```
<oa-1>          Pomme-Ouverte-1
<sa-B>          Pomme-Fermée-B
<ba-right>      Double-Pomme-Flèche-à-droite
<sa-ctrl-C>     Pomme-Fermée-Control-C
```

Viennent s'y ajouter un certain nombre de codes d'instructions non modifiables associés aux nouvelles commandes Pomme-Ouverte (page 14) et Pomme-Fermée (page 13) d'AppleWorks.

Les instructions peuvent être écrites indifféremment en majuscules ou en minuscules, mais les espaces entre les lettres des instructions sont illégaux. Par exemple, <rtn>, <RTN> et <Rtn> sont corrects et équivalents, mais <r tn> ne serait ni reconnu, ni accepté par le compilateur.

Quand plusieurs codes d'instructions se suivent directement, il suffit d'ouvrir les crochets au début, et de les fermer à la fin. Les instructions peuvent être séparées soit par deux points ":", soit par un espace. Ainsi, tous les exemples ci-après sont corrects :

```
<up><up><left>
<up:up:left>
<up up left>
<up><up left>
```

Note : Voir page 21 le cas particulier de l'instruction <print>.

Il est possible de mettre des commentaires non seulement après une macro, mais dans le corps même de la macro, à condition :

a) de les mettre entre parenthèses, et
b) que ces parenthèses soient elles-mêmes entièrement comprises entre les deux crochets d'une série d'instructions.

Exemple :

```
a:<awp:up:(ceci n'est pas inclus dans la macro)up:left>!
```

Note : On peut mettre des parenthèses à l'intérieur des crochets, mais pas des crochets à l'intérieur des parenthèses.
Exemple :

```
a:<all bell (sonne un bip) stop>!      Ceci est correct
b:<all bell( --> Attention) stop>!     Ceci ne l'est pas
```

Le texte entre parenthèses ne consomme d'ailleurs aucune place dans la table des macros : le compilateur l'ignore purement et simplement. Par contre, si les parenthèses sont hors des crochets, elles sont traitées comme du texte ordinaire.

Note : En écrivant vos instructions, vous pouvez mettre des espaces, des commandes AppleWorks de mise en page et des retours chariots : le compilateur n'en tiendra pas compte. Par contre, la présentation peut y gagner en clarté. Attention à la façon de traiter le texte entre les instructions ! Exemple :

```
a:<awp                    (domaine : TdT)
    up                    (monte)
    up                    (monte)
   >UltraMacros<          (écrit UltraMacros)
    left                  (à gauche)
   >!                      fin
```

Voir aussi le fichier MACROS.SPECIAL, qui est un modèle du genre.

Macros globales et locales

Chaque macro est forcément locale ou globale, et il est obligatoire d'indiquer son domaine d'application. Une macro locale est limité à un seul domaine (Traitement de texte, Base de données, Tableur ou Application TimeOut). Le domaine d'application est indiqué aussitôt après le caractère ou la commande qui nomme la macro.

Dans une table de macros, le même nom ne peut désigner qu'une seule macro globale (si par hasard vous en mettez une deuxième, elle ne sera tout simplement jamais exécutée). Il peut y avoir une macro du même nom pour chaque domaine d'application.

L'ordre dans lequel les différentes macros de même nom paraissent dans la table est important ; quand vous déclenchez une macro, UltraMacros agit comme suit :

1. Il cherche, en commençant au début de la table, la première macro du même nom.
2. Quand il l'a trouvée, il vérifie son domaine d'aplication :
   -> Si c'est <all>, il l'exécute, où que vous soyez dans AppleWorks ou dans TimeOut
   -> Si c'est <ato>, il ne l'exécute que si vous êtes dans TimeOut
   -> Si le domaine est AppleWorks, il vérifie la concordance du domaine. Si le domaine concorde, il exécute la macro ; sinon, il poursuit la recherche.

Par conséquent, si vous avez plusieurs macros du même nom dans une table, il faut mettre d'abord les macros locales, puis la macro TimeOut, puis la globale.
Comme TimeOut fait partie d'AppleWorks, vous n'êtes pas obligé d'indiquer <ato> pour les macros destinées à TimeOut. L'instruction <ato> a pour seul rôle d'empêcher une macro de fonctionner en-dehors de TimeOut.

Note : En fait, une seule touche (par exemple "A" ou "a", car le compilateur ne fait pas la différence entre majuscules ou minuscules) pourra avoir jusqu'à HUIT définitions tout à fait différentes dans la même table : une commande Double-Pomme et une commande Pomme-Fermée pour chacun des domaines <awp>, <adb>, <asp> et <ato>.

Les macros saisies au clavier (avec OA-W) sont globales par défaut. Si vous voulez les transformer en macros locales, utilisez l'option "Afficher macros actuelles" du compilateur (Macro Compiler de TimeOut), changez le code du domaine d'application à la main, et re-compilez.

Appel d'une macro par une autre

Une macro peut en appeler une autre, de deux façons différentes. Ainsi, dans le fichier MACROS.ULTRA.VF, nous avons ces deux exemples :

```
y:<all><sa-left oa-D>S<down left rtn>!    Supprime une ligne
9:<awp:oa-9:up:goto sa-y>!  Supprime la dernière ligne du fichier
```

Dans la première macro, SA-Y appelle la macro SA-LEFT, laquelle envoie le curseur en début de ligne et "rend la

main" à SA-Y qui continue (et déplace toute la ligne sur le presse-papier). Dans la deuxième macro, SA-9 utilise la commande <goto> pour appeler la macro SA-Y, mais cette fois-ci sans retour.

Les programmeurs en Basic verront tout de suite l'analogie :

1. La première méthode (qui appelle une macro par son seul nom) est analogue à l'instruction GOSUB
2. La deuxième fonctionne comme le GOTO du Basic.

Les appels peuvent s'imbriquer les uns dans les autres ("macro nesting" en anglais). UltraMacros peut se souvenir de 18 niveaux d'imbrication successifs ; voilà pourquoi une macro qui s'appelle elle-même se répète 18 fois et puis s'arrête.

```
1:<all>$<sa-1>!          Imprime 18 astérisques
```

Pour maîtriser une répétition, et aller au-delà de 18, il faut utiliser des boucles avec une variable qui sert de compteur (tout ceci est expliqué plus loin). Exemple :

```
<ba-1>:<all:A=120:begin:print "*":A=A-1:if A>0 then rpt>! Imprime 120 astérisques
```

Attention : Quand vous êtes sur le point de supprimer une macro d'un fichier, assurez-vous qu'elle n'est appelée par aucune des autres macros. Si par exemple vous voulez supprimer la macro F, faites d'abord une recherche de "sa-f".

Macros réservées (commandes Pomme-Fermée, ou sa-)

Il s'agit de macros spéciales qui ne peuvent être ni modifiées, ni supprimées, ni "écrasées". Elles sont à votre disposition à tout moment (sauf indication contraire) :

-> soit de façon isolée au clavier (avec Pomme-Fermée)
-> soit lors d'une saisie de macro au clavier (avec Pomme-Fermée)
-> soit dans une définition de macro (avec le code d'instruction)

`<sa-del>` `POMME-FERMEE-DELETE`
Supprime le caractère sous le cuseur (voir aussi oa-del)

`<ahead>` `POMME-FERMEE-.`
Avance jusqu'au premier espace libre à droite du curseur. Cette macro fonctionne partout où AppleWorks autorise une écriture de texte normale (Fichiers TdT, catégories de BdD, saisies en bas d'écran, etc.).

`<back>` `POMME-FERMEE-,`
Recule jusqu'au premier espace libre à gauche du curseur.

`<date>` `POMME-FERMEE-'`
Affiche la date sous la forme 14 Juillet 1988 (format français) ou Juillet 14, 1988 (format américain). Pour passer d'un format à l'autre, utiliser l'application TimeOut "Macro Options", option "7. Autres activité", puis "8; Définir format date".

`<date2>` `POMME-FERMEE-"`
Affiche la date sous la forme 14/07/88 ou 07/14/88 (même option que ci-dessus).

`<time>` `POMME-FERMEE-=`
Affiche l'heure sous la forme 1:42 pm. Si vous n'avez pas d'horloge, l'heure sera toujours 12.00 am. Dans la Base de données, si un nom de catégorie contient "heure", AppleWorks convertira le format 12 heures en format 24 heures.

`<time24>` `POMME-FERMEE-+`
Affiche l'heure sous la forme 13:42.

`<find>` `POMME-FERMEE-RETURN`
Dans le traitement de texte :
Recherche le retour de chariot suivant, et y place le curseur. Cette commande ne fonctionne que si le message "Tapez

une information ou option" est visible en bas de l'écran.

Dans tout menu numéroté ou qui affiche les options en inverse :

Recherche dans la liste affichée le nom qui est enregistré dans la Macro Zéro, et y place le curseur. Si le nom n'est pas trouvé, la macro s'arrête, ou rend la main à la macro "appelante", si <find> était dans une macro appelée par une autre macro ; la macro appelante se poursuit.

Note : On peut ainsi prévoir une macro appelante qui recherchera un fichier dans plusieurs disques successifs, à condition de mettre un <stop> à la fin de chaque macro appelée qui contient <find> : de la sorte, si le fichier est trouvé, la macro s'arrête ; s'il n'est pas trouvé, la macro appelante passe au disque suivant, etc.

Recette : si vous êtes face à un très long menu, appuyez sur OA-0 pour définir la macro 0, tapez le nom recherché, faites RETURN puis SA-RTN : le curseur ira automatiquement sur le nom voulu. Dans une macro, c'est une instruction commode pour retrouver des fichiers par leur nom.

Cette commande fonctionne aussi dans le menu du Bureau (appelé par OA-Q) et le menu TimeOut (appelé par OA-Esc). Vos macros pourront retrouver des fichiers sur le Bureau d(après le nom du fichier, ou démarrer des applications TimeOut en les appelant par leur nom (dans ce dernier cas, définisssez la Macro 0 à l'avance, car la commande OA-0 n'est pas disponible dans le menu TimeOut).

Recette : Dans les menus d'impression, indiquez l'imprimante par <find> suivi du nom de l'imprimante si vous n'êtes pas sûr du numéro d'option qu'elle occupera.

Notes : L'instrucion <find> ne lit que les 15 premiers caractères de la Macro 0 ($0). <find> peut être associé à <store> et <recall> pour chaîner des fichiers. Voir détails page 30.

`<findpo>` `POMME-FERMEE-`

Dans le traitement de texte seulement : Avance jusqu'au prochain accent grave (`) dans le texte (utile pour retrouver une option d'impression). Cette commande ne fonctionne que si le message "Tapez une information ou option" est visible en bas de l'écran.

On peut associer à cette instruction l'instruction <screen> pour vérifier quelle est l'option d'impression qui correspond à l'accent grave. Par exemple, la macro suivante reconnaît les options de soulignement, mais passe sans s'arrêter sur toutes les autres :

```
s:<awp:findpo:$8=screen 42,24,4:if $8="Soul" then stop else rpt>!
```

`<print>` `DOUBLE-POMME-CONTROL-P`

L'instruction <print> s'emploie uniquement dans les macros "écrites" (mode d'emploi complet page 21). Elle ne peut pas s'employer du clavier, et la commande ba-ctrl-P est réservée.

Sont également réservées les commandes : sa-Ctrl-à et oa- ^.

Nouvelles commandes Pomme-Ouverte- , ou oa-

Elles sont à votre disposition à tout moment (sauf indication contraire) :

-> soit de façon isolée au clavier (avec Pomme-Fermée)
-> soit lors d'une saisie de macro au clavier (avec Pomme-Fermée)
-> soit dans une définition de macro (avec le code d'instruction)

`pas de code` `POMME-OUVERTE-W`

Début de saisie de macro au clavier.

`pas de code` `POMME-OUVERTE-DELETE`

Supprime le caractère sous le curseur. Identique à Pomme-Fermée-Delete.

Note : en saisie de macro, utiliser sa-del

`pas de code` `POMME-OUVERTE-0 (zéro)`

Présente le caractère ">" sur la ligne de bas d'écran. C'est un invite à enregistrer du texte (maximum 60 caractères) dans la Macro 0 (zéro).

Cette commande s'utilise directement du clavier. Ne pas l'utiliser pour la saisie des macros à partir du clavier.

Pour l'instruction de saisie de chaînes de texte en cours de macro, voir l'instruction <getstr>, page 20.

`<oa-ctrl-à>` `POMME-OUVERTE-CONTROL-à`

Envoie un Control-à à AppleWorks, pour des commandes d'imprimante ou d'interface. S'emploie comme instruction dans des macros écrites, ou directement pour la saisie au clavier.

Souvenez-vous que Control-à employé seul arrête net une macro.

Note : AppleWorks 1.4 a un "bug" qui l'empêche de reconnaître Control-à, ce qui est bien gênant pour les définitions d'inprimante. Le menu de la disquette UltraMacros a une option qui permet de corriger ce défaut.

`<uc>` `POMME-OUVERTE-:`

Met en majuscule le caractère sous le curseur et passe en mode recouvrement

`<lc>` `POMME-OUVERTE-;`

Met en minuscule le caractère sous le curseur et passe en mode recouvrement

`<insert>` `POMME-OUVERTE-!`

Impose le mode insertion. Pour revenir au mode recouvrement, faire oa-A.

`<zoom>` `POMME-OUVERTE-à`

"Zoom arrière" : cache toutes les options d'impression dans le traitement de textes, affiche les valeurs dans le Tableur, affiche en colonnes dans la base de donnees. Pour revenir au "Zoom avant", faire oa-Z.

`<read>` `POMME-OUVERTE-/`

À partir du clavier :

Inscrit l'un après l'autre les caractères sous le curseur dans la Macro 0 (zéro). On peut déplacer le curseur entre chaque <read>. Avant de faire une nouvele lecture, faire Esc.

En saisie de macro au clavier :

Inscrit chaque caractère sous le curseur dans la macro en cours de saisie (les caractères deviennent texte dans la définition de la macro).

Instruction <read> dans une définition de macro :

Enregistre chaque caractère sous le curseur dans la Macro 0 (zéro). Des <read> successifs dans une même macro ajoutent les caractères dans la Macro 0 les uns à la suite des autres. Pour "purger" la Macro 0 avant de commencer une nouvelle lecture, mettre <read> dans une autre macro, et appeler la macro, comme ceci :

```
a:<all sa-b:if $0 = "." then left else rpt>!
b:<all read>!                          Envoie curseur en fin de phrase
```

Une meilleure façon de lire le curseur est de faire un "peek $10f5" :

`c:<all x=peek $10f5:ifnot x=174 then right rpt>!` (même effet, mais la macro est plus rapide, et n'ajoute pas un "espace" en allant à la ligne !)

`<disk>` `POMME-OUVERTE-&`

Enregistre le Préfixe (volume actuel + éventuellement sous-volume) dans la Macro 0 (zéro). Cette commande ne peut s'employer qu'à partir d'une liste de fichiers à l'écran. Un bref clignotement en haut à gauche de l'écran indique que la commande a été exécutée.

`<path>` `POMME-OUVERTE-*`

Enregistre le Nom d'accès (volume actuel + éventuellement sous-volume + nom de fichier) du fichier sélectionné dans la Macro 0 (zéro). Cette commande ne peut s'employer qu'à partir d'une liste de fichiers à l'écran. Un bref clignotement en haut à gauche de l'écran indique que la commande a été exécutée.

`<cell>` `POMME-OUVERTE--`

Enregistre le contenu d'une cellule (dans un Tableur) ou d'une catégorie (dans une Base de données) dans la Macro 0 (zéro). Placez le curseur dans la cellule ou la catégorie, et actionnez la commande : un bref clignotement en haut à gauche de l'écran indique que la commande a été exécutée.

Cette commande fonctionne avec tous les modes d'affichage (zoom avant ou arrière).

Elle enregistre littéralement :

1. En Tableur :
   - -> Une valeur, si c'est la valeur qui est affichée (même si elle est calculée).
   - -> Une formule, si la formule est affichée.

-> Un label.

2. En base de données :
   -> La rubrique complète, telle qu'elle apparaîtrait en format monofiche.

Elle fonctionne du clavier, ou en instruction à l'intérieur d'une macro. Dans ce cas, utiliser <cell> pour définir une chaîne (string). Exemple :

```
c:<asp:$3=cell:down:print $3 :rtn>!          copie une cellule
```

Attention : Cette macro est très commode, mais ne lui demandez pas ce qu'elle n'est pas censée faire (c'est-à-dire réfléchir à votre place) : comme il s'agit d'une copie littérale, vous risquez d'avoir quelques surprises si un label a des apparences de valeur, par exemple 14/7/88 (il sera traité en valeur), ou s'il commence par un caractère non admis en début de label, tel que &, é, etc (ce caractère n'apparaîtra pas). Pour copier des labels, tous les labels mais rien que des labels, modifier la macro comme suit :

```
c:<asp:$3='"'+cell:down:print $3 :rtn>!          copie un label
```

Note : L'instruction <cell> peut aussi enregistrer une ligne de traitement de texte. Dans le Tableur, elle a parfois des fantaisies. Voir dans le fichier MACROS.ULTRA.VF la macro ba--, qui "photographie" exactement ce qui est affiché dans une cellule à l'écran.

`<store>` `POMME-OUVERTE-,`

Emmagasine le contenu de Macro 0 (zéro), à concurrence de 15 caractères, dans une zone spéciale d'un fichier TdT, BdD ou Tbl. Le nom emmagasiné s'affiche à droite en bas de l'écran. Cette commande sert surtout à chaîner des fichiers (voir page 30).

`<recall>` `POMME-OUVERTE-.`

Restitue dans Macro 0 (zéro) le texte emmagasiné par <store>.

`<inc>` `POMME-OUVERTE-CONTROL-W`

Incrémente le caractère sous le curseur ("a" devient "b", "1" devient "2", etc.). Le tableau page 36 donne une liste complète des caractères dans l'ordre d'incrémentation.

`<dec>` `POMME-OUVERTE-CONTROL-A`

Opération contraire de <inc>.

`<bell>` `POMME-OUVERTE-CONTROL-G`

Envoie un "bip".

`<nosleep>` `POMME-OUVERTE-CONTROL-N`

Si une "macro dormante" est en attente, elle est annulée par cette commande (et ne sera donc pas activée le moment venu).

`<clear>` `POMME-OUVERTE-CONTROL-X`

Remet à zéro toutes les variables numériques, et retire les définitions de toutes les chaînes. Un bref clignotement en haut à gauche de l'écran indique que la commande a été exécutée.

Si vous avez un IIGS, la touche "CLEAR" (rectangle barré d'une croix) du pavé numérique est équivalente à CONTROL-X.

Instructions spéciales d'UltraMacros

Les instructions spéciales suivantes ne s'emploient que dans des définitions de macros. Elles ne peuvent pas être saisies au clavier, et n'ont d'ailleurs pas de commande-clavier équivalente. Elles sont utilisées telles quelles, c'est-à-dire sans paramètres.

`<input>`

Quand une macro arrive à cette instruction, elle s'arrête et attend Elle permet donc, en cours de macro, de saisir du texte (ou des commandes AppleWorks).

Un RETURN termine la saisie (il n'est pas inclus lui-même), et la macro continue. Si au lieu de taper RETURN vous faites CONTROL-à, la macro est définitivement interrompue.

Note : Il est recommandé de faire précéder <input> d'un message (avec l'instruction <msg>), pour rappeler à

l'utilisateur ce qu'il doit faire quand la macro s'arrête.

`<idf>`

Renvoie le numéro (unique) de l'application TimeOut en cours. Si TimeOut n'est pas en activité, le numéro est zéro. Cette instruction n'est jamais employée seule, elle doit faire partie d'une définition de variable ou de tout autre expression numérique. Exemple :

```
a:<all: A=idf:$1="Vous êtes dans l'application TimeOut No " + str$ A:msg $1>!
```

ou encore :

```
a:<ato:a=idf:if a=7 then msg ' FileMaster ':stop else if a=8 then msg ' Macro Compiler
':stop>!
```

`<ifkey>`

Vérifie si une touche donnée a été pressée , et ne continue qu'en cas de correspondance exacte. Cette instruction ne fait pas partie des règles de logique if-then-else. Exemple :

```
a:<all:sa-b rpt>!       Boucle continue de test
b:<all:ifkey rtn then print "RETURN": rtn>!
```

On sort de la macro avec ESCAPE.

`<key>`

Pause et attend qu'on appuie sur une touche (n'importe laquelle). Contrairement à SuperMacroWorks, la touche N'EST PAS transmise à AppleWorks. Employée seule, cette instruction sert uniquement à faire une pause.

Si <key> est dans une équation, elle renvoie la valeur de la touche enfoncée. Exemple :

```
a:<all:A=key:if A<128 then print A>!
```

Si vous appuyez sur RETURN, la macro imprimera 13 (qui est sa valeur ASCII). Si vous faites Pomme-Ouverte-RETURN la macro n'imprimera rien, car Pomme-Ouverte ajoute 128 à la valeur ASCII de la touche.

`<begin>`

Doit être associé à <rpt>. Sert à marquer le début d'une boucle de répétition, si on veut éviter la répétition de la macro entière.

`<rpt>`

Répète la macro, ou partie de la macro. Quand l'instruction <rpt> est rencontrée, UltraMacros revient en arrière jusqu'à <begin> ou jusqu'au début de la macro, et recommence.

Toutes les instructions qui suivent <rpt> ne seront jamais exécutées (sauf si elles sont introduites par une clause IF-THEN-ELSE). Une commande conditionnelle doit être utilisée pour sortir de la macro, pour éviter une boucle sans fin. Exemple :

```
g:<all:bell:rpt>!  N'arrête jamais. Faites ESCAPE pour couper court.
h:<all:L=32:begin:L=L+1:print chr$ L:if L<126:rpt>!  Imprime tous les carctères et s'arrête.
```

Note : pour des raisons obscures, l'emploi de certains nombres à l'intérieur d'une boucle begin/rpt détraque la macro. Ces nombres sont 17, et 37121 à 37375. Si vous en avez besoin, représentez-les à l'intérieur de la boucle par une variable. Vous risquez aussi d'avoir des problèmes si vous appelez une macro "Double-Pomme" de l'intérieur d'une telle boucle.

`<stop>`

Arrête net la macro. À employer pour empêcher une macro appelée de revenir à la macro appelante, ou pour sortir d'une boucle sans fin (voir illustration de la macro avec <findpo>, page 14).

Instructions spéciales d'UltraMacros avec paramètres

Le groupe suivant d'instructions est associé à des paramètres. La plupart des paramètres étant des variables, la première des choses à faire est de décrire les variables d'UltraMacros. La façon d'écrire les instructions sera expliquée ensuite.

Définition des variables numériques

On peut définir les variables numériques de nombreuses façons. Il faut veiller à ne pas re-définir accidentellement une variable si une autre macro doit s'en servir encore.

Il est recommandé d'utiliser les variables U, V, W, X, Y, Z comme variables "jetables" (dont on n'aura besoin qu'une

seule fois). Il sera ainsi entendu d'avance que n'importe quelle macro peut les re-définir.
Il est également recommandé de réserver la variable Q pour enregistrer le numéro d'un fichier que vous quittez par la commande OA-Q, et auquel vous comptez revenir (Les macros sa-1 et sa-2 du fichier MACROS.ULTRA.VF en donnent une illustration).

UltraMacros accepte 26 variables numériques, de A à Z. Le signe = sert à définir une variable. Par exemple : A = 25.

Une variable numérique peut être définie par référence à :

une autre variable	A à Z
un nombre décimal	7
un nombre hexadécimal	$10
une touche du clavier	key
la longueur d'une chaîne	len $1
la valeur d'une chaîne	val $2
la valeur d'une adresse	peek
un identificateur TimeOut	id£

Ces variables peuvent être associées à des expressions conditionnelles (if, ifnot) et à des opérateurs logiques (>, =, <).
Les termes sur lesquels agissent les opérateurs sont les opérandes.
On peut rassembler plusieurs opérandes au moyen des quatre opérateurs arithmétiques élémentaires (+ - / *). Les parenthèses ne sont pas autorisées, et les équations sont évaluées strictement de gauche à droite, comme elles se lisent.

Définition des variables texte

Les variables texte peuvent aussi être définies de diverses façons. UltraMacros accepte 10 variables texte, qui sont les chaînes $0 à $9. Une variable texte peut être définie par référence à :

un simple texte	"texte"
une autre chaîne	$2
une autre chaîne, indirectement	$(A) c'est-à-dire $2 si A=2
la date	date, date2
l'heure	time, time24
une saisie clavier	getstr
une cellule tableur	cell
une catégorie BdD	cell
du texte écran 80 col.	screen
la valeur ASCII d'une variable	chr$
l'équivalent chaîne d'une variable	str$
la partie gauche d'une chaîne	left
la partie droite d'une chaîne	right

Ces variables peuvent être associées à des expressions conditionnelles (if, ifnot) et à des opérateurs logiques (>, =, <).
Les termes sur lesquels agissent les opérateurs sont les opérandes.
On peut rassembler plusieurs opérandes par concaténation (opérateur +) Les parenthèses ne sont pas autorisées, et les équations sont évaluées strictement de gauche à droite, comme elles se lisent. Nombre maximum de caractères : 80 (les caractères en excès sont ignorés).

Quelques exemples de définition de variables texte :

— Un texte ordinaire, entre guillemets ou entre apostrophes :

```
a:<all:$8="Du texte ordinaire, ">!
b:<all:$9='et même du texte "entre guillemets"'>!
c:<awp:sa-a:sa-b:print $8><print $9>!
```

— La date et l'heure :

```
a:<all:$0=date:$1=date2:$2=time:$3=time24>!
b:<awp:sa-a:print $0><rtn:print $1><rtn:print $2><rtn:print $3>!
```

— Le contenu de cellule de Tbl, ou de la catégorie de Bdd où se trouve le curseur au moment donné :

```
a:<all:$8=cell>!
```

— Une portion de l'écran, grâce à l'instruction <screen> :

```
a:<all:$6=screen 10,1,15>!   Lit le nom du fichier affiché
b:<awp:sa-a:print $6>!   Imprime le nom du fichier affiché
```

— Une saisie directe à partir du clavier :

```
a:<all:$3=getstr 15>!
b:<awp:sa-a:print $3>!  Saisit 15 caractères au clavier et les imprime
```

— Une référence indirecte à une autre chaîne :

a:<all:$1="amis":$2="ennemis":A=1:E=2>!
b:<awp:sa-a>Préservez-moi de mes <print $(A)>, mes <print $(E)> je m'en charge <!>!

— Enfin, une chaîne peut être identifiée à une autre :

a:<all:$0=$1>!

Note : Quand il est prévu dans la définition d'une instruction qu'elle n'utilise qu'une seul paramètre, on ne peut pas la définir au moyen d'une équation. Exemple :
`"getstr 2 + 3"` est une erreur de syntaxe. Il faut écrire `"getstr 5"`.

Paramètres

Voici quelques-unes des manières possibles de définir les paramètres associés aux instructions d'UltraMacros :

MACRO
un nom de macro, par exemple sa-b ou ba-ctrl-d

NUM *(valeur numérique)*
un nombre décimal de 0 à 65535
un nombre héxadécimal de $0 à $FFFF
une variable de A à Z (c'est sa valeur qui sera utilisée)

NUM VAR *(variable numérique)*
une variable de A à Z (c'est sa valeur qui sera utilisée)

NUM2
une NUM (voir ci-dessus)
un code <key>
un code <peek>
un code <len>
un code <val>
un code <id£>

NUM EXP *(expression numérique)*
une NUM (voir ci-dessus)
une NUM2 (voir ci-dessus); si le compilateur envoie un message d'erreur, NUM2 n'est pas disponible pour cette définition de commande.
une équation NUM ou NUM2 ; les NUM sont liées par des opérateurs arithmétiques +,-,/, ou *. L'équation est évaluée telle quelle, de gauche à droite. Les valeurs vont de 0 à 65535, avec boucle en cas de dépassement : 0 - 1 = 65535, et 65534 + 3 = 1. Tous les nombres étant entiers, il n'est pas tenu compte des restes des divisions.

STRING
Une chaîne de caractères entre guillemets.

STRING VAR
Un nom de chaîne (variable texte), de $0 à $9.

STRING EXP *(expression texte)*

une STRING (voir ci-dessus)	une STRING VAR
un code <asc>	un code <chr$>
un code <str$>	un code <date>

un code <date2> un code <time>
un code <time24> un code <cell>
un code <screen> un code <getstr>
une équation STRING ; les STRING EXP sont liées par + exclusivement ; l'équation est évaluée de gauche à droite (maximum 80 caractères).

<asc STRING>

Convertit le premier caractère d'une chaîne en son équivalent ASCII. Exemple :

```
x:<all $0="Test":print asc $0>!       Imprime "84"
y:<all $0="z":x=asc $0 + 2:print x>!  Imprime "124"
```

Voir le fichier MACROS.ULTRA.VF pour des exemples d'emploi de <asc>

<chr$ NUM>

Renvoie le caractère ASCII dont la variable est le code (voir liste complète des codes ASCII en fin de manuel). Ainsi, la macro suivante imprime "1" :

```
a:<all:X=49:print chr$ X>!
```

Le tableau indique bien que "1" est le caractère qui correspond au code ASCII 49. Si X avait été 177, la macro aurait exécuté oa-1. En effet, 177 = 128 + 49.

Cette instruction, précédée de <prf>, est commode pour envoyer des codes spéciaux à votre imprimante.

<getstr NUM>

Affiche ">" à gauche en bas de l'écran : c'est une invite à saisir des caractères au clavier (fonctionne de façon analogue à OA-0). Exemple :

```
a:<all:$3=getstr 15>!
b:<awp:sa-a:print $3>!  Saisit 15 caractères au clavier et les imprime
```

Note : "getstr" et le paramètre doivent être séparés par un espace.

<goto MACRO>

Appelle la macro indiquée et déclenche son exécution. Quand la macro appelée est terminée, il n'y a pas de retour à la macro appelante.

Note : Si l'on veut que la macro appelée "rende la main" à la macro appelante (à la façon d'un GOSUB en Basic), il suffit d'indiquer le nom de la macro à appeler, sans le précéder d'un "goto".

```
a:<all:sa-b:print "merci":rtn>!
b:<all:msg 'Une touche, SVP':A=key goto sa-c print "fin":rtn stop>!
c:<all:if A=13 then print "bravo, et ":else>!
```

Examinez et faites fonctionner l'exemple ci-dessus : La macro sa-a envoie à la macro sa-b, qui envoie à la macro sa-c avec un goto. La macro sa-c, quand elle est terminée, ne rend jamais la main à sa-b (appel par goto), mais renvoie à sa-a (appel normal).

Maintenant, inversez les appels : mettez "goto sa-b" dans sa-a, et enlevez le "goto" dans sa-b. Re-compilez, exécutez sa-a et voyez la différence.

<hilight NUM EXP,NUM EXP,NUM EXP,NUM EXP>

Permet d'éclairer en vidéo inverse n'importe quelle partie de l'écran. L'exemple suivant inverse la totalité de l'écran :

```
h:<all:hilight 1,1,80,24>!
```

Le premier paramètre est la colonne gauche (1-80)
Le deuxième paramètre est la ligne supérieure (1-24)
troisième paramètre est la colonne de droite (1-80)
Le quatrième paramètre est la ligne inférieure (1-24)
Si le premier paramètre est 0 (zéro), les rangées indiquées reviendront en vidéo normale (si la première colonne est zéro, la deuxième valeur de colonne ne compte pas).

<left STRING VAR,NUM>

Conserve les NUM'ièmes caractères à gauche de la chaîne STRING VAR.

```
a:<all:$1="Beagle Bros":$2=left $1,6:print $2>! Imprime Beagle
```

<len STRING VAR>

Renvoie la longueur de la chaîne spécifiée. Exemple :

```
a:<all:L=len $0:print L>!
```

Tapez un mot avec oa-0, puis faites sa-a pour avoir sa longueur.

<msg STRING>

Cette instruction permet d'imprimer un message sur la ligne qui délimite le bas de l'écran (tirets ou ligne continue, selon les écrans). La syntaxe est analogue à celle de <print>. Les messages sont en vidéo inverse, sauf si le message est entre guillemets. Exemple :

```
m:<all:msg "Texte normal":key        (message, puis attend une touche)
  msg 'Texte en inverse':key         (idem                           )
  $9=time24 + " " + date:msg $9 :key (affiche heure et date, attend  )
  msg "">!                            Efface tout
```

Quand un message est affiché, le reste de la ligne se remplit avec l'avant-dernier caractère qui se trouvait déjà sur cette ligne. Ceci efface automatiquement les vestiges des messages précédents. Une chaîne nulle efface tout.

Note : Quand les délimiteurs du message sont des apostrophes, n'employez pas d'apostrophes dans le corps du message ; utilisez ° à leur place, si nécessaire.

<onerr OPTION>

Permet de maîtriser (dans une certaine mesure) les conséquences d'une erreur. En l'occurence, une "erreur" est toute commande qui conduit AppleWorks ou TimeOut à émettre un "bip" d'erreur. Normalement, une macro se poursuit sans tenir compte de ce genre d'erreur (et en réduisant le "bip" au silence par la même occasion). UltraMacros vous permet d'avoir trois attitudes dans ce genre de circonstances :

```
1:<all:onerr stop>!     Commande l'arrêt de la macro en cas d'erreur
```

Si la macro était appelée d'une autre macro, elle s'arrête et rend la main à cette macro, qui se poursuit. Seule la macro qui contenait cette instrucion est arrêtée.

```
2:<all:onerr off>!      Revient à la normale, ne tient compte d'aucune erreur
```

Annule l'effet d'une instruction onerr. La macro continue, pour le meilleur et pour le pire.

```
3:<all:onerr goto sa-h>!
```

En cas d'erreur, exécuter la macro appelée, puis revenir à la macro appelante (celle où l'erreur s'est produite)

Quand une séquence de macros est terminée, le statut "onerr" est toujours remis à la normale.

<posn VAR,VAR>

Attribue les coordonnées du curseur aux deux variables qui suivent l'instruction. Dans AppleWorks, les coordonnées s'interprètent comme suit :

	1ère VAR	2ème VAR
Traitement de texte	colonne	ligne
Base de données	catégorie	fiche
Tableur	colonne	ligne

Le curseur n'est pas dans l'une de ces trois applications, les deux variables seront calées à zéro. L'intruction <posn> est compatible avec les applications TimeOut qui utilisent les applications AppleWorks (par exemple, Graph travaille avec le Tableur, et <posn> peut y être utilisé).

Le fichier MACROS.ULTRA.VF, livré la disquette d'UltraMacros, contient des exemples de macros avec <posn>.

<pr£ NUM EXP>

Détermine vers où l'instruction <print> achemine l'information.

<pr£ 0> envoie tous les caractères de <print> à AppleWorks. C'est la situation normale, par défaut. <pr£ 1> envoie les caractères à la première imprimante de la liste des imprimantes d'AppleWorks. Comme AppleWorks est limité à 3 imprimantes, la limite de pr£ est également 3.

Il faut envoyer <pr£ 0> quand vous en avez fini avec les imprimantes, et que vous voulez revenir à AppleWorks.

Le fichier MC.IMPRIMANTE, sur la disquette UltraMacros, contient des exemples d'emploi de pr£.

<print>

De toutes les instructions d'UltraMacros, c'est peut-être celle qui se prête au plus grand nombre de variantes d'emploi. Le compilateur vous signalera la plupart de vos éventuelles erreurs, mais votre meilleure garantie de compétence est d'étudier à fond les explications qui vont suivre.

Imprimer du texte :

Comme avec le Basic, l'instruction <print> permet d'imprimer du texte. Il faut mettre ce texte entre guillemets ou entre apostrophes. La limite est de 70 caractères par chaîne.

```
a:<all:print "Ceci est du texte : <rtn>">! Ce <rtn> n'est évidemment pas exécuté !
```

```
a:<all:print 'Des "guillemets" ? Pas de problème.'>!
```
Les apostrophes servent à imprimer les guillemets.

Imprimer des variables numériques :

Print peut être utilisé pour afficher la valeur de n'importe quelle variable numérique. Par exemple, si la variable Q contient le numéro de Bureau AppleWorks du fichier auquel vous voulez revenir, utilisez la macro :

```
2:<all:oa-q:print Q:rtn>!
```

— Un texte peut précéder la variable. Par exemple :

```
a:<all:A=3:print "En décimal, A est égale à "A>!
```

— Si <print> est immédiatement suivi de "$", UltraMacros donne la valeur hexadécimale de la variable (en deux ou quatre caractères) :

```
a:<all:A=8:print$ "En hexa, A est égale à "A>!
b:<all:B=61453:print$ B>!
```

— Les variables numériques peuvent aussi être imprimées sous la forme de caractères : l'instruction <chr$> convertit la valeur numérique en caractère ASCII correspondant (liste complète des équivalences en fin de manuel) :

```
a:<all:X=185:Y=$41:print chr$ X:print chr$ Y>!
```

La liste indique que 185 correspond à Pomme-Ouverte-9 et $41 correspond au A majuscule. Donc la macro envoie en fin de fichier puis imprime"A".

Imprimer des chaînes (variables texte) :

Les dix chaînes $0 à $9 peuvent être imprimées isolément. Dans ce cas, il ne faut pas leur associer d'autres instructions. Les chaînes peuvent contenir du texte ou des caractères de commande. Pour définir une chaîne contenant des commandes et non du texte, il faut définir la Macro 0 (zéro), qui est aussi la chaîne $0, puis la renommer éventuellement :

```
a:<all:$2=$0:print $2>!    Exécute la Macro 0 (zéro)
```

En effet, comme la Macro 0 (zéro) et $0 sont identiques, les instructions sa-0 et <print $0> sont équivalentes. Une chaîne peut être imprimée par référence indirecte à une autre chaîne :

```
a:<all:$1="amis":$2="ennemis":A=1:E=2>!
b:<awp:sa-a>Préservez-moi de mes <print $(A)>, mes <print $(E)> je m'en charge <!>!
```

Note : Chaque instruction <print> doit obligatoirement être suivie de deux points ":" ou d'un crochet ">". Les autres instructions peuvent être séparées de la suivante par un simple espace, mais <print> est une exception à cette règle.

`<rem STRING>`

Cette instruction sert à incorporer une remarque à une macro. Même syntaxe que pour <print> : mettez le texte de la remarque entre guillemets ou entre apostrophes. Évidemment, la remarque n'intervient pas dans l'exécution de la macro.

Mais cette instruction ne fait-elle pas double emploi avec les commentaires que l'on peut mettre après le "!" qui marque la fin d'une macro, ou insérer dans le corps d'une macro en les mettant entre parenthèses ? Non , car les remarques <rem> sont compilées avec la macro, et restituées par l'option "Afficher macros actuelles" du compilateur, alors que les autres ne le sont pas. L'emploi de <rem> est donc un choix personnel.

Voici un exemple d'emploi de <rem> :

```
Z:<all:oa-d>S<rem "Déplace sur le presse-papiers":oa-9:rtn:rem "jusqu'au bout du
   fichier":left>!
```

Les remarques et commentaires sont toujours utiles pour ponctuer le déroulement d'une macro — surtout si vous y revenez dans un mois, dans un an...

`<right STRING VAR,NUM>`

Conserve les NUMièmes caractères à droite de la chaîne STRING VAR.

```
a:<all:$1="Beagle Bros":$2=right $1,4:print $2>! Imprime Bros
```

`<screen NUM EXP,NUM EXP,NUM EXP>`

Copie n'importe quelle partie de l'écran AppleWorks dans une variable texte. Exemple :

```
a:<all:$6=screen 10,1,15>!   Lit le nom du fichier affiché
```

Le premier paramètre est la colonne de gauche (1-80)

Le deuxième paramètre est la ligne (1-24)

Le troisième paramètre est la longueur de la chaîne (1-80)

Tous les caractères sont traités comme des caractères en vidéo ordinaire, quelle que soit leur aspect à l'écran ; en

particulier, cette instruction ne tient pas compte de la vidéo inverse.

<str$ VAR NAME>

Cette instruction convertit une variable numérique en son équivalent décimal (texte). Elle doit s'employer dans une équation. Exemples :

```
x:<all:A=4:$3=" A = " + str$ A + " ":print $3>!    Imprime " A = 4 "
y:<all:B=$ff:$1=str$ B:print $1>!                  Imprime "255
```

Cette instruction est commode pour inclure des variables dans un message, comme cet exemple qui identifie une application TimeOut par son numéro :

```
a:<ato:I=id£:$1="Application TimeOut No " + str$ I:msg $1>!
```

<val STRING VAR>

L'instruction <val> et le contraire de <str$>. Elle convertit une variable texte en une valeur numérique. Elle doit aussi s'employer dans une équation. Si la chaîne spécifiée commence par un caractère non-numérique, la valeur renvoyée sera toujours zéro. Si le premier caractère est un chiffre, il sera converti (ainsi que tous ceux qui le suivent immédiatement) en sa valeur numérique. Exemples :

```
a:<all:$3="essai4":A=val $3:print A>!          0
b:<all:$2="48612":X=val $2:print X>!           48612
c:<all:$8="280ZX":V=val $8:print V>!           280
d:<all:$0="14,88":P=val $0:print P>!           14
```

<wait NUM EXP>

Retarde le déclenchement d'une macro d'un laps de temps déterminé (ou jusqu'à ce qu'une touche soit pressée). La durée exacte du "retardement" dépend de votre ordinateur. A vous de le calibrer. La macro ci-après vous facilitera ce travail :

```
R:<all:msg ' Tapez une valeur de retard ':$0=getstr 5:R=val $0:bell:wait R:bell:msg " Quitter
   = ESC ":K=key:if K=27 stop else rpt>!
```

Autre emploi de <wait> : faire défiler un texte à l'écran automatiquement. La barre d'espace arrête/redémarre le défilement, ESC arrête la macro.

```
b:<awp:D=400:wait D:down:A=peek $c010:if A=155 stop else if A=160 goto sa-c else rpt>!
c:<all:key:goto sa-b>!
```

<wake MACRO at NUM EXP:NUM EXP>

Pour cette instruction, votre Apple doit être équipé d'une horloge. Cette instruction "met en sommeil" une macro, et la "réveille" à une heure préétablie (format time24).

Après avoir activé la commande <wake>, vous pouvez continuer à travailler comme si de rien n'était, y compris avec des macros ou des applications TimeOut. Quand l'horloge de votre Apple arrive à l'heure indiquée, la macro qui était "en sommeil" - nous l'appelerons "macro latente" - se déclenche d'elle-même.

Cette commande peut remplacer un réveil, servir de "pense-bête", ou sauver un fichier à intervalles réguliers, etc. Il ne peut y avoir qu'une seule macro "latente" à un moment donné, mais en se "réveillant" une macro peut, à son tour, mettre une autre macro "en sommeil" pour plus tard, et ainsi de suite.

```
a:<all:wake sa-b at 12:00>!
b:<all:bell bell bell:msg "Il est temps d'aller déjeuner":wake sa-c at 18:00>!
c:<all:bell bell bell:msg ' La journée est finie '>!
```

On peut utiliser des variables pour les heures et les minutes :

```
x:<all:M=50:H=7:goto sa-s>!
s:<all:oa-s:M=M + 10:if M=60 then M=0:H=H + 1:elseoff wake sa-s at H:M>!
```

Dans les deux macros précédentes, sa-x cale les heures et minutes à 7:50, et envoie à sa-s. Celle-ci sauve le fichier en cours, incrémente les minutes de 10 et prévoit de se réveiller elle-même à 8:00 heures, et ainsi de suite toutes les 10 minutes, jusqu'à ce que la commande <nosleep> soit employée.

Expressions logiques If-Then-Else

L'un des gros atouts d'UltraMacros est son aptitude à utiliser un jeu d'authentiques expressions logiques du type If-Then-Else. Les autres macro-programmes acceptaient (parfois) un "if" ou deux liés au curseur. Mais UltraMacros permet de manier une gamme complète d'instructions conditionnelles portant sur ses variables numériques et texte.

Les instructions sont au nombre de cinq : if, ifnot, then, else, elseoff.

<if>

C'est la clef de voûte de toutes les macros conditionnelles. L'instruction <if> est toujours :
-> suivie d'une variable (numérique ou chaîne texte),
-> qui elle-même est suivie d'un opérateur (plus grand que >, plus petit que <, ou égal =),
-> suivie à son tour du critère d'évaluation.
Exemple d'expression conditionnelle : `if A = 5`
Si l'expression est vraie, la macro qui la contient continue normalement. Si l'expression est fausse, la macro s'arrête (sauf s'il y a un <else> ensuite).

```
a:<all:if A=5 goto sa-5>!  Si A <> 5, la macro s'arrête, et n'envoie pas à sa-5.
```

<ifnot>

C'est l'opposé logique de <if>, c'est-à-dire que l'expression doit être fausse pour que la macro puisse continuer normalement.

```
a:<all:ifnot A=5 goto sa-5>! La macro s'arrête si A est égal à 5.
```

Toutes les expressions conditionnelles numériques ont la forme d'une des six variantes suivantes :

```
if NUM = NUM EXP        ou          ifnot NUM = NUM EXP
if NUM > NUM EXP        ou          ifnot NUM > NUM EXP
if NUM < NUM EXP        ou          ifnot NUM < NUM EXP
```

où NUM est une variable numérique et NUM EXP EST une expression numérique (voir leur description sous le titre "paramètres", page 17).
Toutes les expressions conditionnelles texte ont la forme d'une des six variantes suivantes :

```
if STRING = STRING EXP        ou          ifnot STRING = STRING EXP
if STRING > STRING EXP        ou          ifnot STRING > STRING EXP
if STRING < STRING EXP        ou          ifnot STRING < STRING EXP
```

ou STRING est une chaîne texte et STRING EXP EST une expression texte (voir également "paramètres", p. 17).
Si vous voulez savoir si une chaîne texte est plus petite, égale, ou plus grande qu'une autre, essayez cette macro :

```
c:<all:
   msg 'Entrez une chaine, puis RTN':
   $1 = getstr 10:
   msg 'Entrez une autre chaine, puis RTN':
   $2 = getstr 10:
   if $1 = $2 then msg $1 + " = " + $2><else
   if $1 < $2 then msg $1 + " < " + $2><else
   if $1 > $2 then msg $1 + " > " + $2>!
```

<then>

Cette instruction ne fait absolument rien, sinon consommer un octet de mémoire. Elle censée rendre la macro plus lisible (du moins en anglais).

<else>

Cette instruction permet de poursuivre la macro en "sautant par-dessus" une expression "non-vraie". Il n'y a pas de limite au nombre de <else> que l'on peut inclure dans une macro.
Cette instruction inverse l'état "vrai/faux", ou "exécute/n'exécute pas" de la logique du déroulement des instructions.
L'instruction <else> n'est d'ailleurs pas forcément liée à <if>. Elle peut être utilisée toute seule. Dans ce cas, la règle est la suivante : "Quand un <else> est rencontré au cours d'une macro, l'exécution saute au <else> suivant, ou à la fin de la macro". On peut utiliser cette propriété pour "occulter" une instruction dans une macro lors de sa mise au point. Comparez, par exemple, ces deux macros :

```
a:<all: print "A ": else print "Une":else print "Deux": else print "Trois">!
b:<all: else print "A ": else print "Une":else print "Deux": else print "Trois">!
```

Dans sa-a, "A" s'imprime, l'instruction suivante est occultée, "Deux" s'imprime, "Trois" est occulté. Dans sa-b, l'adjonction du premier <else> inverse la situation.
Bien que l'association de<if> et de <else> ne soit pas obligatoire, ces deux instructions font bien "équipe ensemble" :

```
a:<all:if A=5 then print "égale cinq":else print " différent de cinq">!
```

Selon la vérité ou non de la relation A=5, la macro imprime correctement "égale cinq" ou "différent de cinq".

<elseoff>

Cette instruction n'a pas d'intérêt en soi, à moins d'être précédée soit de <else>, soit d'une instruction conditionnelle. Son rôle est de forcer l'exécution de toutes les instructions qui la suivent, indépendamment de l'état "vrai/faux" ou "exécute/n'exécute pas" qui la précède. On peut dire qu'elle supprime l'état conditionnel, et rétablit le cours normal des choses.

```
a:<all:A=5:if A=5 then print "égale cinq":else print " différent de cinq":elseoff print "
actuellement">!
```

Cette macro imprime : " égale cinq actuellement". S'il n'y avait pas eu <elesoff>, elle aurait seulement imprimé "égale cinq". D'ailleurs, avec <elseoff>, "actuellement" sera imprimé dans tous les cas.

Pour programmeurs chevronnés uniquement !

Avec les trois instructions suivantes, UltraMacros vous permet d'accéder directement aux adresses mémoire et à des routines en langage machine.

Il n'est pas conseillé de s'y aventurer à moins de savoir exactement ce que l'on fait. Les Beagle Bros les ont incluses ici pour faire plaisir aux bidouilleurs et — pourquoi le cacher — pour se faire plaisir à eux-mêmes, et se garder une porte ouverte pour la construction de macros super-puissantes. Les fichiers inclus sur la disquette UltraMacros en contiennent déjà quelques exemples.

`<call>`

L'instruction <call> sert à appeller des sous-routines en langage machine. Elle fait simplement un jsr à l'adresse spécifiée. À vous de vous assurer que cette adresse est correcte, et que la routine rendra la main à la macro par un rts (avec tous les paramètres dans l'état voulu).

Un bon endroit pour insérer des routines en langage machine est le buffer temporaire d'AppleWorks de $800 à $9FF.

Prudence : Ce buffer est détruit lors d'un accès-disque, et par certaines commandes d'UltraMacros.

Quand une macro fonctionne, la page zéro de la mémoire auxiliaire est activée, ainsi que le deuxième banc de mémoire $D000. La page 1 de l'affichage 80 colonnes est également active. Si vous touchez à l'une quelconque de ces zones de mémoire, vous devez impérativement les ramener à leur état initial avant que votre sous-routine ne rende la main à AppleWorks. Sinon, c'est AppleWorks qui rendra l'âme.

L'instruction <call> est une arme à double tranchant, réservée aux experts. N'y touchez pas si vous n'êtes pas du nombre!

`<poke>`

Voilà une commande bien pratique, mais dangereuse elle aussi. Vous pouvez y recourir pour construire des routines en langage-machine que vous appellerez avec <call>.

On peut aussi l'utiliser pour changer des "drapeaux" dans AppleWorks. Par exemple, l'instruction <insert> impose le mode "insertion". Pour imposer le mode recouvrement, vous pouvez utiliser l'instruction suivante :

```
o:<all:poke $10f1,1>!
```

Le curseur d'insertion reste visible jusqu'à ce qu'une touche ait été enfoncée, et il passe alors en mode recouvrement. Pour un changement immédiat, ajoutez une touche "vide" à la macro :

```
o:<all:poke $10f1,1: ctrl-x>!
```

Les Beagle Bros ne sont pas autorisés à fournir de liste des adresses internes d'AppleWorks. Il vous faudra explorer par vous-mêmes.

`<peek>`

C'est l'instruction la moins dangereuse des trois. Elle lit, sans modifier. Voici un exemple tiré du fichier MACROS.ULTRA.VF. Le <peek> sert à déterminer le numéro de Bureau du fichier actuel (pour pouvoir y revenir) :

```
1:<all><q = peek $c54:oa-q esc>!  Va au menu principal, en se souvenant du numéro du fichier.
2:<all:oa-q print q:rtn>!         Revient à ce fichier.
```

Le compilateur

C'est une application TimeOut, qui vous permet de compiler de nouvelles macros. Il faut la lancer à partir d'un document de traitement de texte qui contient les définitions des macros à compiler. Le compilateur transformera ces définitions en macros actives, utilisables par UltraMacros.

Une deuxième option du compilateur permet de lister, dans un fichier traitement de texte, les macros actuelles, c'est-à-dire celles qui sont actives au moment où cette option est choisie.

Si vous avez lu ce manuel en commençant par le début, vous avez sans doute déjà fait connaissance avec le compilateur pour compiler quelques essais de macros. Voici, plus en détail, les références du compilateur.

Placez sur le Bureau un fichier traitement de texte avec des macros, faites Pomme-Ouverte-Escape, sélectionner "Macro Compiler". Deux options vous sont proposées :

1. Compiler nouvelles macros
2. Afficher macros actuelles

Compiler nouvelles macros

1. Faites RETURN pour la première option.

2. Le compilateur demande : "Pause à chaque ligne ?". Si vous répondez "O" pour "Oui", la compilation se fera pas-à-pas, une ligne à la fois ; il vaus faudra appuyer sur une touche à chaque fois pour avancer. Si vous répondez "N" pour "Non", la compilation démarrera, et ne s'arrêtera qu'en fin de macro, ou si elle bute sur une erreur.
3. Avant que le compilation ne commence, une autre question vous est posée : "Compilation du ? Début Curseur". La compilation commencera, selon votre réponse, du début du fichier ou de la position où se trouvait le curseur au moment où vous avez appelé le compilateur.
   Si la compilation commence en début de fichier, le compilateur recherchera le mot "start" (qui doit être placé seul, en début de ligne) ; s'il ne le trouve pas, il refusera la compilation avec le message d'erreur "START ???".
   Si la compilation commence au curseur, le "start" n'est pas nécessaire. En combinant "Pause à chaque ligne" et "Du curseur", on peut compiler sélectivement une seule macro (sortie par Escape).
4. Pendant la compilation, les lignes du texte défilent à l'écran. Le nom de la macro en cours de compilation est affiché, ainsi qu'un décompte des octets pris sur le maximum des octets disponibles de la table des macros.
5. Si la compilation s'est faite sans erreur, le compilateur annonce "C'est bon", et vous invite à appuyer sur une touche pour revenir à AppleWorks.
6. S'il y a eu erreur, le compilateur s'efforce de vous fournir un maximum d'informations. Quand vous aurez appuyé sur une touche, vous reviendrez dans le fichier AppleWorks avec le curseur placé sur (ou à proximité de) l'erreur.
7. Il se peut que le compilateur vous envoie le message "Macros d'origine intactes" et "C'est bon". Cela signifie que la compilation n'a pas pu se faire pour des raisons autres qu'une erreur, et que vous disposez toujours du jeu de macros actives précédemment.

Les messages d'erreur de compilation

Plusieurs types d'erreur peuvent arrêter le compilateur : seules les macros compilées jusqu'à l'erreur seront utilisables. Celles sitées au-delà de l'erreur, même si elles sont correctes, ne seront pas compilées. Lisez le message d'erreur et appuyez sur une touche : vous êtes renvoyé dans le fichier de macros, au voisinage immédiat de l'erreur à corriger.

Voici les messages possibles, et leur signification :

-> *C'est bon*
Le compilateur a passé en revue l'ensemble des macros, et n'y a relevé aucune des erreurs qu'il sait déceler.
En principe, celà signifie que la compilation s'est faite correctement. Le compilateur indique la dernière macro compilée, et le nombre d'octets pris dans la table des macros.
Si le message "Macros d'origine intactes" est affiché, la compilation n'a pas pu avoir lieu pour un défaut autre qu'une erreur de définition. Le jeu des macros précédent est toujours actif.
Il se peut aussi que le compilateur n'ait pas su déceler certaines erreurs potentielles, qui se manifesteront lors de l'exécution de la macro. Dites-vous bien que les messages d'erreur sont une aide, mais que la preuve ultime de "non erreur" est une macro qui fonctionne...

-> *START ???*
Vous avez demandé une compilation depuis le début du fichier, et le compilateur n'a pas trouvé le mot "start" qui doit lui signaler à partir d'où commence la compilation.
Le mot "start" (majuscules ou minuscules, peu importe) doit se trouver *seul*, en *début de ligne*.
Si vous avez choisi de compiler à partir du curseur, vous n'avez pas besoin de "start".

-> *Macro réservée*
Vous avez essayé de définir une macro réservée (voir page 13). Souvenez-vous que certaines touches réservées (comme RTN) ont un équivalent Ctrl-.

-> *Mauvais nom d'application*
Chaque macro doit comporter au début (après sa désignation et ":") un nom d'application. Les seules instructions correctes sont <all>, <awp>, <adb>, <asp> et <ato>.

-> *Table des macros pleine*
Vous avez défini plus de macros que la table ne peut loger. Il s'agit d'une capacité en octets, réservée aux seules macros (indépendemment de tout autre texte dans le fichier, qui n'entre pas en ligne de compte). La seule solution est de scinder votre fichier en plusieurs fichiers de macros.

-> *Erreur de syntaxe*
C'est le message "fourre-tout". Tout ce que le compilateur ne reconnaît pas est qualifié d'erreur de syntaxe. Dans la plupart des cas, ce sera une faute d'orthographe lors de la saisie.

-> *Manque de paramètres*
L'instruction concernée doit être associée à un ou plusieurs paramètres de plus. Reportez-vous à la liste des instructions qui exigent des paramètres.

-> *Trop de paramètres*
Il y a au moins un paramètre de trop. Reportez-vous à la liste.

-> *Erreur de logique*
La définition de la macro pêche par sa logique. Peut-être êtes-vous d'un autre avis, mais à quoi bon discuter ? Dites-vous bien qu'UltraMacros est très à cheval sur l'emploi de ses variables numériques et texte.

-> *Chaîne trop longue*
Une chaîne (texte) incluse dans une définition dépasse le nombre de caractères autorisé.

-> *END présent en ligne...*
La compilation a été arrêtée non par une erreur, mais par la présence de "end". Toutes les macros précédant "end" sont actives.

-> *Arrêté par Escape*
Vous avez appuyé sur Escape en cours de compilation pour arrêter. Toutes les macros déjà compilées sont actives.

Afficher macros actuelles

Cette option vous permet d'imprimer toutes les macros actuellement actives dans un fichier traitement de texte. La liste peut bien sûr être modifiée et re-compilée. La raison principale de cette option est de matérialiser les macros saisies au clavier avec la commande oa-W. Le fichier de traitement de texte peut être vide ou non. S'il ne l'est pas, la liste s'imprimera en fin de fichier.

Macros : options

C'est une autre application TimeOut, qui permet à l'utilisateur d'UltraMacros d'en modifier manuellement un certain nombre de fonctions. L'accès, comme pour toutes les applications TimeOut, se fait directement à partir d'AppleWorks par Pomme-Ouverte-Escape. Une fois sélectionné, Macro Options affiche le menu suivant :

1. *Démarrer Task*
2. *Créer fichier Task*
3. *Sauver la table des macros*
4. *Désactiver macros*
5. *Réactiver pas-à-pas*
6. *Version*
7. *Autres activités*

Les deux premières options concernent les fichiers "Task", qui sont décrits plus longuement — ainsi que leur emploi — à la page page 27.

1. Démarrer un fichier Task

Cette option, quand elle est sélectionnée, passe en revue le disque AppleWorks à la recherche des fichiers Task. Leurs noms sont affichés à l'écran. Sélectionnez le fichier qui vous convient, ou faites Escape pour revenir au menu de Macro Options.

2. Créer un fichier Task

Cette option transforme l'ensemble des macros alors actives dans AppleWorks en un fichier Task, qu'elle sauve sur le disque de démarrage d'AppleWorks. Vous êtes d'abord invité à attribuer un nom au nouveau fichier Task. Il faut indiquer un nom de fichier répondant aux règles ProDos (de 1 à 15 caractères au maximum commençant par une lettre, et ne contenant que des lettres ou des chiffres éventuellement séparés par des points).
Si le disque de démarrage d'AppleWorks n'est pas localisé, vous êtes invité à l'insérer (vous pouvez interrompre la procédure avec Escape).
Une fois que le fichier a été écrit, vous pouvez retirer le disque et remettre en place celui qui était dans le lecteur.

3. Sauver la table des macros
Cette option prend l'ensemble des macros alors actives dans AppleWorks et les sauvegarde dans le fichier ULTRA.SYSTEM de votre disque AppleWorks. Comme c'est le fichier qui démarre AppleWorks, ces macros seront immédiatement disponibles à chaque démarrage (sans compilation supplémentaire).
Avant que les macros ne soient sauvegardées, vous devez préciser si vous voulez activer la macro de démarrage automatique. Si vous répondez "Oui", la première macro de cet ensemble sera automatiquement activée aussitôt après chaque démarrage d'AppleWorks.
Si le fichier ULTRA.SYSTEM n'est pas localisé, vous serez invité à insérer votre disque de démarrage AppleWorks (vous pouvez interrompre la procédure avec Escape). Une fois que le fichier a été écrit, vous pouvez retirer le disque et remettre en place celui qui était dans le lecteur.

4. Désactiver les macros
Cette option déconnecte complètement UltraMacros d'AppleWorks. Choisissez cette option pour imprimer avec le buffer d'impression d'Applied Engineering. Si les macros ont été désactivées, l'option 4 devient "Réactiver macros", et vous permet de revenir aux macros et à la souris.

5. Réactiver pas-à-pas
Cette option permet d'exécuter les macros octet par octet, ce qui est commode pour débusquer les erreurs éventuelles.
L'option 5 devient "Désactiver pas-à-pas", pour permettre le retour à la normale.
Pendant l'exécution pas-à-pas, la macro fait une pause chaque fois qu'elle envoie un caractère à AppleWorks, et il vous faut appuyer sur une touche pour faire exécuter le "pas" suivant. En appuyant sur Escape vous arrêtez la macro, mais vous ne sortirez du mode "pas-à-pas" qu'en repassant par l'option 5 de Macro Options.

6. Version
Vous renseigne sur la version actuelle d'UltraMacros. La version française est une exclusivité © Daniel Lurot & Éditions MEV.

7. Autres activités
Un tiroir s'ouvre, rempli de tout une série de commandes utiles :

1. *Régler clignotement curseur*
2. *Régler bouton souris*
3. *Régler la souris*
4. *Désactiver la souris*
5. *Réactiver Key-Lock*
6. *Réactiver sécurité écran*
7. *Régler délai sécurité écran*
8. *Définir format date*

Autres activités

Ces options vous permettent de modeler le tempérament d'UltraMacros à vos goûts personnels. Au moment de quitter le menu (par Escape), vous devrez indiquer si si vous souhaitez sauver les modifications sur le disque AppleWorks. Si vous répondez "Oui", le disque sera réécrit, et vous retrouverez les modifications à chaque démarrage.

1. Régler le clignotement du curseur
Cette option modifie la vitesse de clignotement du curseur AppleWorks. La vitesse actuelle est affichée, et vous êtes invité à donner une nouvelle valeur, choisie entre 1 et 255 (1 pour le clignotement le plus rapide, 255 pour le plus lent).

2. Régler le temps de latence du clic de la souris

Cette option modifie le temps de latence du bouton de la souris (lors des sélections cliquées). Si vous sautez plusieurs étages de menus avec un seul clic, il y a lieu d'allonger le temps de latence. Le délai actuel est affiché, et vous êtes invité à donner une nouvelle valeur, choisie entre 1 et 255 (1 pour le délai le plus court, 255 pour le plus long).

3. Régler la souris

Il s'agit, avec cette option, de définir le rapport entre le parcours de la souris et celui du curseur.
Le réglage horizontal actuel est affiché d'abord, et vous êtes invité à donner une nouvelle valeur, choisie entre 1 et 255 (1 pour le déplacement le plus rapide, 255 pour le plus lent).
Puis vient le réglage du déplacement vertical (même procédure).
Les possesseurs du IIGS peuvent aussi modifier la vitesse de la souris (valeur élevée ou basse) en passant par le tableau de bord.

4. Désactiver la souris

UltraMacros reste branché, mais ignore la "carte" souris. Utile pour ceux qui n'ont pas encore de souris avec leur //c, car la carte du //c n'est pas amovible, et l'absence de souris peut être cause de perturbations. Si votre curseur fait des sauts sporadiques et incontrôlés (quel que soit votre Apple), essayez cette option. Les options 2 et 3, si vous essayez de les sélectionner, vous préviendront que la souris est introuvable. L'option 4 est devenue "Réactiver la souris".

5. Réactiver Key-Lock

Key-Lock est une option surtout destinée à aider les utilisateurs d'AppleWorks handicapés physiquement, mais aussi tous ceux qui ont du mal à utiliser les combinaisons Pomme-Ouverte et Pomme-Fermée en association avec d'autres touches.
Avec cette option, quand on appuie sur les touches Pomme-Ouverte ou Pomme-Fermée celles-ci restent "engagées". Une pression sur la touche suivante les "libère", et enregistre la commande complète Pomme-Ouverte (ou Pomme-Fermée), + "touche". Pour la commande Double-Pomme, appuyer d'abord sur Pomme-Ouverte, puis sur Pomme-Fermée, puis sur la touche voulue.
Pour rappeler qu'une touche Pomme est "engagée", un O, un S ou un B en vidéo inverse s'affichent en bas à droite de l'écran (respectivement pour Pomme-Ouverte, Pomme-Fermée ou Double-Pomme), et le curseur cesse de clignoter. Escape annule la manoœuvre.
Pour revenir en mode normal, choisir l'option 5 (qui affiche "Désactiver Key-Lock).

6. Réactiver sécurité écran

En choisissant cette option, vous activez l'extinction automatique de l'écran. L'écran s'éteint si aucune touche n'a été enfoncée pendant une durée déterminée. Il suffit d'appuyer sur une touche pour ré-enclencher la vidéo. Pour annuler l'option, sélectionner 6 à nouveau (qui est devenu "Désactiver sécurité écran).
Cette extinction sert à ménager le phosphore de votre écran, sinon celui de votre cerveau !

Note : Si vous avez suspendu votre travail sur une option du genre Pomme-Ouverte-C (Copier), qui affiche une option en vidéo inverse au bas de l'écran, l'écran ne s'éteindra pas. Mais l'extinction d'écran fonctionne à l'intérieur des applications TimeOut.

7. Régler délai sécurité écran

Choisissez cette option pour fixer le délai avant extinction de l'écran. La durée du délai est fonction du rythme de clignotment du curseur. Si vous modifiez la vitesse de clignotement, le délai avant extinction sera modifié proportionnellement.
L'option affiche le délai actuel, et vous demande de proposer une autre durée. Donnez une valeur de 1 (minimum) à 255 (maximum).

8. Définir format date

Cette option vous donne à choisir, pour les commandes <date> et <date2>, entre le format américain et le format français. Le format français est le format par défaut. Il est reconnu par AppleWorks 1.4 dans toutes ses applications (TdT, BdD, Tbl).

Transfert de données Tableur/BdD

C'est une application TimeOut, qui est jointe en prime à la disquette UltraMacros. Grâce à elle, le transfert de données Tableur vers Base de données (via le Presse-papiers) et vice-versa n'est plus un problème.

Pour transférer des données dans le sens Tableur -> BdD, commencez par copier (avec oa-C) du Tableur sur le Presse-papiers. Appelez ensuite le menu TimeOut avec oa-Esc, et sélectionnez "Data Converter". Les données dans le Presse-papiers sont instantanément converties en données de type BdD, il ne reste plus qu'à les récupérer dans une base de données avec la commande habituelle AppleWorks : oa-C, "du Presse-papiers".
Il faut que la BdD soit en format "tableau" au moment de la copie. Chaque colonne du Tableur correspondra à une catégorie de la BdD.
Même procédure pour la copie en sens inverse, BdD -> Tableur : copiez d'abord normalement dans le Presse-papiers, puis actionnez "Data Converter", et récupérez normalement dans le Tableur. Chaque catégorie de la BdD deviendra une colonne du Tableur.

Patch Control-à

AppleWorks 1.4 contient un "bug" qui empêche d'envoyer le code CONTROL-à dans une définition d'interface d'imprimante.
Le menu principal de la disquette d'UltraMacros vous permet de faire en quelques secondes un patch à AppleWorks pour corriger ce défaut. Il suffit de choisir l'option et de suivre les instructions.

Note : En cours de saisie de macro, si vous voulez insérer CONTROL-à, tapez Pomme-Ouverte-Ctrl-à. Si vous tapez Ctrl-à directement le macro s'arrêtera. De même, en écrivant une définition de macro, écrivez <oa-ctrl-à>.

La souris

En ajoutant UltraMacros à AppleWorks, vous dotez ce dernier par la même occasion de la souris : dans le TdT, les BdD, le Tbl et les menus — sans parler de toutes les applications TimeOut.
Règle générale : Les déplacements de la souris ont le même effet sur le curseur — en plus rapide et efficace — que l'emploi des flèches. Pour un déplacement beaucoup plus rapide, appuyez en même temps sur Pomme-Ouverte (souvenez vous, par exemple, que <down> avec la flèche descend d'une ligne, alors que <oa-down> descend d'un écran entier).
Contrairement à certains autres logiciels de traitement de textes, vous n'avez pas besoin de "cliquer" après avoir "placé" le curseur : la saisie peut se faire directement.

Le clic de la souris a les effets suivants :

1. Dans un menu AppleWorks ou TimeOut, il sélectionne une option (même effet que RETURN).
2. Dans un fichier TdT, le bouton de la souris enfoncé fait défiler le texte — vers le haut ou vers le bas, selon le mouvement initial de la souris. Il n'est donc pas nécessaire de "rouler" la souris : il suffit d'amorcer le mouvement, et d'enfoncer le bouton.

La "vivacité" de la souris peut être réglée, dans le sens horizontal comme dans le sens vertical, par l'application "Macro Options" (option "Autres activités"). On peut aussi régler le temps de latence du "clic", et même désactiver complètement la souris si on le souhaite (c'est nécessaire si votre Apple n'est pas équipé d'une souris).

Note : Un des effets de la "règle générale" ci-dessus est que, lors de la création des formats d'étiquette, en BdD, on peut déplacer très commodément les rubriques sur l'écran en mettant le curseur sur la première lettre d'une rubrique, en enfonçant la touche Pomme-Ouverte et en les "traînant" avec la souris.

Comment lier des fichiers entre eux

Vous pouvez lier des fichiers avec les commandes <store> et <recall>.
L'instruction <store> emmagasine le contenu de la Macro 0 (zéro), à concurrence de 15 caractères, dans une zone spéciale d'un fichier TdT ou Tbl. Le nom emmagasiné s'affiche à droite en bas de l'écran.
L'instruction <recall> restitue dans la Macro 0 (zéro) le texte emmagasiné par <store>.
L'instruction <find> — améliorée par rapport à celle de SuperMacroWorks — est également un puissant moyen de chaînage par noms de fichier.
Enfin, les macros sa-1 et sa-2 de MACROS.ULTRA.VF sont un exemple de chaînage entre fichiers sur Bureau, appelés par numéros.

Sur la disquette UltraMacros, le fichier "Chainage" (TdT) illustre diverses façons d'enchaîner des fichiers.

Menus de démarrage

Avec UltraMacros, vous pouvez confectionner des menus de démarrage efficaces pour exécuter telle ou telle tâche. Vous pouvez soit en faire des fichiers "Task" (voir ci-dessous), soit les utiliser à l'intérieur d'AppleWorks. Pour voir fonctionner un exemple de menu, chargez le fichier MACROS.STARTUP, étudiez-le, et faites-le fonctionner.

Fichiers "Task"

Un fichier "Task" est un ensemble de macros pré-compilées, et sauvegardées sur le disque de démarrage AppleWorks sous la forme de fichier de type SYS. Ils peuvent être lancés vite et facilement pour exécuter une tâche donnée (d'où leur nom).

Un nouveau "Task" peut être démarré de diverses manières :

1. De l'intérieur d'AppleWorks : en appelant l'application TimeOut "Macro Options" (voir page 27).
2. De l'extérieur d'AppleWorks, au moyen d'un sélecteur de programmes (MouseDesk, Bird's Better Bye, ProSel, Squirt).
3. Du Basic Applesoft avec la commande "exec" suivie du nom du fichier Task : "–NOM.TASK".

Le disque UltraMacros contient un exemple de fichier Task (c'est le fichier Task.Sample).
Quand un fichier "Task" est lancé de l'extérieur d'AppleWorks, il commence par exécuter ULTRA.SYSTEM, qui à son tour démarre AppleWorks. Enfin la première macro de l'ensemble des macros est exécutée ; elle se présente comme suit :

```
$:<all:rtn:rtn:sa-1>!
```

où sa-1 est la macro suivante

La macro sa-§ contient les deux <rtn> dont AppleWorks a besoin lors des deux poses de démarrage (l'une pour "insérer la face Programme", l'autre pour "insérer la date"). Moralité : ce modèle de sa-§ convient à AppleWorks 1.4, non patché pour "sauter" ces étapes, sur disquette 3,5' et avec horloge. Sinon, il faut l'adapter à votre configuration AppleWorks/Apple. Par exemple, pour AppleWorks 1.4 sur "floppy" 5,25', et sans horloge, la première macro serait :

```
$:<all:key:input:rtn:sa-1>!
```

C'est la deuxième macro qui entame la tâche proprement dite. Quand le fichier "Task" est lancé de l'intérieur d'AppleWorks, il se branche directement sur cette deuxième macro. Dans l'exemple, la deuxième (et dernière) macro est :

```
1:<all oa-q esc rtn down down rtn rtn>NOUVEAU<rtn>!
```

Elle utilise la séquence classique <oa-q esc> pour aller au menu principal d'AppleWorks (elle a le mérite de fonctionner n'importe-où dans AppleWorks), et crée un nouveau fichier de traitement de texte appelé NOUVEAU.

Cet exemple est réduit à sa plus simple expression. Le potentiel des fichiers "Task" est illimité, car ils permettent de commander directement et automatiquement n'importe quelle activité AppleWorks + TimeOut. On peut imaginer les exemples suivants :

1. Charger au démarrage une liste de fichiers prédéterminée sur le Bureau.
2. Copier un disque 3,5' en RAM à l'aide de TimeOut FileMaster.
3. Charger un fichier Tableur et l'afficher en graphique avec TimeOut Graph.
4. Charger un fichier Traitement de textes et l'imprimer avec TimeOut SuperFonts.
5. Lancer toute tâche complexe, mais répétitive, comme par exemple imprimer un compte-rendu (ou bilan comptable) hebdomadaire à partir d'une BdD ou du Tableur.

Il est recommandé de constituer un fichier de vos macros les plus courantes, et de le sauver sous la forme d'un fichier Task (Macros.Defaut), que vous pourrez relancer instantanément après avoir utilisé des macros plus spécialisées.

Le cas à part de la Macro 0 (zéro)

La Macro 0 (zéro) est une macro vraiment pas comme les autres, en raison de nombreuses instructions spéciales d'UltraMacros qui recourent à elle.
Vous pouvez, certes, utiliser la touche 0 (zéro) comme une autre, pour la saisie au clavier par oa-W, mais il y a bien

d'autres façons de définir la Macro 0. C'est justement pour cette raison que cette macro ne doit servir que de mémoire très temporaire, car on aura souvent recours à elle comme "plaque tournante" pour le transfert d'informations, et il n'est que trop facile d'écraser son contenu par inadvertance.

Voici toutes les façons de définir la Macro 0 (zéro) rien qu'à partir du clavier :

1. Avec la commande oa-W de saisie au clavier : un maximum de 80 caractères peuvent être ainsi enregistrés "en écho" dans la Macro 0, pendant que ces mêmes caractères sont normalement transmis à AppleWorks.
2. Avec la commande oa-0 (zéro) : un maximum de 60 caractères peuvent être enregistrés dans la Macro 0, sans que ces caractères soient transmis à AppleWorks (voir page 14).
3. Avec la commande oa-/, qui enregistre le caractère sous le curseur dans la Macro 0. Cette commande peut être répétée, les caractères s'ajoutant les un aux autres (voir page 15). C'est l'équivalent-clavier de <read>.
4. Avec la commande oa-& (dans un menu) pour enregistrer dans la Macro 0 le Préfixe ProDos actuel (voir page 15). C'est l'équivalent-clavier de <disk>.
5. Avec la commande oa-* (dans un menu) pour enregistrer dans la Macro 0 le nom d'accès complet ProDos (voir page 15). C'est l'équivalent-clavier de <path>.
6. Avec la commande oa-- pour enregistrer dans la Macro 0 la catégorie de BdD actuelle, ou la cellule Tbl sous le curseur (voir page 15). C'est l'équivalent-clavier de <cell>.
7. Avec la commande oa-, pour rappeler dans la Macro 0 le nom de fichier inscrit précédemment avec <store> (voir page 16). C'est l'équivalent-clavier de <recall>.

Les instructions citées ci-dessus peuvent évidemment être utilisées à l'intérieur d'une définition de macro, et il faut encore y ajouter l'instruction <$0=...>, qui permet de définir une variable texte de diverses manières (voir page 18).
Une fois que le contenu de la Macro 0 a été défini, ce contenu peut être envoyé à AppleWorks par la commande sa-0 (Pomme-Fermée-0). Ainsi l'association de sa-- et sa-0 permet de "cueillir" le contenu d'une catégorie de BdD, ou de cellule de Tbl, et de le "transplanter" n'importe où ailleurs dans AppleWorks.
Enfin, le contenu de la Macro 0 peut être utilisé de façon spécialisée par <store> et par <find> (voir page 13) pour une recherche dans un menu.
Le fichier Macro 0 Memo, sur le disque UltraMacros, résume ces diverses utilisations.

Une macro bien utile

Dans le fichier MACROS.ULTRA.VF (celui qui contient les macros installées d'office par UltraMacros), la macro sa-ctrl-P vous aide à comptabiliser vos appels téléphoniques interurbains. Elle se sert des fichiers "Adresses" et "Telephones" inclus dans la disquette UltraMacros.
Cette macro est commentée pas-à-pas. Reportez-vous y. En partant de la rubrique "No Téléphone" de la fiche-adresse de votre correspondant, il vous suffit de faire sa-ctrl-P pour que la macro aille dans le fichier "Telephones" et y crée une nouvelle fiche donnant le nom du correspondant, son numéro de téléphone et l'heure du début d'appel. Dès que l'appel est terminé, il suffit de taper sa-+ pour enregister l'heure de fin de communication.

SuperMacroWorks : qu'en reste-t-il ?

Une bonne surprise : la compatibilité avec SuperMacroWorks a été en majeure partie préservée, pour assurer la continuité. Les familiers de SMW n'aurant pas à "désapprendre" pour "réapprendre". Ils seront au contraire très avantagés par rapport aux nouveau-venus complets.
Mais UltraMacros n'est PAS une nouvelle version de SuperMacroWorks. On peut dire que c'est un nouveau langage de macro-programmation, qui a conservé (mais ce n'est qu'une partie du tout) un grand nombre des commandes de SMW.

Deuxième bonne surprise : la syntaxe d'UltraMacros est aussi très proche de SMW, de sorte que la plupart des macros simples de SMW risquent de fonctionner telles-quelles. Pour ce qui est des macros plus complexes, le compilateur indiquera très clairement les parties à revoir.
Mais à vrai dire, il serait un peu ridicule de vouloir à tout prix récupérer des macros complexes de SMW avec UltraMacros : ce dernier est infiniment plus puissant grâce à ses manipulations de variables, et il y a probablement des raccourcis inédits à trouver. Un régal en perspective pour les amateurs de programmation.
Le gain de puissance d'UltraMacros vient de sont association avec TimeOut, sur le quel il a pu se décharger de tâches d'intendance telles que la compilation, la sauvegarde des macros compilées, l'impression des macros actuelles, les

options souris, etc. L'espace-mémoire ainsi récupéré a permis d'inclure toutes les fonctions nouvelles.

UltraMacros a des instructions nouvelles suivantes par rapport à SuperMacroWorks :
<ato>, <begin>, <call>, <cell>, <clear>, <else>, <elseoff>, <getstr>, <goto>, <hilight>, <id£>, <left$>, <nosleep>, <onerr>, <peek>, <poke>, <posn>, <print>, <pr£>, <rem>, <right$>, <screen>, <then>, <time24>, <val>, <wait>, <wake>.

La commande ancienne <find> a été très améliorée ; elle permet maintenant la recherche par nom dans n'importe quel menu. Elle fonctionne en particulier dans l'index de Bureau (appelé par oa-Q) et dans le menu TimeOut (appelé par oa-Esc).
Certaines instructions de SMW n'existent plus. Elles ne sont plus nécessaires, car leur fonctions sont assurées par d'autres commandes :
<end?>, <menu>, <resume>, <swap>, <oa-£>, <oa-$>, <Lprint>, <compile>, <save0>, <load0>, <0=>, <if0>, <var=>, <incvar>, <decvar>, <varnot>, <var>.

Les commandes de la famille "var", en particulier, sont désormais remplacées par un jeu complet de 26 variables numériques.
La Macro 0 (zéro) a été allongée jusqu'à 80 caractères, et peut être définie de façon encore plus variée.
L'instruction <msg> est plus puissante maintenant ; on peut y inclure la date et l'heure (mais les caractères souris n'y sont plus autorisés : il faut se contenter de texte normal ou en vidéo inverse). Dans le tableur, le message est maintenant sur la bonne ligne.
La faculté de charger automatiquement une police de caractères PowerPrint a été abandonnée au profit de l'intégration de TimeOut avec SuperFonts.

Note : Vous pouvez laisser votre SuperMacroWorks sur le disque de démarrage AppleWorks. Le programme d'installation d'UltraMacros renommera le fichier SUPER.SYSTEM en SUPER.SYSOLD, et il vous restera la possibilité de le faire démarrer avec un sélecteur.

Liste des instructions d'UltraMacros

Voici une liste complète, dans l'ordre alphabétique, de toutes les intructions d'UltraMacros, avec renvoi à la page du manuel qui contient leur description :

Instr.	Page	Description sommaire
asc	20	Donne équivalent ASCII du 1er car.d'une chaîne
adb	11	Code d'application AppleWorks
ahead	13	Trouve espace suivant
all	10	Code d'application AppleWorks
asp	11	Code d'application AppleWorks
ato	11	Code d'application TimeOut
awp	10	Code d'application AppleWorks
ba-	11	Double-Pomme-
back	13	Trouve espace précédent
begin	17	Début de boucle (voir rpt)
bell	16	Actionne le Bip AppleWorks
call	25	Exécute sous-routine en langage machine
cell	15	Copie ds.Macro 0 catégorie BdD ou cellule Tbl
chr$	20	Renvoie équivalent ASCII d'une variable
clear	16	Purge toutes variables (numériques et texte)
ctrl-	11	Touche CONTROL
date	13	Donne date format texte
date2	13	Donne date format numérique
dec	16	Décrémente caractère sous curseur
del	11	Touche DELETE
disk	15	Copie Préfixe ProDos dans Macro 0
down	11	Flèche en-bas

else	24	Inverse l'état conditionnel d'une macro
elseoff	24	Annule le caractère conditionnel d'une macro
esc	11	Touche ESC
find	13	Trouve un nom, ou un retour-chariot
findpo	14	Trouve une option d'impression
getstr	20	Définit une chaîne-texte à partir du clavier
goto	20	Va à la macro spécifiée
hilight	20	Met en vidéo inverse un partie de l'écran
id£	17	Renvoie le No d'application TimeOut
if	23	Exécute la macro si la condition est vérifiée
ifkey	17	Vérifie si la touche indiquée a été enfoncée
ifnot	24	Exécute la macro si condition non vérifié
inc	16	Incrémente caractère sous curseur
input	16	Accepte saisie jusqu'à RTN
insert	15	Impose le mode insertion
key	17	Pause, attend touche, renvoie sa valeur
lc	15	Impose minuscule sous curseur
left	20	Renvoie partie gauche d'une chaîne
len	20	Renvoie la longueur d'une chaîne
msg	21	Affiche message à l'écran
nosleep	16	Annule macro "en sommeil"
oa-	14	Pomme-Ouverte
onerr	21	Fixe procédure traitement d'erreurs
path	15	Copie nom d'accès ProDos compet ds.Macro 0
peek	25	Renvoie valeur d'une adresse mémoire
poke	25	Écrit valeur en adresse mémoire
posn	21	Renvoie coordonnées du cuseur
print	21	Imprime variables
pr£	21	Dirige impression vers écran ou imprimantes
read	15	Copie le caractère sous curseur ds.Macro 0
recall	16	Renvoie ds.Macro 0 texte mis ds.fichier par STORE
rem	22	Remarque incluse dans une macro
right	22	Flèche à-droite
rpt	17	Répète la macro
rtn	11	Touche RETURN
sa-	13	Pomme-Fermée
screen	22	Inscrit portion d'écran ds.Macro 0
spc	11	Touche ESPACE
stop	17	Arrête net toutes macros
store	16	Met Macro 0 ds.fichier TdT ou Tbl pour RECALL
str$	23	Renvoie équivalent texte d'une variable
tab	11	Touche TAB
then	24	Conjonction purement esthétique après <if>
time	13	Affiche heure, format am/pm
time24	13	Affiche heure, format 24 h.
up	11	Flèche en-haut
uc	15	Impose majuscule sous curseur
val	23	Renvoie valeur numérique d'une chaîne
wait	23	Attente (jusqu'à fin délai ou touche enfoncée)
wake at	23	Prépare macro "en sommeil"
zoom	15	Impose zoom arrière

Annexe
Utilitaires TimeOut

Il s'agit d'une "application" TimeOut qui accompagne tous les produits de la série TimeOut. Elle possède diverses fonctions qui vous faciliteront beaucoup l'utilisation des applications TimeOut.

Les fonctions des Utilitaires

Assurez-vous tout d'abord que le fichier TO.UTILITAIRES a bien été copié sur votre disque d'applications TimeOut. Démarrez AppleWorks, et tapez Pomme-Ouverte-Escape pour appeler le menu TimeOut. Sélectionnez Utilitaires, validez avec RETURN. Le menu suivant apparaît :

```
Applications TimeOut - Utilitaires

1. Configurer
2. Charger en mémoire
3. Purger de la mémoire
4. Changer de statut
5. Changer le nom
6. Trier le menu
7. Ajouter applications
```

Nous allons passer en revue ses diverses options.

Configurer

C'est l'option qui permet de modifier les réglages standard de vos applications TimeOut (ce qu'on appelle parfois "valeurs par défaut"), tels que : type de l'imprimante, chemin d'accès aux polices de caractères ou aux fichiers utilisés par l'application, etc. Certaines applications simples n'ont pas besoin de configuration.

Comment configurer ? Sélectionner l'option 1 dans le menu UTILITAIRES, puis sélectionnez l'application à configurer. Si l'application est configurable, un menu sera affiché. Il contiendra les paramètres modifiables pour cette application, ainsi que la valeur actuellement attribuée à chaque paramètre (entre crochets < >).
Sélectionnez le paramètre à modifier, attribuez-lui sa nouvelle valeur (suivez les instructions à l'écran). Il faut que le disque avec le fichier de l'application soit en ligne, pour qu'il puisse être mis à jour. La prochaine fois que vous utiliserez l'application, elle utilisera les nouveaux paramètres. Pour revenir au menu Utilitaires, utilisez ESCAPE.

Charger en mémoire

Les applications TimeOut peuvent résider sur disque, ou s'installer en mémoire. Si, selon la configuration habituelle, une application réside sur disque au démarrage d'AppleWorks, vous pouvez la charger en mémoire pour la durée de la séance de travail.
Choisissez l'option, puis l'application à charger, et validez.

Purger de la mémoire

Si AppleWorks vous envoie un message pour dire qu'il manque de mémoire sur le Bureau, vous pouvez faire de la place en dégageant une application installée en mémoire. Choisissez l'option, puis la ou les applications que vous voulez purger. Notez que la quantité de mémoire disponible affichée à droite et en bas de l'écran augmente après chaque "purge". Les applications purgées résideront sur disque pour le restant de la séance de travail.

Changer le statut

Cette option vous permet de décider si une application résidera habituellement sur disque ou en mémoire.
Le fait de changer le statut ne change rien pour la séance de travail en cours ; le nouveau statut s'appliquera au prochain démarrage d'AppleWorks. Pour modifier immédiatement la situation d'une application, utiliser les options "Charger en mémoire" ou "Purger de la mémoire".

Changer le nom

Il vous est possible de modifier le nom de l'application, tel qu'il est affiché sur le menu TimeOut. Vous pouvez le franciser, le rendre plus évocateur ou plus obscur... une seule condition : le nouveau nom ne doit pas être plus long que l'ancien.

Trier le menu

Quand vous installez TimeOut sur votre disque d'AMORCAGE AppleWorks, vous avez le choix de faire trier ou non le menu TimeOut par ordre alphabétique. Si vous n'avez pas fait trier le menu à ce moment-là, cette option vous permet de rattraper l'omission.

Ajouter des applications

Cette option vous donne une grande souplesse de gestion de vos applications TimeOut à partir de plusieurs disquettes. Elle permet d'ajouter de nouvelles applications à tout moment, sans sortir d'AppleWorks, en créant de nouveaux menus d'applications TimeOut.

Vous pouvez fort bien démarrer AppleWorks avec un minimum d'applications TimeOut installées de façon permanente (à la limite, vous pourriez vous contenter des seuls Utilitaires !), puis charger les applications dont vous avez besoin, au fur et à mesure, à partir de leurs disques respectifs.

Procédure : choisissez l'option "Ajouter des applications", et indiquez le disque (ou nom d'accès) des nouvelles applications. Validez. Les noms de toutes les nouvelles applications seront chargés les uns à la suite des autres, et un nouveau menu TimeOut sera créé. Vous pouvez répéter l'opération pour un autre disque : un troisième menu sera créé, et ainsi de suite.

Les nouveaux menus TimeOut sont créés pour la durée de la séance de travail. Il n'y a pas de limite au nombre de menus que vous pouvez créer ainsi. C'est une pure question de convenance personnelle (et de gestion de disquettes).

> *Note : Cette option ne fait que créer un nouveau MENU. Elle ne recopie pas les applications elles-mêmes sur le disque AppleWorks. Pour pouvoir les utiliser, il faut que le disque qui les contient reste en ligne. Si vous avez besoin de libérer le lecteur (et si vous avez assez de mémoire libre), chargez vos applications en mémoire avec l'option "Charger en mémoire" décrite plus haut.*

On circule d'un menu TimeOut à l'autre avec la touche TAB. La même touche TAB permet de permuter les applications dans les options 1 à 4 du menu "Utilitaires".

Table des équivalents ASCII

	B	Car.	H	B	Car.	H	B	Car.	H	B	Car.	H
	0	ctrl-à	128	32	Esp	160	64	à	192	96	`	224
	1	ctrl-A	129	33	!	161	65	A	193	97	a	225
	2	ctrl-B	130	34	"	162	66	B	194	98	b	226
	3	ctrl-C	131	35	£	163	67	C	195	99	c	227
	4	ctrl-D	132	36	$	164	68	D	196	100	d	228
	5	ctrl-E	133	37	%	165	69	E	197	101	e	229
	6	ctrl-F	134	38	&	166	70	F	198	102	f	230
Bell	7	ctrl-G	135	39	'	167	71	G	199	103	g	231
Left	8	ctrl-H	136	40	(	168	72	H	200	104	h	232
Tab	9	ctrl-I	137	41	)	169	73	I	201	105	i	233
Down	10	ctrl-J	138	42	*	170	74	J	202	106	j	234
Up	11	ctrl-K	139	43	+	171	75	K	203	107	k	235
	12	ctrl-L	140	44	,	172	76	L	204	108	l	236
Rtn	13	ctrl-M	141	45	-	173	77	M	205	109	m	237
	14	ctrl-N	142	46	.	174	78	N	206	110	n	238
	15	ctrl-O	143	47	/	175	79	O	207	111	o	239
	16	ctrl-P	144	48	0	176	80	P	208	112	p	240
	17	ctrl-Q	145	49	1	177	81	Q	209	113	q	241
	18	ctrl-R	146	50	2	178	82	R	210	114	r	242
	19	ctrl-S	147	51	3	179	83	S	211	115	s	243
	20	ctrl-T	148	52	4	180	84	T	212	116	t	244
Right	21	ctrl-U	149	53	5	181	85	U	213	117	u	245
	22	ctrl-V	150	54	6	182	86	V	214	118	v	246
	23	ctrl-W	151	55	7	183	87	W	215	119	w	247
	24	ctrl-X	152	56	8	184	88	X	216	120	x	248

	25	ctrl-Y	153	57	9	185	89	Y	217	121	y 249
	26	ctrl-Z	154	58	:	186	90	Z	218	122	z 250
Esc	27	ctrl-°	155	59	;	187	91	°	219	123	é 251
	28	ctrl-ç	156	60	<	188	92	ç	220	124	ù 252
	29	ctrl-$	157	61	=	189	93	$	221	125	è 253
	30	ctrl-^	158	62	>	190	94	^	222	126	¨ 254
	31	ctrl-_	159	63	?	191	95	_	223	127	Del 255

Table des matières

Chapitre 1 — UltraMacros TimeOut : présentations 2

Chapitre 2 — Installation 3

Installation de TimeOut 4

Chapitre 3 — UltraMacros : prise en main 7

Les nouvelles commandes qu'apporte UltraMacros 7
Saisie de vos macros au clavier 8
Création de nouvelles macros 9
Adaptation des macros existantes 9
Création d'un fichier de macros 9
Création d'une macro permanente 10

Chapitre 4 — UltraMacros : Références 10

Anatomie d'une macro 10
Macros réservées 13
Nouvelles commandes Pomme-Ouverte 14
Instructions spéciales d'UltraMacros 16
Instructions spéciales d'UltraMacros avec paramètres 17
Expressions logiques If-Then-Else 23
Pour programmeurs chevronnés uniquement ! 25
Le compilateur 25
Macros : options 27
Autres activités 28
Transfert de données Tableur/BdD 29
Patch Control-à 30
La souris 30
Lier des fichiers entre eux 30
Menus de démarrage 31
Fichiers "Task" (dédiés à une tâche spécifique) 31
Le cas à part de la Macro 0 (zéro) 31
Une macro bien utile 32
SuperMacroWorks : qu'en reste-t-il ? 32
Liste des instructions 33

Annexe — Utilitaires TimeOut 35

Tableau des valeurs ASCII 36

Loi du 3 juillet 1985 : Protection des auteurs de logiciels de Micro-Informatique

Sept articles de la loi du 3 juillet 1985, applicable à compter du 1er janvier 1986, sont consacrés à la lutte contre le piratage des logiciels. Cette loi définit la notion de contrefaçon et prescrit les sanctions applicables aux contre-facteurs.

Définition de la contrefaçon au sens de la loi du 3 juillet 1985

L'article 47 de la loi du 3 juillet 1985 dispose :
"Par dérogation au 2 de l'article 41 de la loi N° 57–298 du 11 mars 1957 précitée, toute reproduction autre que l'établissement d'une copie de sauvegarde par l'utilisateur ainsi que toute utilisation d'un logiciel non expressément autorisée par l'auteur ou ses ayants–droit est passible des sanctions revues par ladite loi".

La reproduction d'un logiciel

Constitue le délit de contrefaçon de logiciels, la reproduction d'un logiciel faite pour l'usage privé de celui à qui est fourni le logiciel. La seule reproduction licite en vertu de la nouvelle loi vise à l'établissement d'une copie de sauvegarde pour l'usage de celui à qui est fourni le logiciel.

Utilisation en dehors des conditions prévues par la licence concédée au concessionnaire

Constitue également un délit de contrefaçon une utilisation d'un logiciel qui n'a pas été expressément autorisée par l'auteur ou ses ayants–droit. Ainsi, toute personne utilisant sans autorisation le logiciel qui lui a été fourni ou l'utilisant pour un autre usage que celui qui lui a été concédé par le propriétaire des droits, peut-être poursuivie pénalement et civilement par l'auteur, même si cette utilisation ne constitue pas une violation du monopole de la reproduction.

La mise en œuvre des poursuites

Elle se fait par le biais d'une saisie-contrefaçon suivie ensuite d'une assignation délivrée au contrefacteur soit devant le tribunal civil, soit devant le tribunal correctionnel.

La saisie-contrefaçon peut-être uniquement descriptive. Dans ce cas, l'huissier ou le commissaire se contente de saisir deux exemplaires des œuvres contrefaites pour obtenir un moyen de preuve. Elle peut être réelle et porter sur l'intégralité des exemplaires saisis. Toutefois dans ce cas, l'autorisation préalable du juge, saisi par voie de requête, (dont bien sûr la partie adverse n'a pas connaissance) est nécessaire ; de plus, la saisie-contrefaçon réelle de tous les exemplaires a lieu aux risques et périls du saisissant.

La saisie est faite soit par un huissier de justice, soit par un commissaire de police. L'un et l'autre peuvent se faire assister par un homme de l'art de leur choix.

Sanctions applicables au délit de contrefaçon au sens de la loi du 3 juillet 1985

L'article 47 de la nouvelle loi prévoit que le piratage de logiciels est passible des sanctions prévues par la loi du 11 mars 1957, qui donne à l'auteur du logiciel la possibilité d'exercer l'action en contrefaçon afin de faire condamner le contrefacteur, tant devant le tribunal civil que le tribunal correctionnel, puisque la contrefaçon en matière de droits d'auteur, constitue un délit pénal. La loi du 11 mars 1957 renvoie elle-même aux articles 425 à 429 du Code Pénal. Les deux textes principaux sont les articles 425 et 426 du Code Pénal qui prescrivent les sanctions suivantes :

Article 425 — Alinéa 1 et 2

Toute édition d'écrits, de composition musicale, de dessin, de peinture ou de toute autre reproduction, imprimée ou gravée en entier ou en partie, au mépris des lois et règlements relatifs à la propriété des auteurs, est une contrefaçon, et toute contrefaçon est un délit.
La contrefaçon, sur le territoire français, est punie d'un emprisonnement de trois mois à deux ans et d'une amende de 6 000 francs ou de l'une de ces deux peines seulement.

Article 426

Est également un délit de contrefaçon toute reproduction, représentation ou diffusion par quelque moyen que ce soit d'une œuvre de l'esprit en violation des droits de l'auteur, tels qu'ils sont définis et réglementés par la loi.